

Optimization of Airport Terminal-Area Air Traffic Operations under Uncertain Weather Conditions

by

Diana Michalek Pfeil

B.A., University of California at Berkeley (2004)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

© Massachusetts Institute of Technology 2011. All rights reserved.

Author
Sloan School of Management
April 26, 2011

Certified by
Hamsa Balakrishnan
Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by
Dimitris Bertsimas
Co-Directory, Operations Research Center

Optimization of Airport Terminal-Area Air Traffic Operations under Uncertain Weather Conditions

by

Diana Michalek Pfeil

Submitted to the Sloan School of Management
on April 26, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Convective weather is responsible for large delays and widespread disruptions in the U.S. National Airspace System, especially during summer. Although Air Traffic Flow Management algorithms exist to schedule and route traffic in the face of disruptions, they require reliable forecasts of airspace capacity. However, there exists a gap between the spatial and temporal accuracy of aviation weather forecasts (and existing capacity models) and what these algorithms assume. In this thesis we consider the problem of integrating currently available convective weather forecasts with air traffic management in terminal airspace (near airports).

We first demonstrate how raw convective weather forecasts, which provide deterministic predictions of the Vertically Integrated Liquid (the precipitation content in a column of airspace) can be translated into reliable and accurate probabilistic forecasts of whether or not a terminal-area route will be blocked. Given a flight route through the terminal-area, we apply techniques from machine learning to determine the probability that the route will be open in actual weather.

This probabilistic route blockage predictor is then used to optimize terminal-area operations. We develop an integer programming formulation for a 2-dimensional model of terminal airspace that dynamically moves arrival and departure routes to maximize expected capacity. Experiments using real weather scenarios on stormy days show that our algorithms recommend that a terminal-area route be modified 30% of the time, opening up 13% more available routes during these scenarios. The error rate is low, with only 5% of cases corresponding to a modified route being blocked while the original route is in fact open. In addition, for routes predicted to be open with probability 0.95 or greater by our method, 96% of these routes are indeed open (on average) in the weather that materializes.

In the final part of the thesis we consider more realistic models of terminal airspace routing and structure. We develop an A*-based routing algorithm that identifies 3-D routes through airspace that adhere to physical aircraft constraints during climb and descent, are conflict-free, and are likely to avoid convective weather hazards. The proposed approach is aimed at improving traffic manager decision-making in today's

operational environment.

Thesis Supervisor: Hamsa Balakrishnan
Title: Assistant Professor of Aeronautics and Astronautics

Acknowledgments

My most sincere thank you goes to my advisor Hamsa Balakrishnan, for helping me stumble upon and define research problems, for honest and helpful feedback, for supporting and encouraging my research path, and all the proof-reading and input of research write-ups. It's rare to find a advisor so energetic and enthusiastic about research, and I feel fortunate to have been able to soak up the excitement and keep myself interested and motivated.

I would also like to thank my other two thesis committee members Cindy Barnhart and Amedeo Odoni for their interest, plentiful questions, feedback, and guidance. I thank Daniel Delahaye for being a fantastic host during my research visit to ENAC, and Andreas Schulz for advising me during my first year at the ORC. I was lucky to TA for Arnie Barnett, and am thankful for his advice and insights into life and statistics, and for his incredible sense of humor.

Obtaining data is (unfortunately) one of the greatest challenges of doing applied research. I thank Marilyn Wolfson and the MIT Lincoln Lab Weather Sensing group for kindly providing me with access to CIWS weather data. In particular I'd like to thank Rich DeLaura and Mike Matthews for answering my (plentiful) questions, for the informative and helpful conversations, and for their enthusiasm for my research.

I'm thankful to (the faculty, students, staff, and environment at) the ORC for the passion and love I developed for all things operations research. We really have a fascinating, fun, and almost all-encompassing field, and it's been a thrill to learn about some of my now-favorite topics such as primal-dual algorithms.

To students who became friends at my two homes at MIT, the ORC and ICAT labs: thank you for your advice, research (and life) discussions, encouragement, and sense of humor. Thanks in particular to Jason Acimovic, Doug Fearing, Claire Cizaire, Ioannis Simaiakis, Nikolas Pyrgiotis, and Lishuai Li for your friendship. To friends outside the ORC, I'd especially like to thank Lisa Messeri, Abby Spinak, and Michael Baym for the climbing and outdoor adventures, which rank among my best times during grad school.

Finally, I'd like to thank my family: my mother Lubica Michalek for always believing in my abilities and decisions, and calling constantly to talk; my father Peter Michalek for always emphasizing my education, and encouraging and supporting my love of math; I could not ask for more brave and loving parents. To my little sista Lillian Michalek, for her love and growing friendship. I'd also like to thank my parents-in-law Garry and Kathy Pfeil for their generosity and kindness.

Most importantly, I'd like to thank my husband and favorite collaborator Greg Pfeil for among countless other things, being there for me throughout my PhD, from the stresses of studying for quals and generals to the elation of having research published and defending my thesis. Thank you for your cheer, sense of humor, and love.

Contents

1	Introduction	17
1.1	The causes and impacts of air traffic delay	18
1.2	Background and Related Literature	20
1.2.1	The National Airspace System	20
1.2.2	Air Traffic Flow Management Models	23
1.2.3	Dynamic Airspace Configuration	24
1.2.4	Convective Weather Forecasts	25
1.2.5	Validation of Aviation Weather Forecasts	26
1.2.6	Capacity Estimation	27
1.2.7	Integration of aviation weather forecasts with ATFM	28
1.3	Contributions of thesis	28
1.3.1	Probabilistic model of route robustness	29
1.3.2	Model for terminal-area dynamic airspace	30
1.3.3	Identification of realistic 3D airspace trajectories for tactical planning	31
1.3.4	Validation with actual weather conditions	32
1.4	Organization of thesis	32
2	Convective weather forecasts for traffic flow management	33
2.1	Lincoln Lab convective weather forecast	33
2.2	Pixel-based evaluation of CIWS forecast	35
2.2.1	Evaluation of skill scores	35
2.2.2	Distribution of actual weather given forecast	36

2.3	Forecast objectives from an operational perspective	39
2.4	Model for a route-based forecast	41
2.4.1	Definition of open route	41
2.4.2	Terminal airspace setup	41
2.4.3	Dynamic weather grid	42
2.4.4	Identifying robust routes through terminal airspace	44
2.5	Generation of data sets	44
2.5.1	Selection of weather scenarios	44
2.5.2	Selection of routes	45
2.5.3	Details of route dataset	49
2.5.4	Conclusion	53
3	Prediction of robust routes through terminal airspace	55
3.1	Features for route blockage prediction	56
3.2	Identifying robust routes using individual features	58
3.2.1	The conditional probability of route blockage	58
3.2.2	Comparison of results across features	60
3.2.3	Results across flight direction and planning horizon	62
3.3	Feature selection using mutual information	64
3.4	Classification algorithms for route availability	66
3.4.1	Introduction to supervised learning and classification	67
3.4.2	Classification training objectives	68
3.4.3	Classification of route blockage	69
3.5	Classification Results	71
3.5.1	Results for Ensemble of SVMs Classifier	72
3.5.2	Sensitivity of classifier to inner radius R_I	74
3.5.3	Sensitivity of classifier to wiggle room B	75
3.5.4	Results for Weighted Random Forest classifier	77
3.5.5	Two more classifiers	78
3.6	Probabilistic prediction of route availability	79

3.7	Conclusion	82
4	Dynamic Reconfiguration of Terminal Airspace	85
4.1	Motivation for dynamic terminal airspace	86
4.1.1	Terminal airspace structure and air traffic constraints	86
4.1.2	Terminal airspace operations during convective weather	87
4.1.3	Comparison of terminal and enroute Dynamic Airspace Configuration	90
4.2	Terminal-area DAC model setup	91
4.2.1	Terminal airspace sectorization model	91
4.2.2	Route robustness model	92
4.3	Model 1: Dynamic terminal routes	93
4.3.1	Algorithm Description	93
4.3.2	Algorithm Analysis	94
4.3.3	Stability of route selection	98
4.4	Model 2: Dynamic terminal routes with renegotiation of sector boundaries	101
4.4.1	Integer programming formulation	101
4.4.2	Computation of model solution	105
4.4.3	Analysis of results for fixed parameter settings	106
4.4.4	Analysis of results as objective function varies	111
4.5	Conclusion	113
5	3D routing in terminal airspace	115
5.1	Constraints for realistic terminal airspace trajectories	116
5.2	Blank slate design of terminal airspace	117
5.2.1	Previous Research	117
5.2.2	Problem Statement	118
5.2.3	Solution Approach	118
5.2.4	IP formulation for selection of optimal terminal fixes	119
5.2.5	A^* -1: algorithm for 3D conflict-free route identification	121

5.3	Results for blank slate terminal design	123
5.3.1	Chicago airspace configuration	124
5.3.2	Demand data	124
5.3.3	Altitude profiles	126
5.3.4	Results: optimal terminal fixes	128
5.3.5	Results: conflict-free 3D routes	130
5.4	3D routes through convective weather	132
5.4.1	Algorithm	133
5.4.2	A^* -2: Further modifications to the A^* algorithm	133
5.4.3	Experimental Results	134
5.4.4	Caveats and future directions	136
5.4.5	Conclusion	137
6	Conclusions	139
6.1	Review of Thesis	139
6.2	Future Directions	141
A	Interviews with Air Traffic Controllers	145
B	Pseudocode for 3D route planning	149
B.1	Details of the A^* algorithm	149
B.2	Pseudocode for A^* -2 : IS_SINK and PRUNE	150

List of Figures

1-1	Flight delays by cause for 2008.	18
1-2	Flight delays in the United States over the last 10 years. Source: FAA OPSNET	19
1-3	Diagram of a terminal-area structured in a four corner post configuration.	22
2-1	Diagram of MIT Lincoln Lab's CIWS VIL forecast near ATL	34
2-2	Four skill scores for the CIWS weather forecast, at increasing time horizons.	36
2-3	Histogram of true VIL when Level 3 VIL (in the range [133, 162]) is forecast, for 30-min (left) and 60-min (right) horizons.	37
2-4	Distribution of true weather level given a forecast at a pixel, for varying time horizons across a large set of summer 2008 weather scenarios.	38
2-5	Illustration of potential advantages of route-based forecast evaluation over a pixel-based skill scores.	40
2-6	Diagram of terminal airspace: (a) depicts the set-up and (b) shows the dynamic weather grid.	42
2-7	An example of departure routes through a dynamic weather grid.	43
2-8	Illustration of the eight routes sampled for an arrival forecast scenario.	47
2-9	Examples of routes synthesized in the forecast grid (left), and validated against the observed weather (right).	50
2-10	Skill scores for the route-based weather dataset improve over scores for pixel-based forecast evauluation.	52
3-1	Illustration of minumum cut M and features 9-11 of the weather forecast.	57

3-2	Probability of route blockage conditional on the value of each feature, for departures at the 60-minute planning horizon.	60
3-3	Sensitivity of route blockage to flight direction, for feature 1 (mean VIL along route) at the 60-minute planning horizon.	63
3-4	Sensitivity of route blockage to varying planning horizon, for feature 8 (max VIL density near route) and departures.	64
3-5	Comparison of mutual information across features and planning horizons for arrivals (left) and departures (right)	65
3-6	Process for training an ensemble classifier on an imbalanced dataset. .	70
3-7	Skill scores for the ESVM forecast of route blockage (solid lines), as compared to the raw route-based weather forecast (dotted lines). . . .	74
3-8	Sensitivity of the ESVM classifier to R_I , as measured in terms of accuracy (left) and false positive rates (right).	75
3-9	Box plot of sensitivity of the ESVM classifier to wiggle room B , as measured in terms of accuracy (gray) and false positive rates (blue). .	76
3-10	Comparison of false positive and accuracy rates of WRF for each planning horizon, as a function of weight (penalty against false positives).	78
3-11	Validation of classifier's probabilistic prediction of route availability, for departures at increasing planning horizons, compared to the calibration line $x = y$	81
4-1	Terminal airspace structure for ATL, typical of other four corner-post airports.© Google 2010, Image U.S. Geological Survey, USDA Farm Service Agency.	86
4-2	Comparison of air traffic flow near ATL during nominal and convective weather conditions.	88
4-3	Model of terminal airspace with standard sectors and fixes	92
4-4	Selection of an optimal route and associated fix for one sector, as a function of the weather forecast and standard fix location.	94
4-5	Example of dynamic route movement for six consecutive time periods.	99

4-6	Model of terminal airspace with standard sectors divided into wedges and allowable range of sector boundary locations.	102
4-7	Sectorization results for two illustrative weather scenarios, one with a sector boundary shift (bottom), and one without (top).	107
4-8	Sectorization results as a function of objective function parameters . .	112
5-1	Diagram of demand-based optimal outer fix selection model	120
5-2	Air traffic over Chicago airspace on June 4 2008 (top), with close-ups of ORD (bottom left) and MDW (bottom right). © Google 2010, Image U.S. Geological Survey, USDA Farm Service Agency.	125
5-3	Altitude profiles for arrivals (left) and departures (right), estimated empirically using flight tracks for ORD.	127
5-4	Comparison of outer fixes selected by IP approach (left) with the STAR and SID fixes at ORD (right).	129
5-5	3D conflict-free terminal routes, at different angles	132
5-6	Experimental results for three weather scenarios. Red routes are conflict free routes identified by the A^*-2 algorithm.	135

List of Tables

2.1	Weather scenarios considered during 2007 and 2008 convective seasons.	46
2.2	Overall statistics for each of the 10 datasets in Data2007.	51
3.1	Validation results for the ensemble SVM classifier (ESVM), compared to the raw weather forecast (Fx)	73
4.1	Overall results for dynamic route and fix movement algorithm.	95
4.2	Analysis of route movements.	96
4.3	Dynamic route movement as a function of the predicted probability of being open, \hat{p}	97
4.4	Stability of route movements between consecutive time periods T_6 and T_7	100
4.5	Overall results for terminal route and fix optimization with renegotiation of sector boundaries.	108
4.6	Analysis of route movements for model 2.	110
4.7	Dynamic route movement of model 2 as a function of the predicted probability of being open, \hat{p}	111
5.1	Estimated altitude profiles for arrivals and departures for varying confidence intervals	128

Chapter 1

Introduction

The increase in demand for air travel in the United States has been accompanied by an increase in congestion and delays in the National Airspace System (NAS), and has made the system more susceptible to weather disruptions. This problem is particularly intense during summer months, when travel demand is high and there is frequent convective weather activity (thunderstorms) over much of the continental United States.

Although mathematical models exist to optimize flight routes and minimize delay in the face of reduced capacity, these models tend to make strong simplifying assumptions about the form and accuracy of weather forecasts, and usually do not validate solutions against actual weather scenarios and operational conditions. This thesis focuses on bridging the gap between available aviation weather forecasts and air traffic flow management algorithms in terminal airspace. We develop a probabilistic model of convective weather impact using archived weather data and techniques from machine learning. This weather model is integrated into several traffic flow management models which dynamically restructure terminal airspace to use more robustly available routes. We validate all models against real weather scenarios, and evaluate potential benefits to operations.

1.1 The causes and impacts of air traffic delay

Following a steady increase in revenue over the last decade, the global commercial airline industry generated \$483 billion dollars of revenue in 2008 (IATA, 2010). In the domestic United States alone, this activity included over 660 million passenger enplanements and the transport of over 13 million revenue ton-miles of cargo (Bureau of Transportation Statistics, 2010a). Although 2009 saw a decline in air carrier capacity due to the downturn in the global economy, the Federal Aviation Administration (FAA, 2009) forecasts an annual increase of 2.2% in passenger enplanements through 2025.

This large demand for air travel, coupled with limited capacity in the National Aviation System (NAS), has led to rampant year-round flight delays. In both 2007 and 2008, over 20% of domestic flights in the United States were delayed by more than 15 minutes (Bureau of Transportation Statistics, 2010a). According to an FAA-sponsored study estimating the economic impact of delays, the cost of all US air transportation delays totaled \$32.9 billion, including \$8.3 billion cost to airlines and a

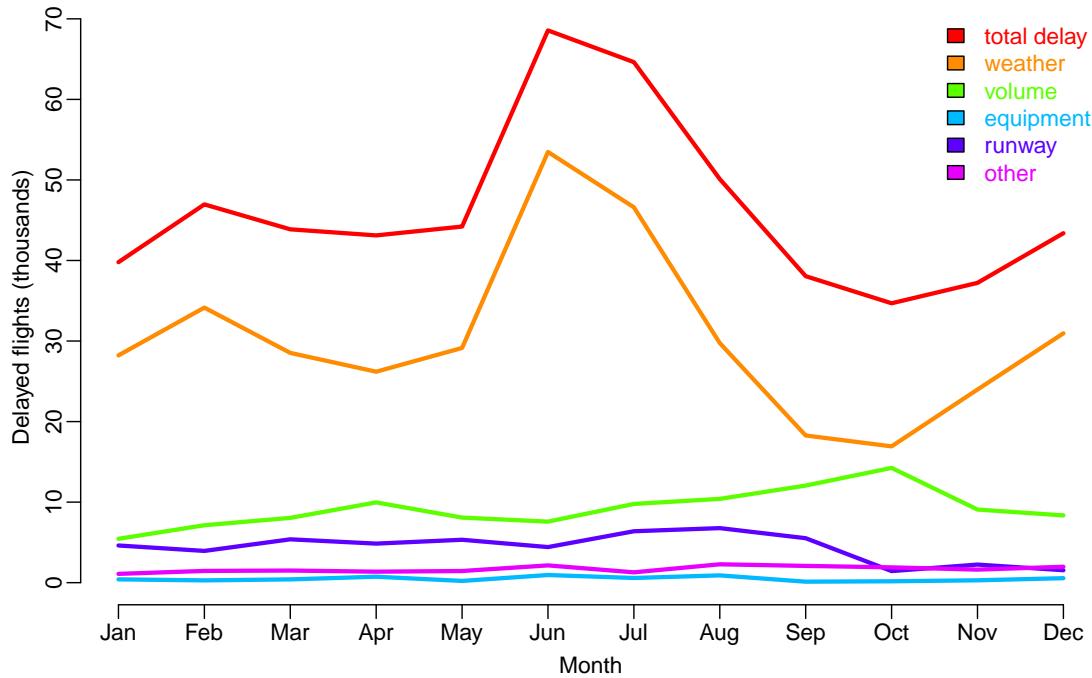


Figure 1-1: Flight delays by cause for 2008.

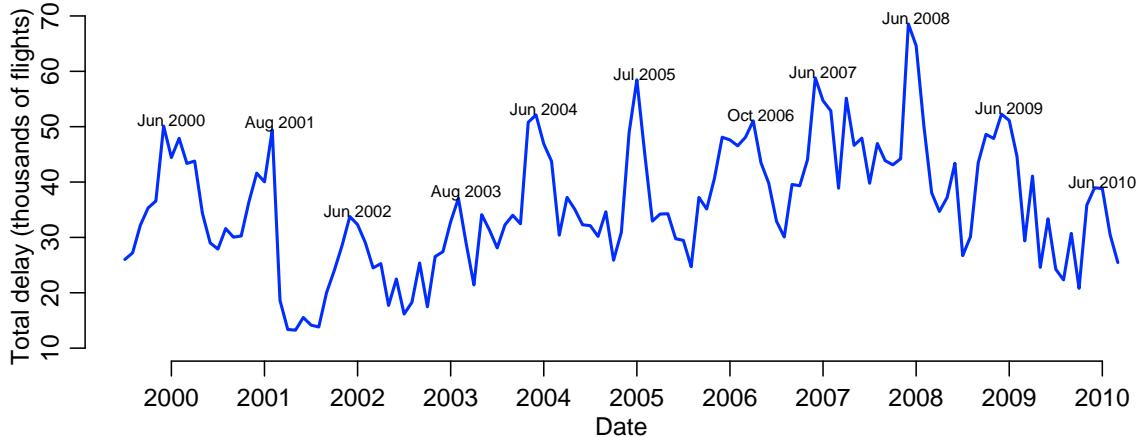


Figure 1-2: Flight delays in the United States over the last 10 years. Source: FAA OPSNET

staggering \$16.7 billion cost to passengers (NEXTOR, 2010). This estimate accounted for passenger time lost due to missed connections, flight cancellations, and delayed flights.

Figure 1-1 shows a time series of the total number of delayed flights in the United States by cause for the year 2008, as reported by the FAA’s OPSNET database. A delayed flight in this dataset is defined as a flight which arrives more than 15 minutes later than its scheduled arrival time. Although OPSNET is known to greatly underestimate the true extent of delay (due to the definition used and the resulting practice of airlines to add large buffers to scheduled arrival times) (El Alj, 2003), the delay trends are nevertheless illuminating. We see that although the causes of air transportation delay include aircraft equipment issues, runway maintenance, traffic volume, security delay, and late passengers, the primary cause of aviation delay has historically been weather. Indeed, in 2008, 45% of total minutes of delay and 66% of delayed flights were weather-related (Bureau of Transportation Statistics, 2010b; OPSNET, 2010). These delays tend to spike in the summer months when travel demand is high and convective weather activity hits congested airports and airspace in the eastern United States, as shown in Figure 1-1 for the year 2008.

Figure 1-2 shows the total number of delayed flights over the last 10 years, and clearly illustrates the consistent spikes in delay during the summer months of each

year. Although weather is inherently chaotic and hence poses a challenge to scheduling, only about 6% of flight delays are due to extreme weather which prevents flying, leaving a large opportunity for improvement (Bureau of Transportation Statistics, 2010b).

With the demand for air traffic operations expected to grow significantly over the next two decades, it has become increasingly important to develop approaches that will enable the efficient operation of the airspace system, even in the presence of convective weather. In the next section, we review the current state of affairs in terms of aviation weather forecasts and traffic flow management techniques that try to decrease delays.

1.2 Background and Related Literature

Although traffic flow management has been an active area of research for decades, activity has recently been invigorated with the FAA's Next Generation Air Transportation System (NextGen) plan, and Europe's corresponding Single European Sky ATM Research (SESAR) program. These programs have highlighted the needs and goals for modernizing airspace and airport operations as aviation demand increases through the years 2025 and 2020, respectively, and are funding research and development in critical areas, including aviation weather and dynamic airspace configuration (Joint Planning and Development Office, 2004; SESAR Consortium, 2006). This section summarizes research relevant to this thesis.

1.2.1 The National Airspace System

This section gives a brief description of the structure and day-to-day operation of the NAS.

At the highest level, the NAS is partitioned into volumes of airspace, each controlled by an Air Route Traffic Control Center (ARTCC, or *center*). The airspace of each center is further subdivided into a set of *sectors*, which are the units of airspace typically controlled by individual air traffic controllers. As aircraft fly between sectors (and centers), control of these aircraft is handed off between controllers. The specific

partition of airspace into a set of sector boundaries is a *sectorization*.

Near major airports, there are additional Terminal Radar Approach Control (TRACON) facilities to control traffic going into and out of the corresponding airport(s). The *terminal-area*, or *terminal airspace*, is the airspace controlled by the TRACON, and is typically divided into a set of sectors, each corresponding to a direction of air traffic (arrival or departure). Each major airport has an air traffic control tower which controls aircraft on the airport surface and in nearby airspace.

The airspace *capacity* of a sector is the number of aircraft that can simultaneously be present in the sector. This number is a function of air traffic controller complexity, and can vary depending on the complexity of flow patterns within the sector or other conditions such as the presence of weather hazards. The Airport Acceptance Rate (*AAR*) is the number of aircraft per unit time (usually an hour) that can be supported by the airport. Under adverse weather conditions, these sector capacities and AARs are lowered, and can be highly uncertain and variable.

Aircraft flying under Instrument Flight Rules (IFR) follow a filed *flight plan*, which is represented by a sequence of *waypoints* (2D points in airspace, sometimes corresponding to a physical navigational aid such as a VORTAC station) connected by airways.

Each flight through the terminal-area must follow a Standard Instrument Departure (*SID*) when departing an airport, and a Standard Terminal Arrival Route (*STAR*) when arriving. These routes are specified by a sequence of waypoints, along with rules governing the speed, heading, and altitude of aircraft at certain waypoints. Waypoints are also referred to as *fixes*. In this thesis, we refer to the fixes at which handoff occurs between the TRACON and Center as the *outer fixes* of the terminal. The term *standard route* refers to either a STAR or SID. The terminal airspace sectorization, as well as all arrival and departure waypoints and routes, are fixed, even when the presence of hazardous weather renders them unusable. An airport has multiple STARS and SIDs, and the assignment of an aircraft to a route is a function of its origin (or destination) airport, aircraft type, runway restrictions, and load balancing of runways.

A commonly-seen terminal-area layout is the *four corner post configuration*, in which airspace is divided into four arrival sectors alternating with four departure sectors, each containing an arrival or departure *gate* along with one or more standard routes (and corresponding outer fixes) per gate. This layout is common for non-metroplex airports in which one airport is the dominant player in surrounding airspace. Figure 1-3 contains a diagram of a terminal-area structured in a four corner post configuration.

The primary function of Air Traffic Control (*ATC*) is to facilitate the safe and efficient operation of the NAS. To ensure safety, ATC is responsible for maintaining the separation of aircraft (through voice radio communication and aircraft position tracking using various automated and manual systems) and providing information to aircraft (including weather conditions and airport conditions). Aircraft that have filed IFR flight plans must obtain approval from ATC throughout their flight. ATC also performs strategic and tactical planning to organize traffic flows so as to manage congestion. The Air Traffic Control System Command Center (*ATCSCC*) located near Washington D.C. manages strategic planning across the entire NAS by coordinating various traffic management interventions when the local Center or TRACON facilities are unable to resolve capacity imbalances. For instance, the ATCSCC may initiate a ground holding program for flights into Chicago if a large Midwestern storm is expected to decrease capacity in the region.

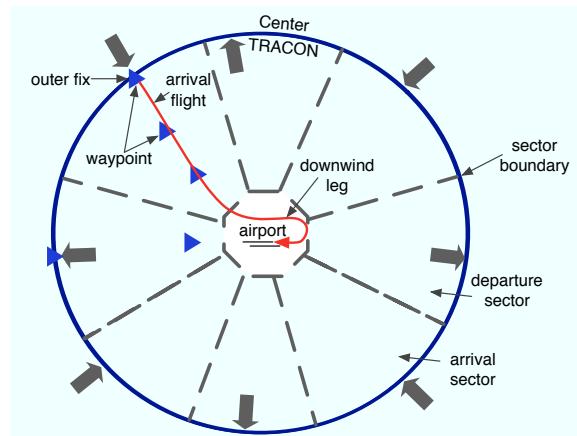


Figure 1-3: Diagram of a terminal-area structured in a four corner post configuration.

Further details about airspace structure and the communication, navigation, and surveillance systems of air traffic control can be found in Belobaba et al. (2005, chap. 13).

1.2.2 Air Traffic Flow Management Models

Air Traffic Flow Management (ATFM) is the process of making strategic decisions a few hours ahead of the time of operations, to balance the demand for aircraft operations with the capacity of the NAS. The capacity of airspace resources is strongly influenced by ambient weather, since aircraft need to avoid hazardous atmospheric conditions and may therefore be forced to deviate from their planned trajectories.

Early research in the field involved large-scale integer programming models which determined how to route a set of aircraft from their planned departure locations to their planned destinations while minimizing the cost of delays. Capacity was the major system constraint and limited how many aircraft could simultaneously occupy a region of airspace. In their seminal paper, Bertsimas and Patterson (2000) were able to solve realistic-sized instances and obtain near-optimal solutions due to the special structure of their formulation, which featured many facet-defining inequalities, resulting in LP relaxation solutions which were often integral. They made the assumption that the impact of weather on the capacity of a resource at any time was known, and used the deterministic estimates of capacity to route flights between their origins and destinations. Eulerian models for this problem which treat the traffic as continuous flows have also been studied, and (Menon et al., 2006; Sun et al., 2006).

However, deterministic capacity estimates based on weather forecasts can be inaccurate under stormy weather conditions. This fact has motivated optimization approaches that assume multiple capacity scenarios for airspace resources, with associated probabilities of occurrence (Bertsimas and Odoni, 1997). Researchers have also modeled uncertainty in capacity as a stationary Markov chain, and developed a stochastic dynamic programming algorithm to select delay-optimal routing strategies (Nilim and Ghaoui, 2004). More recently, robust optimization approaches have been proposed that assume a set of possible capacity values, and try to keep the

system safe for any possible realization of the capacity (Bertsimas et al., 2007).

1.2.3 Dynamic Airspace Configuration

There has also been a body of research whose approach for more efficient management of air traffic has been to analyze and relax the currently fixed and rigid structure of the NAS. The goal for dynamic airspace configuration is to improve access to available airspace and thereby increase achievable capacity.

In the problem of airspace sectorization, researchers have sought to find methods to partition and re-partition airspace in a way that allows for the safe and efficient management of aircraft flows by air traffic controllers (Leiden et al., 2007). Past research has focused on enroute airspace, and has typically modeled the problem as one of partitioning a geometric space subject to convexity, connectivity, and minimum-time-in-sector constraints. The objectives used have served as proxies of overall controller complexity and workloads, and involve balancing sector workload and minimizing inter-sector crossings.

Researchers have used many different solution techniques to solve the resulting NP-hard problem, including genetic algorithms that partition airspace using Voronoi tessellations which are found by successively moving 2D coordinates (Delahaye and Puechmorel, 2006), and mathematical programming formulations that partition 2D airspace into hexagons and then assign the hexagons to a set of sectors (Yousefi, 2005). In yet another approach, Basu et al. (2008) develop a method that recursively partitions a geometric space to build sectors, and mention that the pie-cut has potential for sectorization in the terminal-area. However, to the best of our knowledge, there has not been a focus on either the unique challenges and characteristics of resectorizing the terminal-area, or on the effect of weather on resectorization.

There has also been growing interest in the operational concept of adaptable airspace, which focuses on resolving capacity imbalances by dynamically changing local airspace structure (Kopardekar et al., 2009; Klein et al., 2007). Terminal airspace in particular could benefit from this concept, as standard arrival and departure routes are often shut down when they are affected by adverse weather, resulting in decreased

airspace capacity. There is clear potential to recover lost capacity by dynamically altering the terminal airspace structure in these situations. The concept of flexible terminal-area airspace has been highlighted by the Joint Planning and Development Office (2009) as part of the NextGen ATM-Weather Integration Plan.

1.2.4 Convective Weather Forecasts

Several convective weather forecast products are available for the United States NAS. These forecasts generally take the form of a grid, where each grid cell, or pixel, corresponds to a 2-dimensional section of airspace. Each pixel is associated with a value indicating the severity level of weather at that point.

MIT Lincoln Laboratory's Convective Weather Forecast product is a state-of-the-art 0-2 hour forecast, used throughout the United States to aid air traffic control (Wolfson et al., 2004). The forecast is static, meaning that each pixel contains one deterministic value indicating weather severity, with no additional estimate of the likelihood that the forecast is correct, or a distribution over possible severity. Specific details about the forecast are provided in Section 2.1.

Due to randomness in the weather and the resulting inaccuracy of weather forecasts, creating a plan for routes is not realistic using static forecasts alone. Indeed, flying through a region of airspace that turns out to be stormy could compromise safety. This has led to research into developing probabilistic weather forecasts for aviation. NCWF-2 is one such forecast developed by the National Center for Atmospheric Research, which at each pixel gives a probability p that the pixel will contain convective weather. Initial validation of the forecasts show that these values of p have significant errors associated with them, and tend to be large overestimates of true values (Seseske et al., 2006).

Researchers at the National Oceanic & Atmospheric Administration (NOAA) have developed the Rapid Update Cycle (RUC) weather prediction system, which includes an hourly-updated convective forecast for aviation weather, with a grid resolution of 20 km (Benjamin et al., 2004). The RUC Convective Probability Forecast (RCPF) is built on top of this, providing 3-, 4-, 5-, 6-, 7-, 8-, and 9-hour forecasts of the likelihood

of convective activity within a cuboidal grid volume with 40 km edges (Weygandt et al., 2008).

In addition to these grid-based forecasts, there exist several products which define weather events using polygons. Since these polygons tend to be quite large (on the order of several airspace sectors), these forecasts are targeted at more strategic decision-making.

The Collaborative Convective Forecast Product (CCFP) is one such product, developed by the NOAA Aviation Weather Center (2010). The CCFP predicts polygons of convection over the continental United States at lead times of 2-, 4-, and 6-hours, and is updated every 2 hours. Each of these polygons (typically covering several en-route centers) is associated with a coverage level (sparse, medium, or solid), forecast confidence (low or high), echo tops range, and an indication of growth and direction of movement. The product is created using the RCPF as input, and through a collaboration between multiple stakeholders including the Aviation Weather Center, traffic managers, and airlines. It is used by traffic managers for strategic planning of airspace flow.

Sheth et al. (2006) have proposed a probabilistic weather forecast based on polygons. In their model, a weather cell is represented by a polygon, and the probability that weather will occur at a point in the polygon decreases with increased distance from the center. This structure is then used to estimate flight lengths and deviation delays. However, the model has not been validated against the behavior of actual weather. In addition, for the polygons to have much meaning, they may have to be very large and therefore may not be useful for fine-grained ATFM.

1.2.5 Validation of Aviation Weather Forecasts

Traditionally, entities that develop forecast products have provided users with statistics based on pixel-by-pixel comparisons of the forecast with actual weather. These statistics, used to evaluate the performance of the forecast product, include rates of false positives, false negatives, bias, and skill scores such as the Critical Success Index. The studies authored by Wolfson et al. (2004), Kay et al. (2006), Weygandt

and Benjamin (2004), and Seseske et al. (2006) have conducted historical evaluations of CIWS, CCFP, RCPF, and NCWF-2, respectively. These studies have typically concluded that the weather forecasts show poor skill scores, making it unclear how the forecasts can be reliably used for ATFM.

There have also been efforts to develop more ATFM-based metrics for evaluating convective weather forecasts. These include metrics based on co-occurrence probabilities (Chatterji and Gyarfas, 2006), object-oriented approaches to forecast validation (Brown et al., 2004; Mahoney et al., 2004), and studies that evaluate forecast accuracy as a function of the spatial resolution and storm type (Evans et al., 2009). These studies show that while the accuracy increases at the cost of spatial resolution when compared to pixel-based comparisons, the skill scores still tend to be low and show high variability. Overall, the spatial smoothing of these verification techniques, as well as the forecast errors reported in existing validation studies, makes it difficult to use direct predictions in fine-grained traffic flow algorithms.

1.2.6 Capacity Estimation

ATFM algorithms have typically used airspace capacity as a proxy for weather impact. There have been numerous attempts at creating models of airspace capacity in the presence of convective weather.

Krozel et al. (2007) consider the problem of estimating the capacity of a sector of enroute airspace by computing a theoretical capacity given weather in the region. This is done through the application of continuous maximum flow theory. However, this work relies on static weather forecasts and does not incorporate uncertainty intervals or any measure of forecast accuracy. This line of research is taken a step further by Mitchell et al. (2006). They consider weather forecasts accompanied by a region of uncertainty. However, the uncertainty profiles are randomly generated. Finally, Song et al. (2009) study the correlation between sector throughput and various measures of convective coverage, and conclude that these correlations can be incorporated into an algorithm for sector capacity.

1.2.7 Integration of aviation weather forecasts with ATFM

There has been some recent research attempting to bridge the gap between convective weather forecasts and ATFM algorithms.

Notably, researchers at the MIT Lincoln Laboratory have developed and validated a model of pilot deviation, which predicts, given convective weather and echo tops data, the probability that a pilot will deviate from around a region of airspace (DeLaura et al., 2008). In addition, the Route Availability Planning Tool (RAPT) uses convective weather forecasts to model deterministic departure jet route blockage, and is operational in the New York TRACON (DeLaura and Allan, 2003). Liu et al. (2008) use historical airport arrival rates to create scenario trees of arrival capacities at any given airport. Although these scenario trees result in an improvement over previous models of the ground holding problem, capacities are not tuned to reflect day-of weather information, and validation focuses on SFO, which is most often impacted by fog, as opposed to convective weather.

There has also been work on algorithms to efficiently synthesize routes through regions of airspace affected by convective weather (Prete and Mitchell, 2004; Krozel et al., 2006). This work takes fine-grained and time-varying weather forecast data as static weather input, and focuses on synthesizing short and flyable routes which do not get too close to regions of airspace impacted by weather. However, the weather forecasts are treated as ground truth, and routes are not evaluated against actual weather scenarios.

We conclude that while there has been much prior research on ATFM algorithms that assume accurate convective weather forecasts as input, there has been little work in adapting existing convective forecasts, and in evaluating relevant accuracy and error metrics for use in these applications.

1.3 Contributions of thesis

As highlighted in the previous section, there has been a large disconnect between assumptions made in air traffic flow management algorithms with regards to aviation

weather forecasts, and in the accuracy and usability of these forecasts. This thesis' focus is weather-aware air traffic flow management in airport terminal-areas. There are four main contributions:

- 1 a probabilistic model of route robustness which gives reliable predictions of blockage for terminal arrival and departure routes,
- 2 a mathematical model for dynamic terminal airspace configuration in the face of convective weather which is shown to increase airspace capacity during convective events,
- 3 an approach to the design of realistic conflict-free 3D terminal routes which can help traffic managers plan terminal flow during weather events, and
- 4 validation of all models against actual weather scenarios, avoiding assumptions about the performance of aviation weather forecasts.

These contributions will be briefly described in the following sections. The research presented in this thesis has appeared in Proceedings of the ARAM Special Symposium on Weather-Air Traffic Management Integration, American Meteorological Society 89th Annual Meeting (Michalek and Balakrishnan, 2009a), USA/Europe Air Traffic Management R&D Seminar (Michalek and Balakrishnan, 2009b), Proceedings of the 49th IEEE Conference on Decision and Control (Michalek and Balakrishnan, 2010), and Transportation Science (Pfeil and Balakrishnan, to appear in 2011).

1.3.1 Probabilistic model of route robustness

The first contribution of this thesis is the development of a probabilistic forecast of weather impact, with air traffic flow management in mind. We show that the quality and accuracy of convective forecasts for aviation can be measured in terms of the likelihood that a given trajectory will be blocked by true weather conditions. We consider various features (characteristics) of the forecast weather along arrival and departure routes, and identify features that are highly correlated with route blockage.

Using techniques from machine learning, we propose and validate classification algorithms that predict whether a given route is likely to be open or blocked in actual weather, based on the values of different features of the route as determined by the

forecast. We evaluate our algorithms using several metrics, such as the accuracy (the fraction of time that the prediction is correct), the false positive rate (the fraction of time that we forecast that the route will be open but it ends up being closed), and the false negative rate (the fraction of the time that we forecast that the route will be closed, but instead it remains viable). We convert this binary prediction into a probabilistic forecast by assigning a probability to each prediction, representing the classifier’s probability that the route is open.

The end-result is a route robustness model, which predicts the probability p_r that a given route r will be open (subject to certain assumptions and constraints to be described later) for a fixed horizon.

1.3.2 Model for terminal-area dynamic airspace

The second contribution of this thesis is a model for dynamic terminal airspace configuration in the face of convective weather. We begin with the observation that air traffic control often allows small ad hoc displacements in aircraft trajectories in order to temporarily increase arrival or departure throughput in the face of thunderstorms. Motivated by this practice, we identify and evaluate gentle strategies for re-configuring airspace, without drastically rearranging airspace structure and while limiting disruption to existing air traffic control procedures.

We start with a model in which the previously-developed route robustness forecast guides the selection of fixes that are likely to be open when weather materializes; this selection is traded off against the deviation from the default terminal-area configuration. We show that the recommended changes to airspace structure are robust to changing weather conditions, allowing the model to be integrated into terminal airspace planning.

The simple fix movement model is extended to optimally choose terminal-area arrival and departure fixes as well as sector boundaries, for a given weather forecast, subject to constraints on displacement from today’s fixed airspace structure. We develop an integer programming model to solve this more complex problem, and evaluate the potential increases in terminal throughput if the model were adopted.

Experiments using real weather scenarios on stormy days show that our algorithms recommend that a terminal-area route be modified 30% of the time, opening up 13% more available routes that were forecast blocked during these scenarios. The error rate is low, with only 5% of cases corresponding to a modified route being blocked in reality, while the original route is in fact open. In addition, for routes predicted to be open with probability 0.95 or greater by our method, 96% of these routes (on average over time horizon) are indeed open in the weather that materializes.

1.3.3 Identification of realistic 3D airspace trajectories for tactical planning

The third contribution is an approach based on the A* search algorithm to identify conflict-free 3-dimensional routes through terminal airspace which adhere to realistic airport and airspace constraints, and take airspace demand and weather conditions into account.

Each 3D route identified is associated with an altitude profile (it can be thought of as a 2D cone that grows at increasing distance from the runway) corresponding to the fleet mix along the route. All routes (and their altitude profiles) are separated both horizontally and vertically, and are flyable in the sense that they tend to be straight except for a few smooth, wide-angle turns. The algorithm can also be used to model noise restrictions in the form of altitude minima, as well as metroplex constraints in the form of obstacles corresponding to location of nearby STAR and SID flows.

The approach can be integrated into a tactical decision support tool for air traffic control as follows. Up to 90 minutes ahead of operations at an airport operating at a given airspace configuration, the standard 3D arrival and departure routes are evaluated against the weather forecast. For each route which is predicted to be blocked, a conflict-free route between the airport runway and a terminal fix (near the original) are identified using the A*-based shortest path algorithm. Controllers can then shift traffic to these new available routes.

1.3.4 Validation with actual weather conditions

The final contribution of this thesis is the emphasis on validation of models against actual weather scenarios throughout the development and evaluation process. We use weather data from Hartfield-Jackson Atlanta International Airport (ATL) terminal-area for the 2007 and 2008 convective seasons as a case study for all models mentioned. This is a departure from much of the previous research in air traffic flow management.

1.4 Organization of thesis

This thesis is organized as follows. Chapter 2 describes the Lincoln Lab convective weather forecast product, discusses issues with using deterministic pixel-based forecasts for ATFM, and shows how a route-based approach can be used to evaluate convective forecasts. Chapter 3 develops and evaluates the data-driven route robustness model. Chapter 4 introduces a simple model for terminal dynamic airspace configuration and shows how to implement fix changes in a dynamic environment, and then extends the model with an integer programming formulation. Chapter 5 proposes a more realistic model of terminal routes that incorporates demand for a route, physical aircraft and airport constraints, 3D aircraft flow with realistic climb and descent rates, as well as weather forecasts. An algorithm is developed to identify airspace routes subject to these constraints. Finally, Chapter 6 concludes the thesis with a summary and discussion.

Chapter 2

Convective weather forecasts for traffic flow management

This chapter introduces a state-of-the-art convective weather forecast developed by MIT Lincoln Laboratory. We discuss the limitations in forecast accuracy using traditional metrics for forecast evaluation, and show how these methods are not entirely useful from a TFM perspective. We then introduce a route-based approach to viewing and evaluating weather forecasts, and show how the resulting forecasts post much better skill scores, while being useful for TFM.

2.1 Lincoln Lab convective weather forecast

MIT Lincoln Laboratory has developed the Corridor Integrated Weather System (CIWS), which consists of two weather forecast products with time horizons of 0-2 hours, for a grid of $1 \text{ km} \times 1 \text{ km}$ pixels covering (in 2D) a large portion of the NAS (Wolfson et al., 2004).

The first product, depicted in Figure 2-1, consists of predicted values of Vertically Integrated Liquid (VIL) at each pixel. These VIL values are integers in the range $[0, 255]$, which represent the amount of liquid at a point in the sky integrated vertically. For ease of use, VIL values can be mapped (nonlinearly) into seven levels of convective activity, ranging from level 0 (no activity) to level 6 (very severe). A VIL value

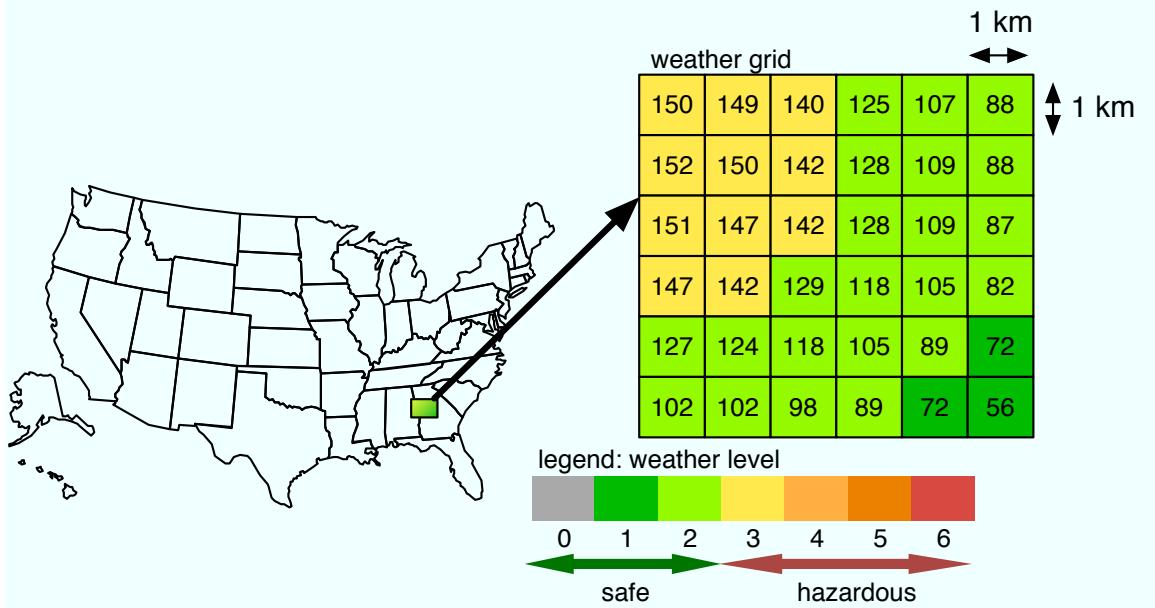


Figure 2-1: Diagram of MIT Lincoln Lab’s CIWS VIL forecast near ATL

above a certain threshold in the observed data (133, in practice) corresponds to weather of severity level 3 or higher, and is commonly considered by pilots to be hazardous (Krozel et al., 2006). Although we use this value as a strict limit to define hazardous weather in this research, the threshold of what pilots will fly through varies depending on numerous factors including airspace demand and proximity to destination airport¹. Forecasts are issued for horizons spanning between 5 minutes and 120 minutes in 5-minute increments, and are updated every 5 minutes. In other words, at time T_0 , forecasts are available for time $T_0 + 5, T_0 + 10, T_0 + 15, \dots, T_0 + 120$. Along with the archived forecast data we also obtained the observed VIL values for the same region of airspace at that time, allowing for validation of the quality of the forecast.

The second CIWS product is a forecast of echo tops (height at top of storm) at the same grid resolution and update times. Since the focus of our research is the airport terminal- area, where storm height is not a significant factor for pilot deviations, we

¹Pilot deviation in the terminal-area is not as well understood by researchers as in enroute airspace, and level 3+ weather may be just one predictor for deviation among other factors such as demand. The willingness of the pilot to fly through Level 3+ weather has been observed in terminal operations

do not discuss this product further.

CIWS gives users a general idea of what weather will look like, and is used in various decision support tools by air traffic controllers and airlines. Lincoln Lab, as well as other forecast providers, track daily statistics such as rates of false positives, false negatives, and skill scores, but these vary daily and by storm.

2.2 Pixel-based evaluation of CIWS forecast

In this section, we evaluate CIWS accuracy by computing several standard statistics on a set of the stormiest weather scenarios over ATL from the summer of 2008, which we refer to as WeatherScenarios2008.². The results highlight some of the issues associated with integrating convective weather forecasts with TFM.

2.2.1 Evaluation of skill scores

We first define a few verification statistics that have traditionally been used to evaluate weather forecasts. Consider a pixel for which a forecast and actual weather data exist. A true positive (TP) is when the forecast and actual weather both show hazardous weather at the pixel. A true negative (TN) is when the forecast and actual both show no hazardous weather. A false positive (FP) is when the forecast predicts hazardous weather, but the pixel is clear in actual weather, and the opposite is the case in a false negative (FN). Using these base statistics, we can form four standard measures of forecast skill: the Critical Success Index (CSI) is defined to be $\frac{TP}{TP+FP+FN}$, the probability of detection (POD) is defined to be $\frac{TP}{TP+FN}$, the false alarm rate (FAR) is defined to be $\frac{FP}{TP+FP}$, and accuracy is defined to be $\frac{TP+TN}{TP+TN+FP+FN}$.

Figure 2-2 shows the performance of the forecast in terms of CSI, POD, FAR, and accuracy, for a range of time horizons. To evaluate the CIWS VIL forecast, we partition VIL into two sets: positive (weather level 3 or higher, corresponding to hazardous weather), and negative (weather level 2 or lower, corresponding to non-hazardous conditions).

Accuracy posts the best performance, with scores above 0.9 for each time horizon.

²Weather scenario selection is described in detail in Section 2.5.1

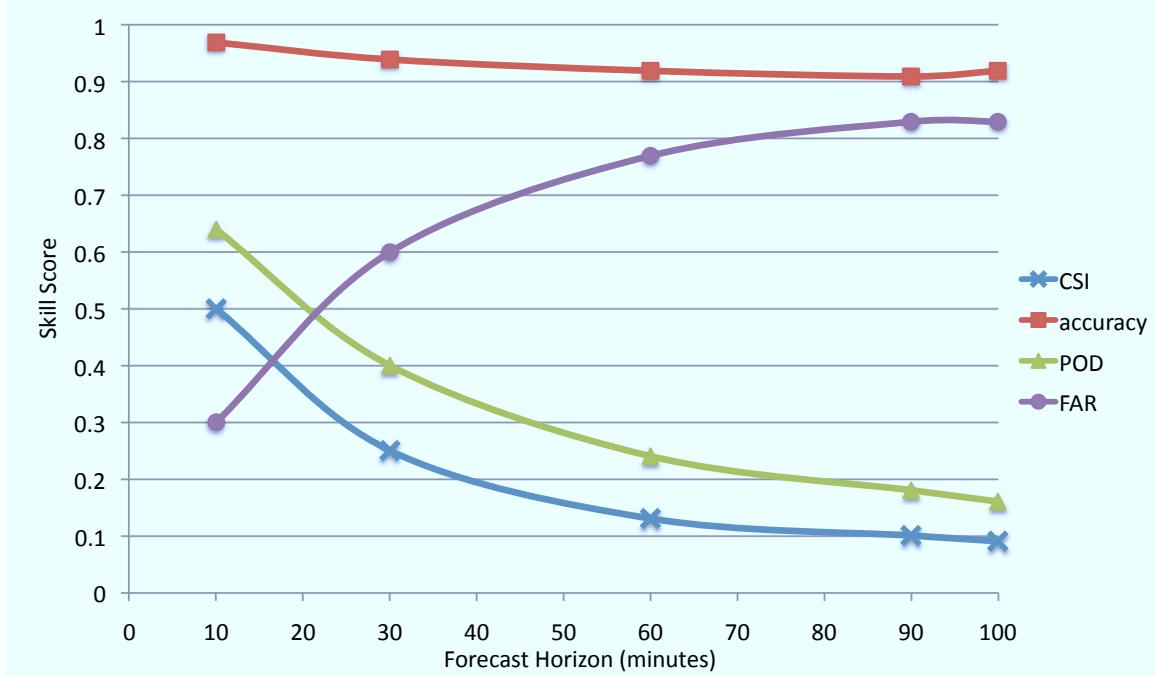


Figure 2-2: Four skill scores for the CIWS weather forecast, at increasing time horizons.

However, this high score merely reflects the high proportion of level 0 pixels present in the weather data (most pixels do not contain weather activity), resulting in a strong majority of true negatives. The other three statistics, which do not include TN and are therefore more relevant for assessing the quality of the forecast, perform very poorly. CSI is 0.5 and below across time horizons, which means that more than half of the time that either the forecast or actual pixel contains weather, the forecast is erroneous. The POD and FAR are similarly low, especially for time horizons of 30-min and higher.

2.2.2 Distribution of actual weather given forecast

Although the forecast skill scores were low at longer time horizons, this does not paint the entire picture of how well the forecast predicts actual VIL. Weather is by nature stochastic, so a deterministic forecast cannot be expected to capture the underlying distribution of true VIL. In this section, we compute the distribution of actual VIL given a forecast, and evaluate and draw further conclusions about forecast

performance.

Let $w(x, t)$ be the observed weather for pixel x at time t . Let $f_\tau(x, t)$ be the τ -minute weather forecast for pixel x at time t . In other words, $f_\tau(x, t)$ is the forecast created at time $t - \tau$, for time t . We would like to evaluate the conditional distribution $\Pr(w(x, t) = v | f_\tau(x, t) = u)$. Note that this distribution is likely to be independent of the specific pixel x or the specific time t , but might depend on the general geographical area or the time of day.

Figure 2-3 shows histograms of VIL that actually occurred given a forecast VIL of level 3, for time horizons $\tau = 30$ (left) and $\tau = 60$ minutes (right). The histograms reflect data from the same set of 2008 weather scenarios near ATL as in the previous section. Although both plots look roughly Gaussian, the mean VIL is level 2, and the mode is level 0 for each. This indicates over-prediction of weather severity, and moreover, when hazardous weather is predicted at a pixel, on average that pixel turns out to be safe to fly through in true weather. Finally, the heavier peak around VIL values of 125 for $\tau = 30$ when compared against $\tau = 60$ confirms higher prediction accuracy at shorter time horizons.

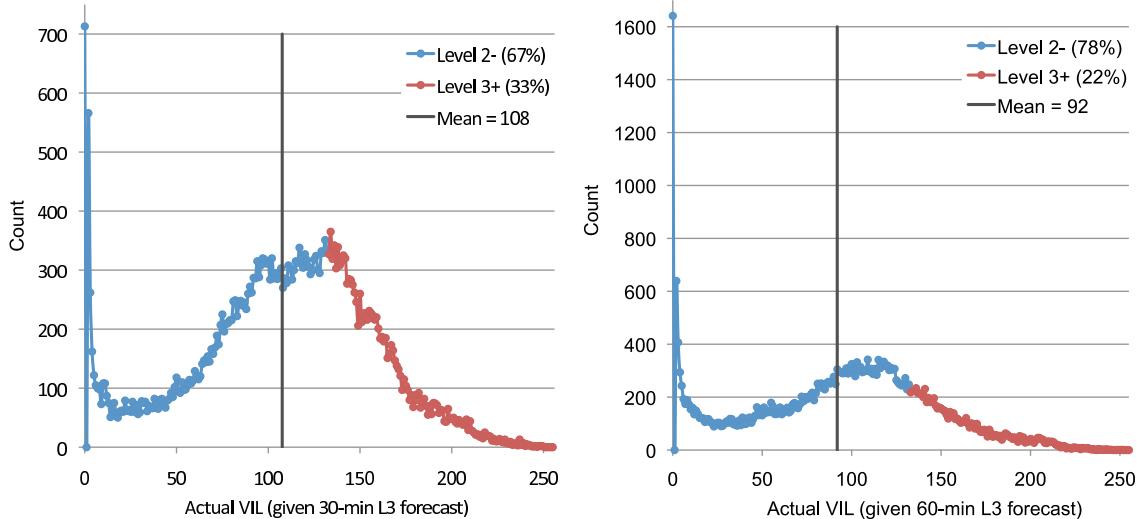


Figure 2-3: Histogram of true VIL when Level 3 VIL (in the range [133, 162]) is forecast, for 30-min (left) and 60-min (right) horizons.

These histograms suggest the possibility that the CIWS forecast gets the general

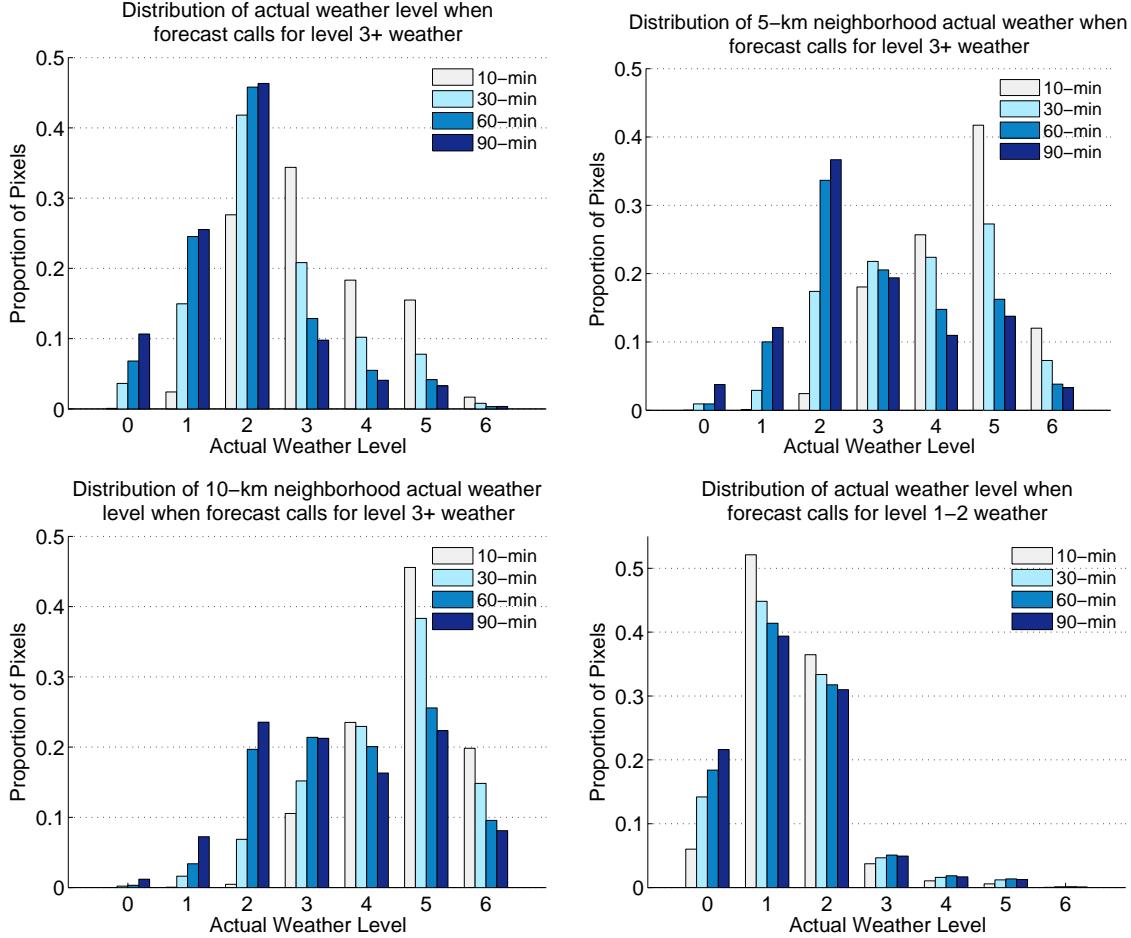


Figure 2-4: Distribution of true weather level given a forecast at a pixel, for varying time horizons across a large set of summer 2008 weather scenarios.

weather trends right, but is incorrect in the spatial position of weather cells. This spatial error in the distribution of true VIL is further explored in Figure 2-4, which shows the distribution of actual weather level given a predicted weather level across time horizons. The top-left plot shows the distribution of actual weather level when level 3 or higher (L3+, from now on) is forecast. For time horizons of 30-min and longer, the distribution reflects the same over-forecast of hazardous weather as seen in the previous section, where true weather is level 2 and lower at the majority of pixels.

However, when the definition of true positive is relaxed so that the presence of L3+ weather within a B km neighborhood of the forecast pixel counts as a correct

prediction, this distribution shifts right, with a majority of pixels at L3+ (as illustrated in the top right histogram with $B = 5$, and the bottom left with $B = 10$). This confirms the presence of spatial error in the forecast, though the value of B that makes this distribution shift varies, and can be too large to be useful for fine-grained planning of flight plans. Finally, the bottom right plot shows the distribution of true weather when level 1 or 2 is forecast. Note that we leave out level 0, because it represents a majority of cases, and would completely shift the distribution to the left. This plot indicates that either the forecast performs well on predicting lack of hazard, or that the true weather is level 2 or lower in the vast majority of cases. Either way, this contributes to the strong TN score.

This section validated CIWS by evaluating forecasts of individual pixels, and found large forecast errors and a tendency to over-predict weather impacts. These findings match evaluations of other forecast products. We next argue that despite these findings, when we move our perspective away from individual pixels and view forecasts in terms of entire routes with a tolerance for spatial error, CIWS may be very useful (and accurate) for air traffic planning.

2.3 Forecast objectives from an operational perspective

The evaluation of a weather forecast by comparing the predicted and true weather at individual pixels does not capture several operational realities of traffic flow planning.

First, it is possible that the forecast gets the general weather trends right, but is incorrect in the exact position of the weather cells. This phenomenon is illustrated with the scenario in Figure 2-5. A storm cell is forecast to hit 10 km north of a filed flight plan. When weather materializes, the actual storm is displaced such that it lies 10 km to the south of the filed flight plan, resulting in very low skill scores: Critical Success Index and Probability of Detection are each 0, and False Alarm Rate is 1. Despite these poor skill scores, this forecast is actually quite good for planning purposes, since a planned trajectory could easily be moved 10 km north. Moreover,

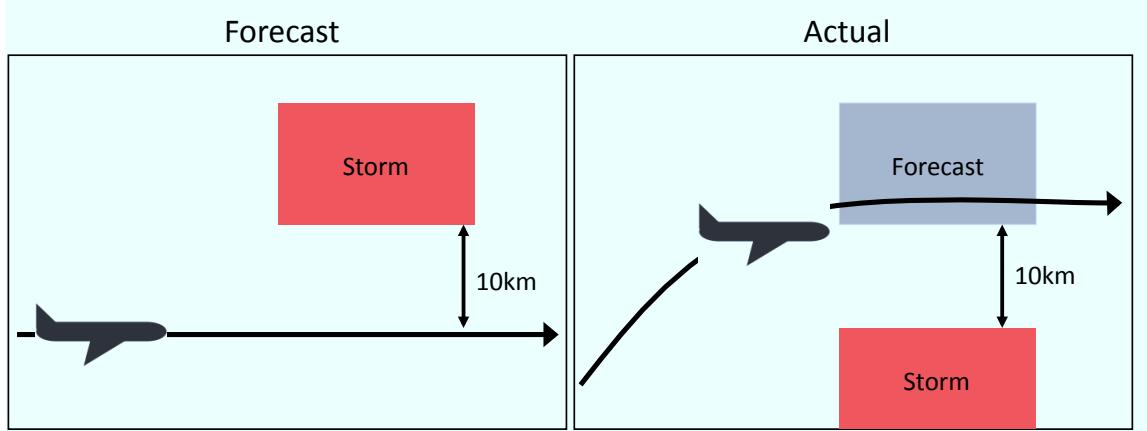


Figure 2-5: Illustration of potential advantages of route-based forecast evaluation over a pixel-based skill scores.

any north-south operations would be unaffected because there is no lateral error in the prediction.

Second, predicting the storm type (regardless of precise location of weather cells) can be very useful for TFM. For example, a storm consisting of many small sparsely distributed cells of weather activity (a popcorn storm) may have low forecast accuracy in pixel-by-pixel comparisons, and yet have many available routes between cells, resulting in no practical capacity reduction.

Thirdly, a forecast with reasonable spatial accuracy (with some error at individual pixels) can be very useful for planning. Indeed, knowing that there is a 30% chance of rain into Boston today, for instance, does not help to determine if there will be a route open from the east into Boston Logan Airport at 5PM, or if flights should incur delay on the ground and avoid entrance into the Boston area between 5 and 6 PM. However, a forecast which predicts weather in a region generally west of the airport may be enough to correctly manage west-bound flows.

These operational realities suggest that a route-centric approach may be a better way to evaluate weather forecasts used for TFM. Identifying persistent routes through weather might identify opportunities for increased capacity even in the presence of storms and of inaccurate forecasts. We next develop a terminal airspace model for evaluating (and using) forecasts in this way.

2.4 Model for a route-based forecast

Motivated by the previous analysis, we would like to study the following problem:

Given a weather forecast for some time in the future and a set of predetermined potential routes, we would like to best identify those routes that are likely to be open in the actual weather that materializes, and also to quantify the uncertainty associated with our prediction mechanism.

2.4.1 Definition of open route

In order to solve this problem, we adopt the following definition for an open route.

A route is defined to be **open** or **clear** in the observed weather if there exists a route within a B km neighborhood of the original route that is not impacted by weather. This potentially displaced route is called the **perturbed route**. Note that the perturbed route may be identical to the original one.

This relaxed definition allows for slight deviations in a planned route that reflect the “wiggle room”, or the ability of an aircraft to make small adjustments to the planned route. The parameter B can be adjusted to reflect operational constraints for a particular terminal-area and the desired level of flexibility, although B will typically be small (between 5 and 10 km).

2.4.2 Terminal airspace setup

We now describe the terminal airspace model that will be used throughout this thesis. It is depicted in Figure 2-6(a).

The input is a terminal-area, defined by two concentric circles: an **outer circle** C_O of radius R_O , and an **inner circle** C_I of radius R_I . Aircraft flying under instrument flight rules currently follow their filed flight plans which are represented by standard waypoints connected by airways. Aircraft flows from and to the airport are typically routed through specific way-points on the outer circle known as **fixes**, which are points of entry into or exit out of the terminal-airspace. The circle C_O represents the points at which arriving aircraft first enter the terminal airspace, while

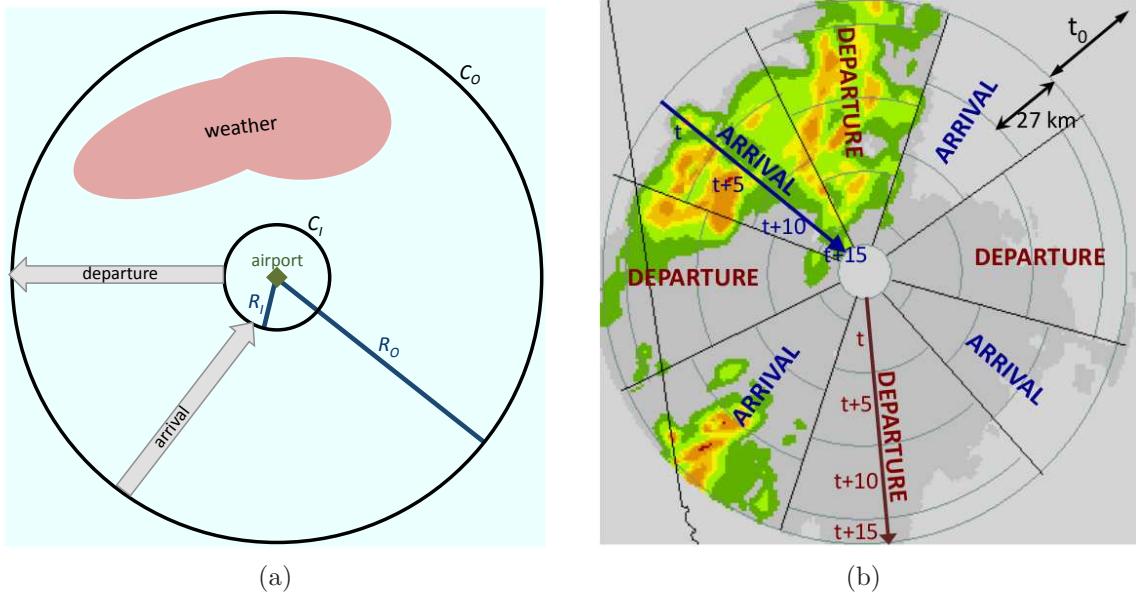


Figure 2-6: Diagram of terminal airspace: (a) depicts the set-up and (b) shows the dynamic weather grid.

C_I represents the point at which aircraft begin their landing procedure (for example, the crosswind leg of the approach) into the airport. In contrast, departures traverse the terminal-area in the reverse direction, entering it close to the airport at C_I and exiting it through the outer boundary C_O . The parameters R_I and R_O are adjustable depending on the characteristics of an individual airport. R_I typically ranges between 10 km and 30 km, while R_O is roughly 100 km.

2.4.3 Dynamic weather grid

Before we can superimpose a weather forecast over the terminal model just described, we must incorporate a notion of time into the model. In particular, we would like to account for the movement of aircraft through airspace and have the forecast at a particular location (pixel) correspond to the time when the aircraft flies through it.

In order to model this aircraft movement, we construct a **dynamic weather grid** by splicing together weather data for consecutive time intervals. This grid is depicted in Figure 2-6(b) using a real weather scenario. The grid is divided into eight sectors alternating between arrivals and departures, which have different dynamic grids. We assume aircraft arrive at C_O at time t . Planning occurs t_0 -minutes in

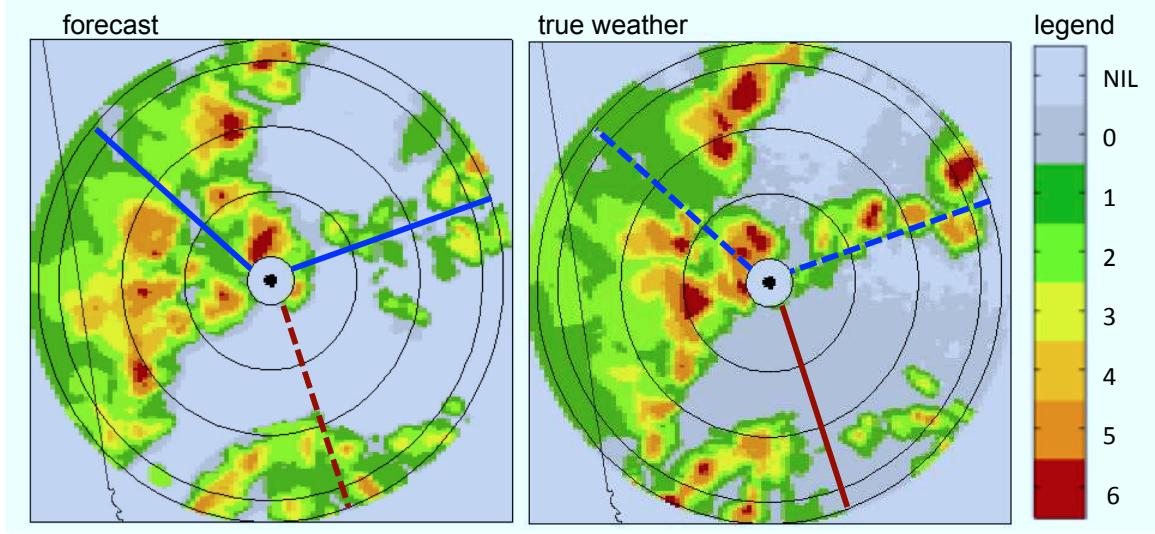


Figure 2-7: An example of departure routes through a dynamic weather grid.

advance of aircraft arrival. For departures, the corresponding dynamic grid assumes aircraft arrive at C_I at time t , with the same t_0 -minute planning horizon. This grid will therefore be used for planning at the current time, namely, $(t - t_0)$.

The distance between two concentric circles in the grid corresponds to the distance flown in 5 min by a typical aircraft. These circles are drawn assuming an average aircraft speed of 180 knots in the terminal-area; we have also conducted similar analysis for aircraft speeds of 85 knots, corresponding to slower, general aviation aircraft. (Michalek and Balakrishnan, 2009a).

Figure 2-7 provides an illustrative example with three departure routes overlaid on a dynamic weather forecast on the left, and the same routes overlaid on the observed dynamic weather for that scenario on the right. In this example, the planning horizon of interest is $t_0 = 30$ minutes. At the current time $(t - t_0)$, we are interested in predicting whether potential routes for aircraft that are currently 30 min from the terminal-area entrance (C_I for departures) will be open. This would allow us to provide recommendations on which terminal-area route to fly through. In order to do this, we use the 30 min forecast in the innermost annulus of the dynamic forecast grid, the 35 min forecast for the next annulus, the 40 min forecast for the next one, and so on. Similarly, for validating the performance of this forecast along this route,

we will consider the observed weather along the route in each annulus at the time that the aircraft flies through it, i.e., the observations 30 min from the current time for the innermost annulus, 35 min from the current time for the next annulus, and so on.

Each potential departure route is a path between a departure fix on C_O and a point on C_I . In this example, the routes are examples of forecast inaccuracy. Two of the three routes (denoted by solid blue lines) are predicted to be open but are blocked by weather in reality, while the third (denoted by a dotted red line) is forecast to be blocked, but is open in the weather that actually materializes. Although this example focuses on departures, the weather grid is simply inverted for arrivals: the outermost annulus would contain the 30-min forecast, etc.

All weather scenarios considered in the remainder of this thesis use this dynamic weather grid.

2.4.4 Identifying robust routes through terminal airspace

We can now restate our problem in more concrete terms:

If an aircraft is routed along a trajectory between C_O and C_I , and given a t -minute weather forecast through the corresponding dynamic weather grid, what is the probability that that this trajectory will be open in the weather that actually materializes?

2.5 Generation of data sets

This thesis uses weather data for Hartsfield Atlanta International airport (ATL) terminal-area extensively. This section describes the selection of weather scenarios, the selection of potential arrival and departure routes within each scenario, and the initial validation of these routes in the forecast grid against perturbed routes in the true weather grid.

2.5.1 Selection of weather scenarios

We focus on the terminal airspace of ATL, which is the busiest airport in the world in terms of total aircraft operations, and experiences significant delays due to con-

vective weather. ATL is also chosen as a case study due to its standard corner post configuration, and the fact that Atlanta is inland, and avoids some additional forecast inaccuracies (which have been observed anecdotally by forecast providers) due to the effect of the ocean on weather patterns.

CIWS data for the summer 2007 and 2008 convective seasons was provided by MIT Lincoln Lab. Each day of CIWS data yields approximately 30 GB of uncompressed binary data, hence weather scenarios were selected using only a partially-automated process. The selection process began by using FAA OPSNET data to determine the most weather-impacted days when ranked according to weather-related delays, during the months of June and July 2007, and June through August 2008. Once a set of dates were narrowed down, ATL terminal-area weather was automatically extracted, the number of level 3+ weather pixels in the true weather grids were computed across entire days, and the periods with significant convective weather activity (as judged by having more than a human-selected threshold of hazardous weather pixels) were visualized using Matlab scripts. A human then selected roughly 4 weather scenarios separated by at least 30 minutes for each high-activity day, trying to include different type of weather situations (developing storm, well-developed storm, popcorn storm, line squall, etc.).

The final set of weather scenarios were partitioned by year, and are referred to WeatherScenarios2007 and WeatherScenarios2008 for the remainder of this thesis. The selected dates are specified in Table 2.1. Each weather scenario actually corresponds to 10 subsets of data, corresponding to planning horizons t_0 of 10-, 30-, 60-, 90- and 100-minutes, for both departures and arrivals. For the remainder of this thesis, we refer to a weather scenario as a date and time t , a planning horizon t_0 , and a direction (arrival or departure).

2.5.2 Selection of routes

Now that a set of weather scenarios have been selected, the next step is to select routes through these scenarios. We can then study how well CIWS predicts route blockage on this set of route, and later improve upon this prediction.

Weather Scenarios 2007		Weather Scenarios 2008	
dates	times (GMT)	dates	times (GMT)
Jun 5	17:30 18:00 18:30 19:00	Jun 1	12:00 14:00 20:45
Jun 8	19:30 20:00 20:30 21:00 21:30	Jun 3	18:15 19:45
Jun 12	5:30 6:00 6:30 7:00	Jul 8	20:00 21:00 22:00 22:45
Jun 14	21:15 21:45	Jul 10	16:30 18:30 19:30 20:30 21:30
Jun 15	18:30 19:00 19:30 20:00	Jul 13	13:30 14:00 15:30 17:30 19:30
Jun 19	19:00 19:30 20:00 20:30	Jul 31	16:15 17:00 18:00 19:00
Jun 25	19:00 20:00 20:30 21:00 21:30	Aug 26	03:00 09:15 11:00 12:30 19:00
Jun 28	22:30 23:00		
Jun 29	21:00 22:30 23:00		
Jul 1	21:00 21:30 22:00 22:30 23:00		
Jul 11	16:00 16:30 17:00 17:30 18:00		
Jul 19	16:00 16:30 17:00 17:30		
Jul 29	19:30 20:00 20:30 21:00		

Table 2.1: Weather scenarios considered during 2007 and 2008 convective seasons.

Routes through forecast weather

Potential aircraft trajectories through each weather scenario are generated by simply sampling eight straight routes between C_0 and C_I , as depicted in Figure 2-8. These eight trajectories represent a random sample of routes through varying weather forecasts and flight orientations.

Routes through true weather

Each route r generated in the manner described above is evaluated using the observed weather data. Recall that r is *open* if there exists a corresponding *perturbed route* r' in the observed weather grid within B km of r . Recall that r' cannot pass through any hazardous weather, and that this B km neighborhood allows for slight perturbations in the original route (on the order of several kilometers).

Open routes are synthesized by solving a shortest-path problem with turn penalties through the dynamic grid of observed weather, modeled as an integer program (IP). Note that although the shortest path problem with turn penalties is known to be solvable in polynomial time (Ahuja et al., 1993), we use an IP approach because it allows for simple and useful augmentations to the formulation. For example, adding a constraint to penalize the deviation of the solution from the original route or to

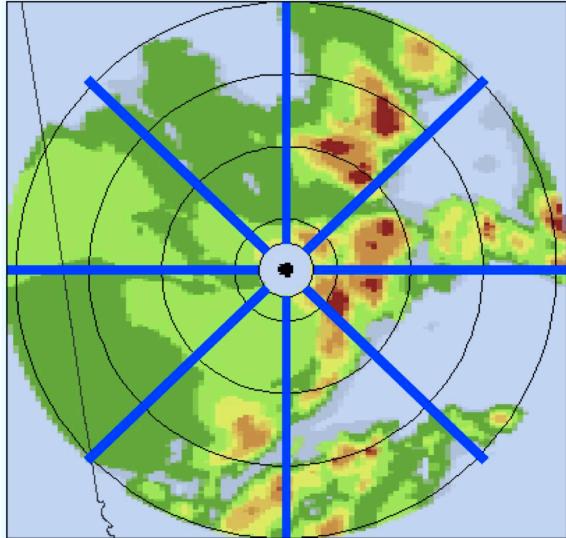


Figure 2-8: Illustration of the eight routes sampled for an arrival forecast scenario.

penalize routes which are too close to L3+ weather, gives better perturbed routes for visualization.

The formulation is as follows. A directed graph $G(\mathcal{N}, \mathcal{A})$ is constructed such that the set of nodes \mathcal{N} contains all pixels within B km of r (in the dynamic observed weather grid) which are free of weather hazards, and such that each set of adjacent nodes forms an arc $a \in \mathcal{A}$. At time t , a unit of flow is sent from a set of source nodes $\mathcal{S} = C_O \cap \mathcal{N}$ (the subset of nodes lying on the outer circle C_O) to a set of sink nodes $\mathcal{T} = C_I \cap \mathcal{N}$. For simplicity, we use a standard transformation and introduce a supersource $\bar{\mathcal{S}}$ and a supersink $\bar{\mathcal{T}}$, and route one unit of flow between the two through the source nodes and sink nodes (Ahuja et al., 1993). To model turn penalties, $\text{NX}(i, j) \in \mathcal{N}$ is defined to be the node which constitutes a straight next arc if (i, j) is used. In other words, nodes i, j , and $\text{NX}(i, j)$ form a straight line in the observed weather grid. Since we would like to recommend a route which, in addition to being short, requires a minimum amount of maneuvering on the part of pilots, the objective is to find the minimum cost flow f such that out of all minimum cost flows,

f has the minimum number of turns.

$$x_{ij} := \text{flow on arc } (i, j) \in \mathcal{A}$$

$$z_{ij} := 1 \text{ if } (i, j) \in \mathcal{A} \text{ is a turn, } 0 \text{ otherwise.}$$

$$\begin{aligned} & \min \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} + \lambda \sum_{(i,j) \in \mathcal{A}} z_{ij} \\ \text{s.t. } & \sum_{\substack{j \in \mathcal{N}: \\ (i,j) \in \mathcal{A}}} x_{ij} - \sum_{\substack{j \in \mathcal{N}: \\ (j,i) \in \mathcal{A}}} x_{ji} = b_i \quad \forall i \in \mathcal{N} \end{aligned} \quad (2.1)$$

$$z_{ij} \geq x_{ij} - \sum_{\substack{k \in NX(i,j): \\ (j,k) \in \mathcal{A}}} x_{jk} \quad \forall (i,j) \in \mathcal{A} \quad (2.2)$$

$$x, z \in \{0, 1\}^n \quad (2.3)$$

Constraints 2.1 are the flow balance constraints, with $b_i := -1$ for a supersource $\bar{\mathcal{S}}$, $b_i := +1$ for a supersink $\bar{\mathcal{T}}$, and $b_i := 0$ for all other nodes i in \mathcal{N} . Constraints 2.2, in conjunction with the penalty term in the objective function, serve to minimize the number of turns in the path without changing the path length, since it is desirable that aircraft trajectories have a limited number of turns for simplicity. All arcs that follow (i, j) , *except* (j, k) for $k = NX(i, j)$, pay a penalty in the objective function. λ is chosen to be sufficiently small (less than the maximum length of any path) to eliminate longer routes with fewer turns. Finally, x and z are restricted to binary variables in 2.3, to ensure that flow is not split up. Note that although this formulation models the case of arrivals, the same IP can be used to model departures by replacing the underlying dynamic grid.

This problem is solved for each of the selected routes in each data set; the infeasibility of the problem implies that the route is blocked in the observed weather grid, while feasibility implies that the route is considered open. A version of this problem can also be solved with different sets of sources and sinks and with $B = \infty$ to generate a large set of candidate routes for a given weather forecast scenario (Michalek and Balakrishnan, 2009a).

2.5.3 Details of route dataset

Once a set of routes is synthesized for each weather scenario, we can analyze the resulting dataset. We have partitioned the weather scenarios by year so that one set can be used for testing, and the other independent set can be used for training, later in the thesis. We refer to these two sets as Data2007 and Data2008. In this section we show how viewing forecast performance through the lens of routes (instead of pixels) with some allowed wiggle room can significantly improve the apparent forecast accuracy.

We begin with a visualization of the resulting routes. Figure 2-9 shows examples of several synthesized routes. Each pair of images corresponds to a single weather scenario, with a sample route in a dynamic forecast grid on the left, and the corresponding perturbed route in the actual weather on the right, if it exists. The topmost weather scenario is an arrival at the 60-minute planning horizon on 070612 at 0630Z, and depicts a route that is open according to the forecast and ends up being open in actual weather. The scenario in the center corresponds to 070619 at 1900Z, and shows an arrival route at the 90-minute planning horizon. The exact original route is blocked according to the forecast, but a nearby perturbed route is available in the true weather grid. The bottom scenario is a departure route on 070608 at 2030Z at the 30-minute horizon. In this situation, the forecast route is not open in the observed weather grid.

Table 2.2 gives a summary of the overall statistics for Data2007 for arrivals and departures at the five planning horizons studied. Although the parameters B , R_I , and R_O are configurable, the table data corresponds to a wiggle room of $B = 8$ km, inner radius $R_I = 10$ km, and outer radius $R_O = 100$ km. Each route is evaluated using the forecasts and observed weather appropriate for the times at which the affected aircraft will traverse the route, as described in Section 2.4.3.

As indicated by the table, each dataset consists of 408 routes, the majority of which are open. The percentages of routes that are forecast open (i.e., routes that do not pass through Level 3+ weather in the forecast) are between 48% and 63% for

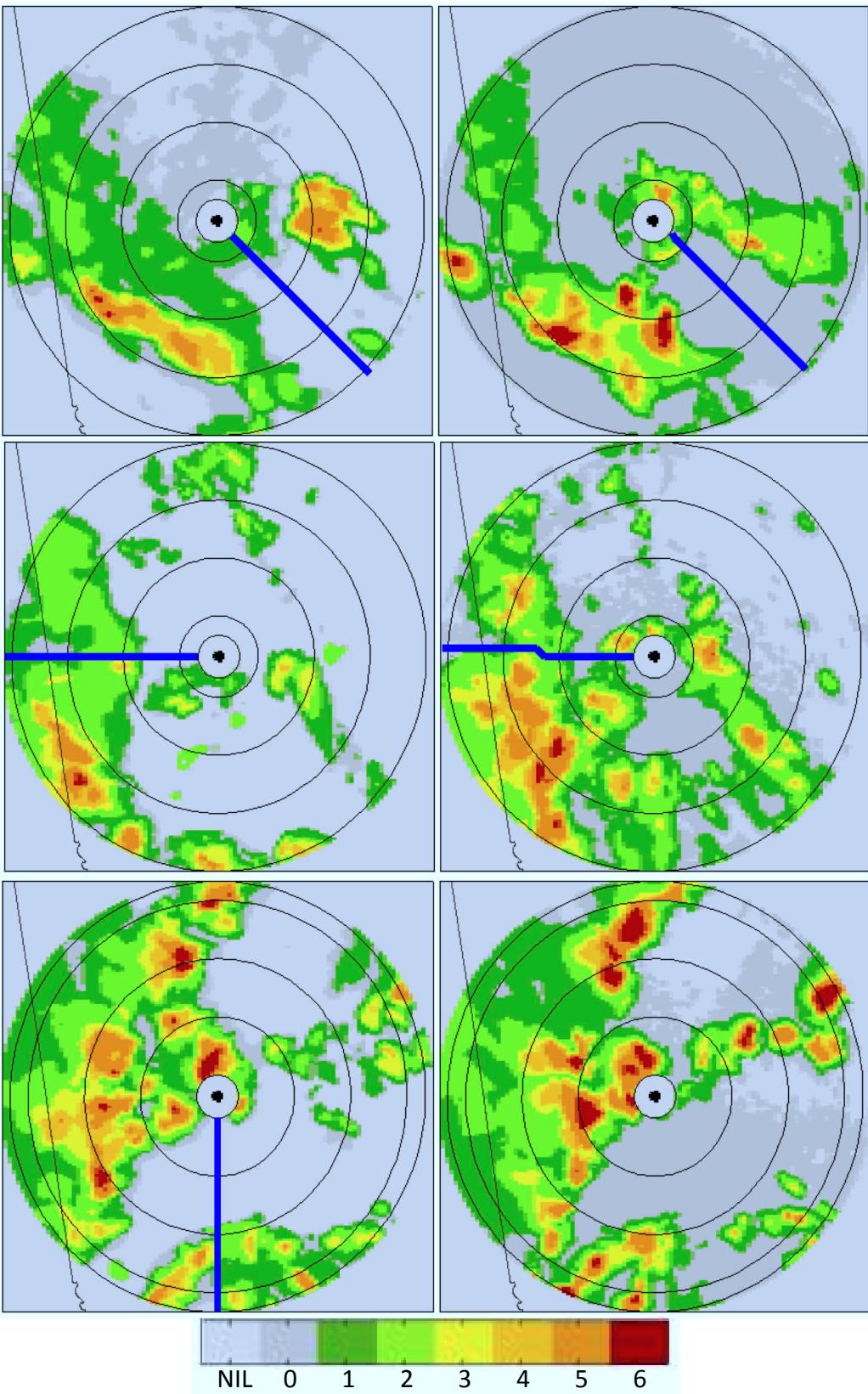


Figure 2-9: Examples of routes synthesized in the forecast grid (left), and validated against the observed weather (right).

dir	t_0	# routes	forecast open (%)	actual open (%)	actual open given forecast open (%)	actual closed given forecast closed (%)
arrivals	10	408	48	74	98	52
	30	408	47	74	94	57
	60	408	48	74	86	64
	90	408	57	74	86	58
	100	408	62	74	84	59
departures	10	408	50	76	99	54
	30	408	50	76	94	59
	60	408	48	76	87	66
	90	408	58	76	89	60
	100	408	63	76	88	56

Table 2.2: Overall statistics for each of the 10 datasets in Data2007.

both arrivals and departures, meaning that approximately half of the routes in the dataset are forecast to be blocked. However, these same routes are open over 74% of the time in the weather that materializes (that is, there is a perturbed route in the neighborhood of the original route which does not pass through Level 3+ weather in the observed weather). The percentage of routes actually open is equal within each flight direction because the set of actual weather scenarios are identical (while the set of forecast scenarios depends on the planning horizon t_0).

The last two columns indicate how the forecasts and true weather differ for individual routes. Routes that are forecast as open are overwhelmingly open in the observed weather grid, with rates of 84% and above. Arrivals have slightly lower rates than departures, and the rates decrease with increasing planning horizon. Both of these trends are to be expected, because arrivals typically encounter the bottleneck at the end of their route through terminal airspace, where the forecasts are less accurate. Finally, routes that are forecast as closed are closed in the true weather approximately 60% of the time. These low rates reflect the effect of the additional flexibility allowed for finding routes in the actual weather.

To further evaluate this route-based method of viewing and using weather forecasts, Figure 2-10 shows forecast skill scores for Data2008, providing a direct comparison against Figure 2-2, which gave skill scores for the same weather scenario forecasts, but evaluated at the raw pixel level.

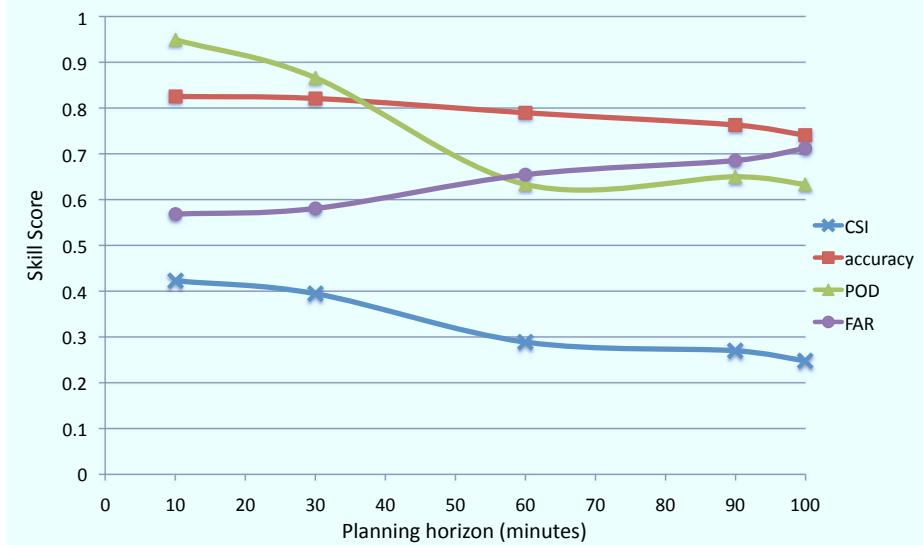


Figure 2-10: Skill scores for the route-based weather dataset improve over scores for pixel-based forecast evaluation.

Before we analyze these results, we must resolve the discrepancy in the notion of time horizon in the two settings. The time horizon τ for pixel-based forecasts does not have an analog in the route-based setting, because the associated dynamic grid has a planning horizon t_0 instead. However, the dynamic grid embeds forecasts of time horizon *at least* t_0 . In particular, t_0 , $t_0 + 5$, $t_0 + 10$, and $t_0 + 15$. Since forecast accuracy decreases with increased time horizon, we would expect this to only *hurt* the skill scores of the route-based forecast.

At the 10-minute horizon, the pixel-based forecasts outperform their route-based counterparts. This is likely due to the known high accuracy in CIWS at the very short 10-minute horizon. Since the route-based setting at the 10-minute planning horizon includes forecasts of length 10-25 minutes (as just discussed), and since $\tau = 25$ minutes gives much higher CIWS forecast error, the route-based skill scores can be expected to suffer.

However, the situation is reversed for time horizons of 30 minutes and greater, where POD, CSI, and FAR are significantly worse for the pixel-based forecasts. Indeed, the probability of detecting blocked routes (POD) is above 0.6 across time horizons, while the pixel-based POD scores are below 0.4 for all but the shortest time horizon. At 100-minutes, the route-based POD score posts a three-fold improvement

over its pixel-based counterpart. The critical success index (CSI) shows a similar story: it decreases from 0.45 to 0.25 with increasing time horizon, while the pixel-based CSI remains below 0.15 for time horizons above 60 minutes. Although the false alarm rate (FAR) is quite high for the route-based forecast (between 0.55 and 0.71), it is even higher in pixel-based context: above 0.75 for time horizons above 60 minutes.

Lastly, although the accuracy of the route-based forecast is above 0.75, it underperforms the pixel-based forecast. This is due to the large imbalance between the L3+ (positive) and L2- (negative) CIWS pixels, causing the true negatives (TN) to dominate the accuracy score. By contrast, the route-based dataset is more balanced, making its accuracy scores lower but more meaningful.

2.5.4 Conclusion

The raw data suggest that subject to minor adjustments, air route planning at horizons up to 100 minutes is quite reasonable, since routes that are forecast to be open end up being overwhelmingly so. Likewise, routes that are forecast to be blocked tend to be open in the observed weather, which indicates potentially underutilized capacity. This is encouraging, and shows that allowing even small adjustments to planned trajectories can improve the quality of decision-making based on the weather forecast.

However, the gap between routes which are predicted to be blocked and those that are actually so in Table 2.2, as well as the less-than-perfect skill scores in Figure 2-10, suggests that there is room for improvement in route-based forecasts. The next chapter introduces a method to bridge these gaps by using features of a weather forecast to better predict route blockage.

Chapter 3

Prediction of robust routes through terminal airspace

This chapter builds upon the route-based approach to evaluating weather forecasts for air traffic management, and uses techniques from machine learning to identify robust routes through airspace. That is, routes which are open once weather materializes, despite inherent spatial and temporal errors in the corresponding convective weather forecast.

We begin with the introduction of a set of features, or functions of the weather forecast, which are likely to influence route blockage. These features are first evaluated based on how well they can predict route blockage individually, and are then incorporated into classification algorithms. The classifiers are designed to predict whether a given route through terminal airspace will be open or blocked, given the values of these features. Several classification algorithms are described and evaluated based on performance, accuracy, and parameter sensitivity. The chapter ends with a translation of the binary classification of route blockage (open or blocked) into probabilities, resulting in a probabilistic predictor of route blockage.

3.1 Features for route blockage prediction

Having generated a dataset of terminal-area routes, the next natural step to improve the prediction of route blockage is to identify characteristics of the convective forecast which may best point to an increased likelihood that a given trajectory will be open. This section introduces the set of features chosen as potential blockage predictors.

Intuitively, if a planned route is forecast to have level 2 weather along its entire length, we may expect this route to be more likely to end up blocked than a route which only passes through forecast weather of level 0. This reasoning of how features of the forecast may indicate higher likelihood of blockage lead to the selection of the following eleven features of potential interest. For each route r through forecast weather scenario F :

- 1 mean VIL along route r ,
- 2 standard deviation of VIL along route r ,
- 3 minimum distance to level 3+ weather along route r ,
- 4 mean distance to level 3+ weather along route r ,
- 5 maximum VIL in neighborhood of route r ,
- 6 length of the most restrictive bottleneck that route r passes through,
- 7 maximum pixel density of level 3+ weather along route r ,
- 8 maximum VIL density along route r .
- 9 theoretical capacity for F ,
- 10 number of segments in the minimum cut of F , and
- 11 length of shortest minimum cut segment of F

These features fall into three feature types: features of the forecast along the route r (features 1-2), features of the forecast in the neighborhood of r (features 3-8), and features of the entire terminal weather forecast (features 9-11).

Features 1-4 are reasonably self-explanatory, and pertain to the forecast VIL along r , and to the proximity of r to hazardous weather. Feature 5 is the maximum VIL forecast in the neighborhood of radius B along r . Here B is flexible, but we match it to the B in our weather model (typically 8-10 km). Feature 6 is the length of the

tightest bottleneck between level 3+ weather through which r passes.

Feature 7 reflects the intensity of the weather in the neighborhood of r . It is computed by taking a B km neighborhood around r , and finding the strip of pixels perpendicular to r with the largest percentage of level 3+ forecast pixels. Feature 8 is computed using the same perpendicular strips as feature 7, except that it considers the largest average VIL along a perpendicular strip. If r is forecast to pass through level 3+ weather, features 3 and 10-11 will all equal 0, but features 7 and 8 may still contain pertinent information about the nature of the hazard.

Features 9-11 refer to the theoretical capacity of the dynamic forecast grid and the corresponding minimum cut, illustrated by example in Figure 1-1. This theoretical capacity is based on continuous maximum flow theory, which shows that the maximum throughput of a continuous domain (in our case, the annulus from C_O to C_I) corresponds to the minimum cut through a corresponding discrete graph (Mitchell, 1988). To compute the theoretical capacity, we follow the developments on continuous maximum flow extended to the case of airspace by Mitchell et al. (2006) and Krozel et al. (2007). This work presents a polynomial-time algorithm for computing the maximum flow through a polygon with holes, from a set of source edges to a set of sink edges. In our case, the polygon represents the terminal airspace, the holes represent weather, and C_O and C_I are sets of sources and sinks, respectively. The algorithm involves the creation of a discrete *critical graph*, where a shortest path

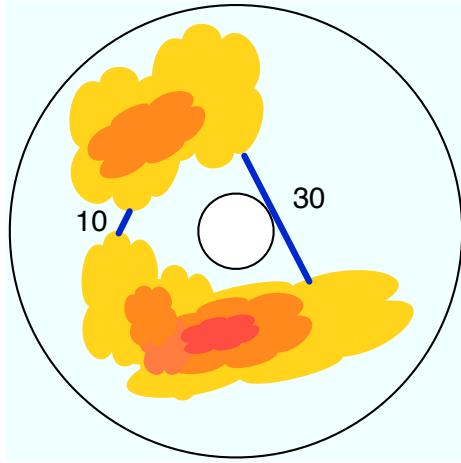


Figure 3-1: Illustration of minimum cut M and features 9-11 of the weather forecast.

through this graph gives the cost of the minimum cut through the continuous region, and is also equal to the maximum flow.

Let M be this minimum cut. Then feature 9 is the length of M , feature 10 is the number of disjoint segments in minimum cut M , and feature 11 is the length of the shortest segment of M . In Figure 3-1, features 9, 10, and 11 are 40, 2, and 10, respectively. Note that since the minimum cut gives the bottleneck for flow through the airspace, all open trajectories necessarily pass through this bottleneck region. However, since r is not necessarily open (it may pass through level 3+ pixels), it does not necessarily cross the minimum cut M .

3.2 Identifying robust routes using individual features

This section develops a simple model of route robustness, based on estimating the conditional probability that a route will be blocked given the value of an individual feature. The resulting estimates also provide a visual explanation of how each feature relates to route blockage.

3.2.1 The conditional probability of route blockage

Given route u through a weather scenario, let $f_i(u)$ be the value of feature i for route u . The following equation is used to empirically estimate the conditional probability that u is open given the value of feature i :

$$P(u \text{ is blocked} \mid f_i(u) = v) = \frac{\#\{\text{route } r \mid r \text{ is blocked} \& f_i(r) = v\}}{\#\{\text{route } r \mid f_i(r) = v\}} \quad (3.1)$$

where the $\#$ operator gives the size of the set. Since most features are continuous, feature values are binned when necessary to avoid zeros in the denominator of Equation 3.1. Bin sizes are chosen so as to balance the conflicting objectives of achieving low sampling error on one hand, and identifying a significant trend on the other. That is, small bin sizes result in bins with very few data points, and thereby in large

sampling error. This large error results in a large confidence interval about the ratio of open to blocked routes within these bins, precluding any firm conclusion about this ratio. In contrast, large bin sizes may wash out some of the trend in blockage probability as a function of feature value. At the extreme, placing all data into a single bin would not tell us anything about the different blockage rates as feature value increases. The bin sizes used are 5, 10, or 20 units.

We make one adjustment to the binning of continuous features to account for the long tail in the distribution of several features. For these features, there exists a very low density of data at values above some threshold, resulting in tail bins with very few or no data points, and hence very large sampling error. To avoid this, we group the last 2% of data into a tail bin. For instance, for bin size 10 and a feature with values in the range [0,100], but with very sparse data above value 50, we would end up with the bins [0, 10), [10, 20), [20, 30), [30, 40), [40, 50), and [50, ∞), and with the respective labels 5, 10, 20, 30 ,40, and 50+.

Let \hat{p} denote the resulting conditional probability in Equation 3.1 for a given bin. We can compute a confidence interval for \hat{p} for each bin, but our data is such that many bins have \hat{p} very close to 0 or 1, and these are often the bins with few data points (n less than 10). To avoid the poor performance of the standard Wald confidence interval under these conditions, we use the Agresti-Coull variation to the Wald confidence interval (Agresti and Coull, 1998; Brown et al., 2001). The Agresti-Coull interval effectively places a Bayesian prior onto \hat{p} , by simply adding 2 data points to each class before computing the point estimate \hat{p} and the confidence interval $\hat{p} \pm z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$, where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ th percentile in the normal distribution.

Due to sampling error introduced in the process of data selection and binning, the conditional probabilities contain some noise and must be smoothed. The smoothed conditional probability $P(u \text{ blocked} | f_i(u) = v)$ is computed by taking the average of 5 neighboring bins (bin v as well as 2 bins on each side of v), weighted by the number of routes in each bin. The window size of 5 was chosen empirically to decrease sampling error without smoothing out local trends.

3.2.2 Comparison of results across features

We now evaluate the set of features in terms of the conditional probabilities of route blockage just described, by plotting \hat{p} at each feature value along with confidence intervals and the smoothed estimate of \hat{p} .

Figure 3-2 shows results for each feature, with planning horizon fixed at 60-minutes, and flight direction fixed to departures. Overall, we see that different features exhibit different relationships to blockage in terms of type and strength of trend. There are several metrics along which we can compare the plots. First, we

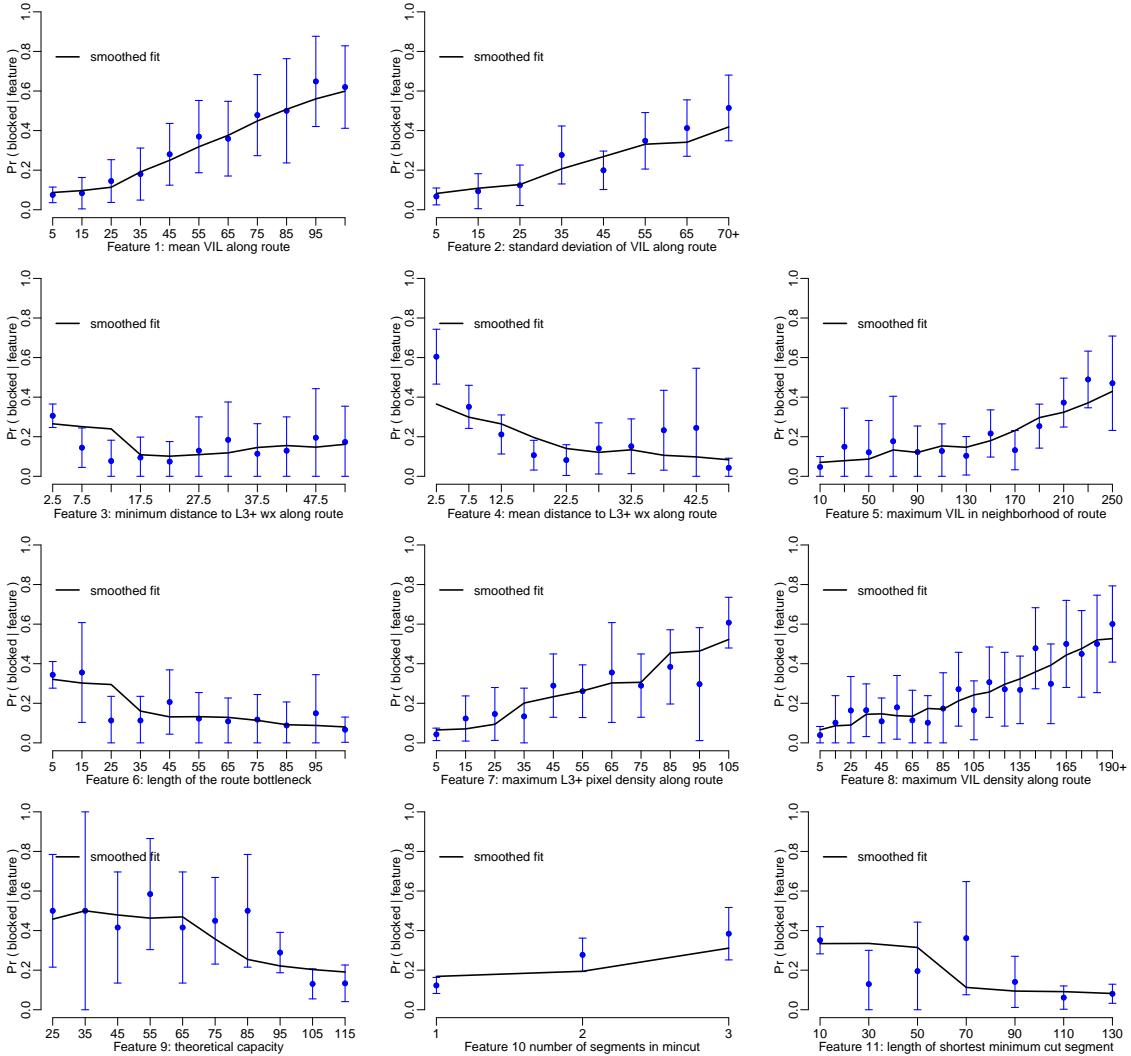


Figure 3-2: Probability of route blockage conditional on the value of each feature, for departures at the 60-minute planning horizon.

can compare the difference between maximum and minimum blockage probability, which gives a measure of magnitude for the blockage fit. Higher magnitude means the blockage estimate is more discerning (has more certainty associated with both open and blocked class predictions), and hence a more useful predictor. Second, the size of the confidence intervals also gives a measure of the uncertainty associated with a prediction, making smaller intervals preferable.

Most features have a monotone trend, either non-increasing or non-decreasing in feature value. While several features show a distinct trend in blockage as a function of feature value, others' blockage line is quite flat. Moreover, several features are accompanied by very large confidence intervals at each bin, making it difficult to identify any trend in the data with much certainty.

For the first two features (mean VIL along route, and standard deviation of VIL along route, respectively), the likelihood of blockage increases with increased feature value. This agrees with intuition, since high VIL values along a route indicate that it passes through some weather-affected regions, which are more likely to show up as level 3 or higher in the true weather. Likewise, higher standard deviation of VIL indicates increased variability in weather conditions along the route, and hence higher likelihood that weather will materialize. These route-based features are among the strongest predictors of blockage.

The next set of features, which are functions of the neighborhood of the route, vary in their performance as predictors. For feature 3 (minimum distance to weather), we see that routes which are very close to hazardous weather (in the forecast) end up more likely to be blocked, while routes which are very far from forecast hazards stay viable in the true weather. Feature 4 (average distance to weather) shows the same trend with higher magnitude. The blockage fits for features 5 (max VIL near route) and 6 (length of bottleneck for route) also agree with intuition: as feature 5 increases, so does blockage probability, while as feature 6 increases, blockage probability decreases. Features 7 (max pixel density near route) and 8 (max VIL density near route) are particularly good indicators of route blockage, with smoothed probabilities of blockage increasing from 0 to almost 0.6 as feature value increases. The only feature which

such high magnitude is feature 1.

The final set of features deal with the weather forecast scenario as a whole. Feature 9 (theoretical capacity) has the noisiest results, with very large confidence intervals and a flat trend line, though there is a slight correlation between increased theoretical capacity and decreased route blockage probability. Feature 10 (number of min cut segments) also shows a relatively flat blockage trend. This is likely because the number of segments in a minimum cut can occur both if the theoretical capacity is very high (there is 1 large cut segment, for instance), or very low (there is a very small cut segment). The high capacity case would make routes more likely to be viable, since they are less likely to be close to adverse weather cells, while the reverse is true for low capacity. And finally, feature 11 (length of shortest min cut segment) does have tight confidence intervals at the extremes of the distribution, with higher probability of blockage at low feature values (though still under 0.5), and vice versa for high feature values. However, the confidence intervals in between these extreme bins are very large. Overall, features 9-11 are the poorest predictors of route blockage.

3.2.3 Results across flight direction and planning horizon

In the previous section we compared route blockage plots for individual features, at fixed flight direction (departures) and planning horizon (60-minutes). We now perform comparisons between arrivals and departures, and between varying time horizons, with other parameters fixed. We demonstrate the general trends by showing one example of each.

Figure 3-3 compares route blockage for arrivals and departures, with feature value and planning horizon fixed at feature 1 and 60-minutes, respectively. The two plots are very similar; both show a monotonically non-decreasing blockage line from 0.1 to 0.6, with identical bins and comparable confidence intervals. There is a slight difference in the smoothness of the two blockage trends, in that departures have a smoother blockage fit than arrivals.

We now compare the sensitivity of route blockage to planning horizon. Figure 3-4 shows the results as planning horizon varies across horizons of 10, 30, 60, and 90

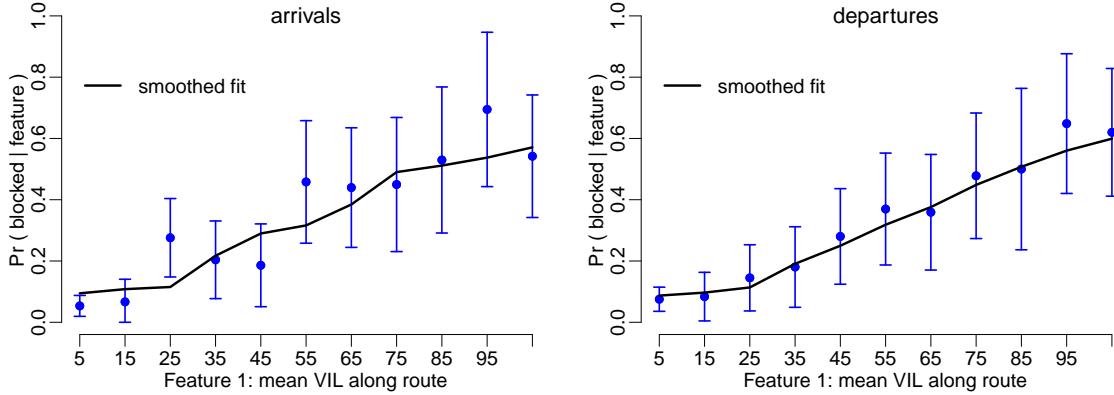


Figure 3-3: Sensitivity of route blockage to flight direction, for feature 1 (mean VIL along route) at the 60-minute planning horizon.

minutes, with feature value and flight direction fixed to feature 8 and departures, respectively.

The difference in plots as time horizon increases is striking. Both the magnitude of blockage and the sizes of confidence intervals are significantly better at the 10-minute planning horizon as compared to the 90-minute. In fact, the curve is close to an ideal “S-shaped” curve at 10-minutes, with endpoints converging close to 0 and 1, and with relatively small confidence intervals. As time horizon increases, the blockage curve decreases in magnitude (it peaks at only 0.5 for $t_0 = 90$ minutes), it flattens out, and the confidence intervals increase in size. These trends are consistent with the fact that forecasts are more accurate at shorter time horizons.

These visualizations of the relationship between individual features and blockage are useful as predictors of route blockage and to better understand feature behavior. However, we have not identified a single best predictor or blockage, nor a way to combine features to create an improved predictor. We spend the rest of the chapter doing just this, by using methods from machine learning to compare individual features more concretely, and then developing a classification algorithm to improve blockage prediction.

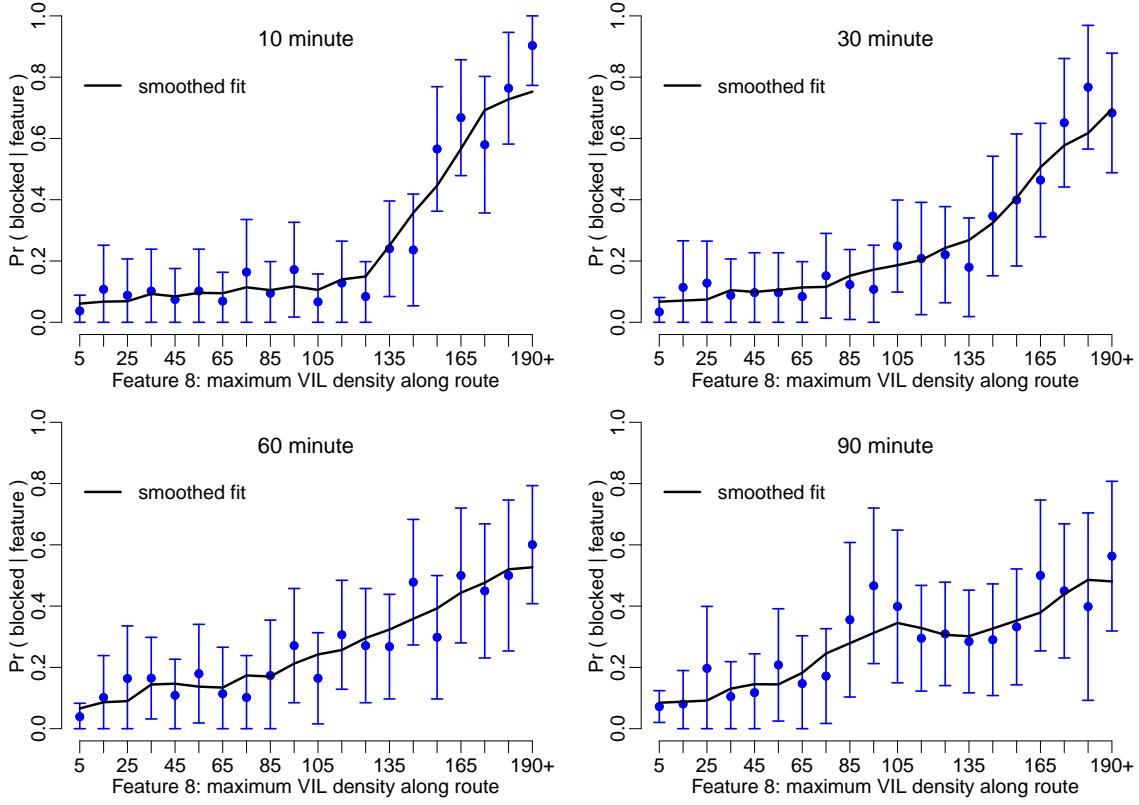


Figure 3-4: Sensitivity of route blockage to varying planning horizon, for feature 8 (max VIL density near route) and departures.

3.3 Feature selection using mutual information

To evaluate features for classification and gain a better understanding of which features best correlate with blockage individually, we compute the Mutual Information (MI) between each feature X_i and the blockage label y (+1 for open, -1 for blocked). Mutual information is an information-theoretic measure of the dependence between two random variables X and Y , and measures how much the uncertainty of X is reduced if Y is observed. This measure considers each feature individually and does not capture situations in which two features combined correlate well with y . In other words, the larger the value of MI for a feature, the greater the correlation of that feature to route availability.

For discrete random variables X and Y with joint pmf $p_{X,Y}(x,y)$, and marginal pmfs $p_X(x)$ and $p_Y(y)$, respectively, their mutual information, $I[X;Y]$, can be ex-

pressed as

$$I[X; Y] = \sum_x \sum_y p_{XY}(x, y) \log \frac{p_{X,Y}(x, y)}{p_X(x)p_Y(y)}$$

We approximate the distribution functions using their Maximum Likelihood parameter estimates, which are valid when the dataset size is much larger than the domain size of the joint pmf. In other words, $\hat{P}_X(x) = \frac{\hat{n}_X(x)}{n}$ and $\hat{P}_{X,Y}(x, y) = \frac{\hat{n}_{X,Y}(x, y)}{n}$ are used as estimates of the pmfs. For features with continuous domains, we do not adopt more complicated approaches to approximating MI for continuous distributions, and instead discretize the data into equally sized bins, as described in the previous section.

Figure 3-5 contains a comparison of MI between each feature and blockage, across planning horizons for both arrival (left-hand-side histogram) and departure (right-hand-side histogram) datasets. The relative MI of features tends to reflect two principles: the spatial accuracy of a weather forecast decreases with increased time horizon, and the weather forecast in the neighborhood of a route (including the structure of weather cells) has bearing on the viability of a route. Indeed, features 7 and 8, which reflect both the structure and intensity of forecast weather in the neighborhood of a route, consistently have the highest MI scores. Feature 1 also tends to perform well at shorter planning horizons, when the weather forecast on the route is subject to relatively low spatial error. On the other hand, features 4 and 6 outperform fea-

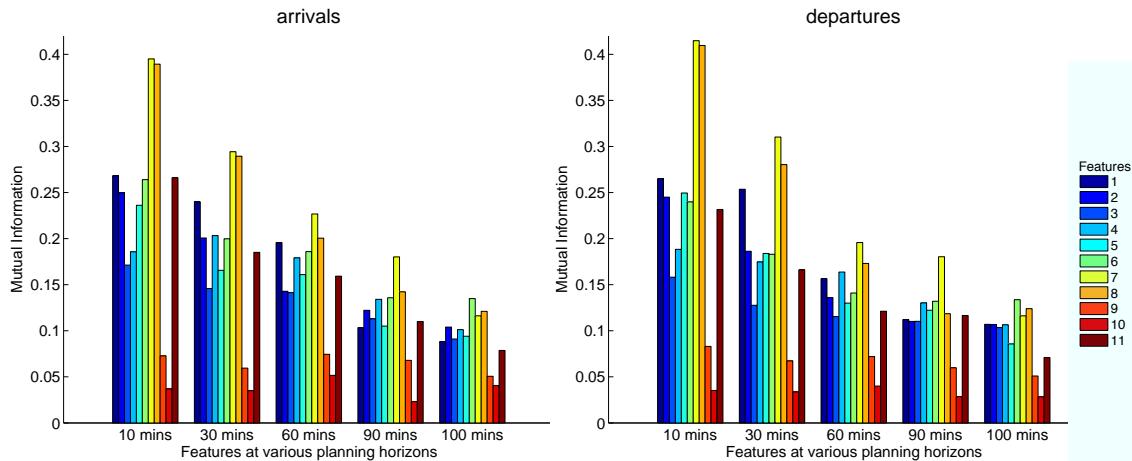


Figure 3-5: Comparison of mutual information across features and planning horizons for arrivals (left) and departures (right)

ture 1 at longer planning horizons, since they reflect the prevalence of weather in the neighborhood of the route (and not just the weather activity on the actual route).

As expected, the MI values tend to be higher overall at shorter planning horizons, reflecting the higher forecast accuracy at shorter time horizons. This also explains why departures have slightly higher MI than arrivals across the board, because arrivals cross the inner circle C_I (a bottleneck region) at the end of their trip through the dynamic terminal-area, while departures cross at the very beginning, when the forecast at C_I has a shorter time horizon. Figure 3-5 also reveals several insights into the relationship between VIL forecasts and route blockage: for example, it shows that feature 3 (the minimum distance between the route and level 3+ weather in the forecast), is a worse predictor of blockage than feature 4 (the mean distance of the route to level 3+ weather).

Finally, Figure 3-5 shows that the theoretical capacity (features 9 and 10), which is a frequently-cited quantity in planning terminal-area routes (Krozel et al., 2007), is a poor predictor of route blockage. This may be because while the theoretical capacity predicts that N arrival routes will be open for the next hour, it does not indicate the constraint (which is critical for planning) that these routes all enter the terminal-area from the west. Moreover, it is possible for the forecast and realized theoretical capacities to be identical, and to yet require that aircraft use trajectories that deviate significantly from the planned routes.

The above analysis provides a better understanding of how well the features of a convective weather forecast correlate with route blockage. In the next section, the features will be combined into a classification algorithm to predict robust routes.

3.4 Classification algorithms for route availability

This section describes the classification algorithms used for supervised learning of route availability. We first give a brief introduction to supervised learning and classification, describe the training objectives, and then describe the ensemble methods used for classification.

3.4.1 Introduction to supervised learning and classification

The task of *supervised learning* is to use *supervised data* to build a prediction function f that can predict an unknown outcome. For example, an individual (Alice) would like to automatically determine whether a new email is spam. Alice already has information about 1000 previously received emails and knows which of those emails are spam. She can use this knowledge to build a function that will predict whether or not new incoming email is spam.

Supervised data is a set of points (x_i, y_i) , $i = 1, 2, \dots, n$, where the x_i are input values, or *features*, and the y_i are output values. The x_i can be a vector of values. In the spam example, the features x_i may be counts of certain words such as “Nigeria” and “free” in the i ’th received email, $i = 1, \dots, n$. The output y_i is 1 if the i ’th email is spam, and 0 otherwise. The process of selecting the features x_i is called *feature selection*.

When the output values y_i are discrete, the supervised learning function is referred to as a *classifier*; when these values are continuous, it is a *regression*. When there are only two classes, the task is a *binary classification* problem. Many classification methods have been developed, and the choice of which to use for a particular problem depends on the data. The process of tuning a classification method to a specific dataset (typically by selecting parameters for the function f in some way) is called *training*. The data used for training is called *training data*. When a trained classifier is used to predict the outcomes of new *test data* for evaluation and validation purposes, the process is called *testing*.

One simple set of binary classification algorithms are *linear classifiers*, which use a linear combination of the features x to predict the outcome y . In other words, a hyperplane is selected to separate the two classes. In particular, the *optimal separating hyperplane* separates the two classes by maximizing the distance between the separating hyperplane and any training point x_i . This optimal hyperplane (β, β_0) can be identified by solving the following quadratic optimization problem (which can be

done efficiently in practice):

$$\begin{aligned} & \min ||\beta||^2 \\ \text{s.t. } & y_i(x_i^T \beta + \beta_0) \geq 1, i = 1, \dots, n \end{aligned}$$

The resulting prediction function $f : \mathbb{R}^n \rightarrow \{0, 1\}$ for input data $x \in \mathbb{R}^n$ is the following: $f(x)$ is 1 if $\beta x + \beta_0 \geq 0$ and 0 otherwise.

When the training dataset is not separable, the formulation above is infeasible. In this case, the idea of finding a maximum margin separating hyperplane can be generalized by introducing a slack variable to the mathematical formulation to deal with unseparable datapoints, along with a penalty for missclassifying training data. The resulting formulation is still a convex optimization problem, and the resulting classifier is a *Support Vector Machine (SVM)*. SVMs are powerful in that hyperplanes can be extended to non-linear functions of the features using *kernel* functions. One commonly used kernel function is the *Radial Basis Function (RBF) kernel*.

An *ensemble classifier* is a set of individual classifiers that are trained separately, and whose predictions are combined into a single ensemble prediction. These individual classifiers can be SVMs, decision trees, other classifiers, or a combination thereof.

For more information about supervised learning, Hastie et al. (2003) and Mitchell (1997) are excellent resources.

3.4.2 Classification training objectives

When evaluating a classifier, the class predictions are compared with the actual classes of a test set, according to the standard *two-class confusion matrix*:

	predicted open	predicted blocked
actual open	TP (True Positive)	FN (False Negative)
actual blocked	FP (False Positive)	TN (True Negative)

We note the difference between these definitions and those in Section 2.2: here, a positive is defined to be an open route (i.e. a route not containing hazardous

weather). This difference is a natural consequence of the move from a pixel-based to a route-based evaluation of forecast accuracy.

Although it is typically desirable to maximize the accuracy (total number of correct predictions) of a classifier on a test set, the context of aviation weather warrants a modified objective. Due to safety concerns, it is more important to correctly predict a route that ends up blocked than one that ends up open. This emphasis on correctly predicting members of the blocked class (minimizing false positives) is complicated by the fact that the dataset is imbalanced, having fewer blocked examples than open, making it inherently harder to perform well on the blocked (minority) class. As we will see in Section 3.5, there exists an inherent tradeoff between the false positive rate and accuracy, which translates into a tradeoff between increased safety and underutilized capacity.

In addition to the FP and FN rate, we compute the following (standard) performance metrics to evaluate our classifier:

$$\begin{aligned} \text{accuracy} &= \frac{TP + TN}{n} \\ \text{true negative accuracy}(a^-; \text{recall}) &= \frac{TN}{TN + FP} \\ \text{true positive accuracy}(a^+) &= \frac{TP}{TP + FN} \\ \text{g-mean} &= \sqrt{a^- \times a^+} \end{aligned}$$

where n is the total number of routes in the data set.

In particular, the recall is a measure of how well the classifier performs on members of the minority class which in this case is the set of routes that are blocked in the actual weather that materializes. We seek to maximize this value through classification.

3.4.3 Classification of route blockage

The machine learning literature has shown that ensemble classifiers tend to perform well on imbalanced datasets, outperforming non-ensemble methods (Hulse et al., 2007; Chen et al., 2004; Liu et al., 2006). We follow these methods closely, and develop two

ensemble classification algorithms: an ensemble of Support Vector Machines, and an ensemble of weighted decision trees, also known as a Weighted Random Forest.

Figure 3-6 depicts the process of training an ensemble classifier on an imbalanced dataset.

The process consists of several steps:

- 1 Data2007, the dataset described in Section 2.5, is randomly partitioned into a training set consisting of 70% of data points, and a test set consisting of the remaining 30%. To ensure an unbiased data set, weather scenarios from the same date are not split up.
- 2 The training set is further processed: m blocked instances of the training set are set aside, and N bootstrap samples of the open instances are generated, each of size m .
- 3 The blocked set is combined with each of the bootstrap samples to create N bootstrap training sets of size $2m$, with a balance between open and blocked instances.
- 4 These N bootstrap samples are then used to train N classifiers with 5-fold cross-

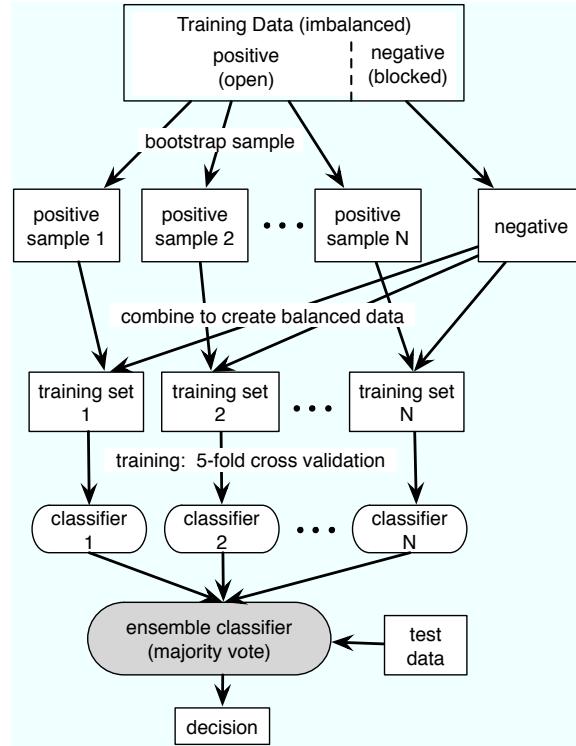


Figure 3-6: Process for training an ensemble classifier on an imbalanced dataset.

validation and a grid search to select parameters which maximize classification recall.

- 5 The resulting N classifiers combine to form a single ensemble classifier. On a new test data point, each ensemble member gives a decision (open or blocked), and the ensemble decision is simply the majority vote.

In our experiments, the number of classifiers in the ensemble is $N = 11$, and m (the bootstrap sample size parameter equal to the number of blocked instances) varies based on planning horizon and other dataset parameters, but averages to 55 (20% of the dataset size).

Two classification methods were trained with this ensemble approach, by changing the classification algorithm used in Step 4. The first is an ensemble of Support Vector Machines (SVM), each trained with an RBF kernel. The second is a Weighted Random Forest, which is trained using a set of increasing weights that penalize the misclassification of blocked examples, and hence promote high recall. We refer to these classifiers as ESVM and WRF, respectively, for the remainder of the chapter.¹

Two additional classifiers were trained on Data2007 to predict blockage, namely, a (single) SVM with an RBF kernel, and a decision tree with a weighted loss function. Since ESVM and WRF outperformed them in terms of maximizing recall, we only discuss them briefly in the next section.

3.5 Classification Results

In this section, we analyze the performance of the ESVM and WRF classifiers that were just introduced. We first discuss the ESVM classifier, and include an analysis of result sensitivity to several model parameters. We next discuss the performance of the WRF classifier, and finish with a brief summary of the nonensemble classifiers, which performed poorly in comparison.

¹The classification algorithms were trained and tested using the R language for statistical computing (R Development Core Team, 2008). The R package e1071 was used for Support Vector Machines (Dimitriadou et al., 2009), and the package rpart was used for Weighted Random Forests (Therneau and Atkinson, 2008).

3.5.1 Results for Ensemble of SVMs Classifier

To evaluate ESVM , we train the classifier using Data2007 for five planning horizons (10-, 30-, 60-, 90-, and 100-minutes), for arrivals and departures, and with model parameters fixed to $R_I = 10$ km, $R_O = 100$ km, and $B = 8$ km. We then test the resulting classifier using Data2008, an independent dataset. We compare the decision of ESVM on each route against the deterministic raw weather forecast, denoted by F_x , which classifies a route as open if each pixel along the route is level 2- in actual weather (in the dynamic grid), and classifies it as blocked otherwise.

Table 3.1 shows the performance of ESVM compared to F_x , with results averaged over five runs of the classifier. Overall, we find that at the shorter planning horizons of 10-, 30-, and 60-minutes, the ensemble classifier outperforms F_x in terms of accuracy, but not recall. This situation is reversed at the longer planning horizons of 90- and 100-minutes, where there is improvement in the recall rate of ESVM at a cost to accuracy. Indeed, arrivals at 90-minutes post a 3% improvement in recall rate over the F_x , with similar cost to accuracy rate. At 100-minutes, the improvement in recall rate is 17%, with a 15% cost to accuracy. The results for departures show similar trends.

These results agree with intuition, since the recall rates of the raw weather forecast (and hence F_x) are known to be high at shorter horizons where weather forecasts are known to be more accurate, leaving little room for improvement. However, the accuracy rates at these short planning horizons can be improved due to the wiggle room introduced when validating routes, allowing for slight shifts in the original routes, often uncovering routes around weather cells present in the area. When planning horizon increases, the classifier can make gains in the recall rate. But this gain comes at a cost to accuracy: because the classifier is trained to be more conservative in declaring routes as open, it is likely to declare some routes which are actually open as blocked. Due to the imbalance in the test set, this necessarily causes a noticeable decrease in accuracy on the open (majority) class. This tradeoff between recall rates and accuracy can be directly controlled in the Weighted Random Forest classifier

	10-min		30-min		60-min		90-min		100-min		
	ESVM	Fx	ESVM	Fx	ESVM	Fx	ESVM	Fx	ESVM	Fx	
arrivals	Acc	88.57	82.14	81.87	77.68	75.80	76.34	76.07	77.68	58.04	72.77
	a^-	83.89	91.67	73.33	77.78	62.78	66.67	75.00	72.22	87.22	69.44
	a^+	89.47	80.32	83.51	77.66	78.30	78.19	76.28	78.72	52.45	73.4
	g-mean	0.87	0.86	0.78	0.78	0.70	0.72	0.76	0.75	0.66	0.71
	% TP	75.09	67.41	70.09	65.18	65.71	65.62	64.02	66.07	44.02	61.61
	% FP	2.59	1.34	4.28	3.57	5.98	5.36	4.02	4.46	2.05	4.91
	% TN	13.48	14.73	11.79	12.5	10.09	10.71	12.05	11.61	14.02	11.16
departures	% FN	8.84	16.52	13.84	18.75	18.22	18.3	19.91	17.86	39.91	22.32
	Acc	90.71	79.91	81.07	77.68	77.05	74.55	69.2	73.66	68.66	74.11
	a^-	78.97	84.62	78.46	82.05	62.05	64.10	61.54	64.10	71.8	66.67
	a^+	93.19	78.92	81.62	76.76	80.22	76.76	70.81	75.68	68.00	75.68
	g-mean	0.86	0.82	0.80	0.79	0.70	0.70	0.66	0.70	0.69	0.71
	% TP	76.97	65.18	67.41	63.39	66.25	63.39	58.48	62.50	56.16	62.50
	% FP	3.66	2.68	3.75	3.12	6.61	6.25	6.70	6.25	4.91	5.80
arrivals	% TN	13.75	14.73	13.66	14.29	10.80	11.16	10.72	11.16	12.50	11.61
	% FN	5.62	17.41	15.18	19.20	16.34	19.20	24.11	20.09	26.43	20.09
departures	Acc	90.71	79.91	81.07	77.68	77.05	74.55	69.2	73.66	68.66	74.11
	a^-	78.97	84.62	78.46	82.05	62.05	64.10	61.54	64.10	71.8	66.67
	a^+	93.19	78.92	81.62	76.76	80.22	76.76	70.81	75.68	68.00	75.68
	g-mean	0.86	0.82	0.80	0.79	0.70	0.70	0.66	0.70	0.69	0.71
	% TP	76.97	65.18	67.41	63.39	66.25	63.39	58.48	62.50	56.16	62.50
	% FP	3.66	2.68	3.75	3.12	6.61	6.25	6.70	6.25	4.91	5.80
	% TN	13.75	14.73	13.66	14.29	10.80	11.16	10.72	11.16	12.50	11.61
departures	% FN	5.62	17.41	15.18	19.20	16.34	19.20	24.11	20.09	26.43	20.09

Table 3.1: Validation results for the ensemble SVM classifier (ESVM), compared to the raw weather forecast (Fx)

through the weight in the training loss function, which places an explicit penalty on misclassified blocked routes. This will be explored later in this section.

For completeness, we evaluate ESVM in the same way we evaluated the pixel-based and route-based forecasts in Sections 2.2, and 2.5.3, respectively. That is, we perform a direct comparison of ESVM against the raw route-based forecast (Fx) in terms of CSI, POD, FAR, and Accuracy skill scores. Figure 3-7 shows the results, with ESVM skill scores shown as solid lines, and Fx scores as dotted lines. The data for ESVM comes from the average of five trained classifiers for both arrival and departures, evaluated on the Data2008 dataset. Note that here the positive class is reversed to match the initial discussion of skill scores in Section 2.2, and the positive class are routes which are blocked/hazardous. We find that ESVM outperforms Fx in terms of CSI and FAR for all time horizons. On the other hand, it under-performs in terms of overall accuracy (except for at the 10-minute planning horizon). POD is more complicated, as ESVM has higher (hence better) POD at the longer planning horizons of 60-minutes and up. The discussion of Table 3.1 above explains these results.

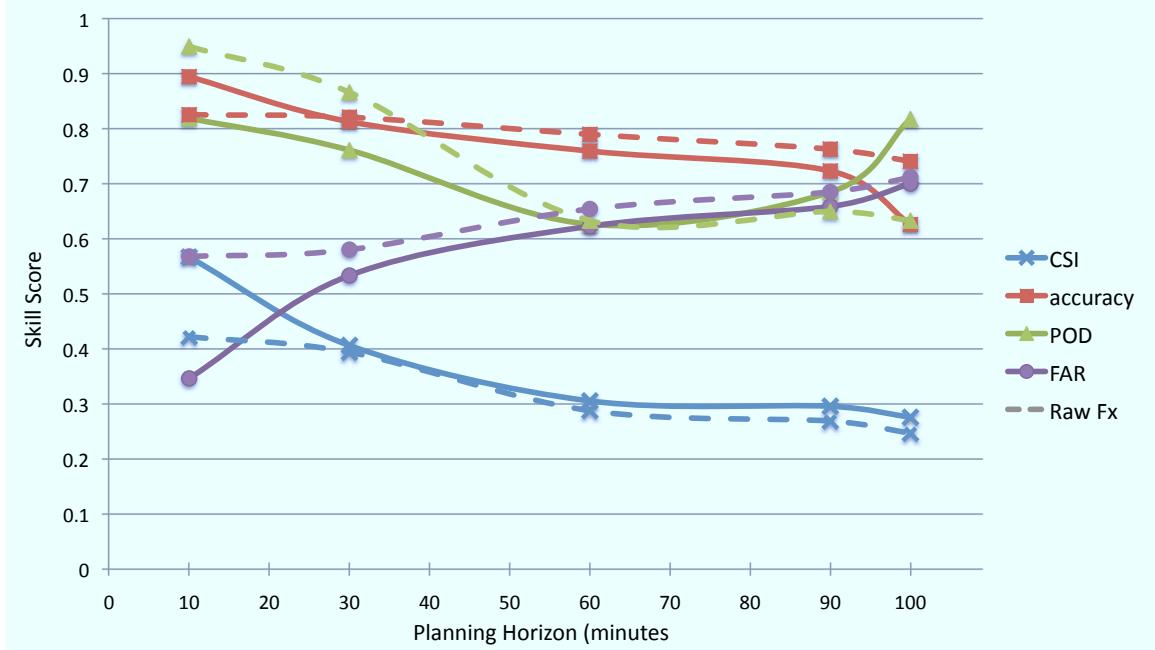


Figure 3-7: Skill scores for the ESVM forecast of route blockage (solid lines), as compared to the raw route-based weather forecast (dotted lines).

Now that we have seen the performance of ESVM for a fixed set of terminal-area model parameters, we analyze its sensitivity to several of these parameters.

3.5.2 Sensitivity of classifier to inner radius R_I

Several parameters of the terminal-area model were fixed in the previous section. We now investigate the behavior of ESVM as two of these parameters vary in real-world ways. Ideally, ESVM is not overly sensitive to any single model parameter.

We begin with an analysis of the sensitivity of ESVM to the inner radius R_I . In our terminal-area model, arriving aircraft begin their final merge, or the downwind leg of their flight, upon crossing R_I . As such, the specific characteristics of a given airport (for instance, a longer downwind leg, or a final merge starting farther from the airport) should determine the appropriate setting of the parameter R_I .

To test classifier sensitivity to R_I , we vary it between 10 km and 30 km in 5 km increments. This represents the range of realistic values of R_I across airports. Datasets are created for each value of R_I (note that feature values will change with R_I), for all planning horizons, for both arrivals and departures, and with wiggle room

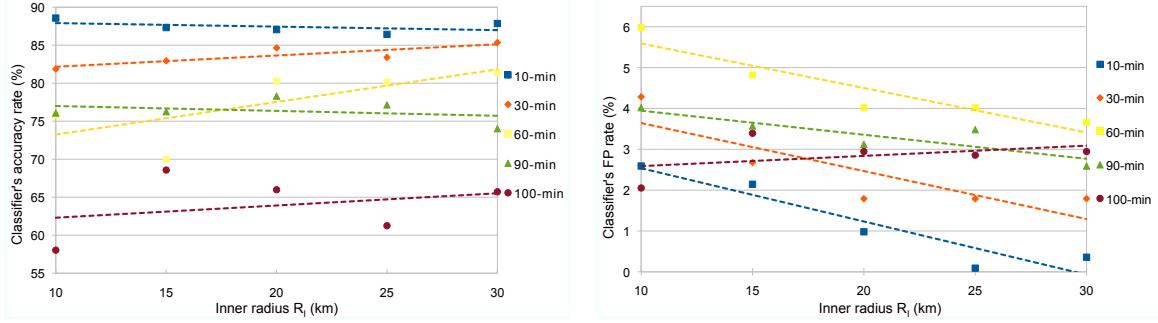


Figure 3-8: Sensitivity of the ESVM classifier to R_I , as measured in terms of accuracy (left) and false positive rates (right).

B fixed at 8 km. The ESVM training process is run for each dataset, and the same statistics as in Section 3.5.1 are calculated for the test set.

Figure 3-8 shows the effect that R_I (varied from 10 to 30 km) has on accuracy and false positive rates, for the case of arrivals at the 10-100 minute planning horizon. The accuracy rates stay fairly consistent with increasing R_I , although variability in accuracy is greater at longer planning horizons. The false positive rates tend to decrease with increasing R_I , effectively giving better performance when routes are shorter. We hypothesize that this is because shorter routes inherently have fewer potential hazards to avoid, and because a larger diameter around the airport relaxes the bottleneck for flow.

3.5.3 Sensitivity of classifier to wiggle room B

In this section we study the sensitivity of the classification results to the wiggle room parameter, B . Although B was previously set to 8 km, it is an adjustable parameter meant to represent the maneuverability that an aircraft is allowed without having to change its (declared) planned route.

Figure 3-9 illustrates the effect of varying B between 0 km (no wiggle room allowed) and 20 km on classification accuracy (gray) and false positive rates (blue). The box plot end points represent the minima and maxima of the data, which includes both arrivals and departures, and inner radius R_I values of 10 and 20 km.

The results show that average accuracy rates tend to increase with B , while av-

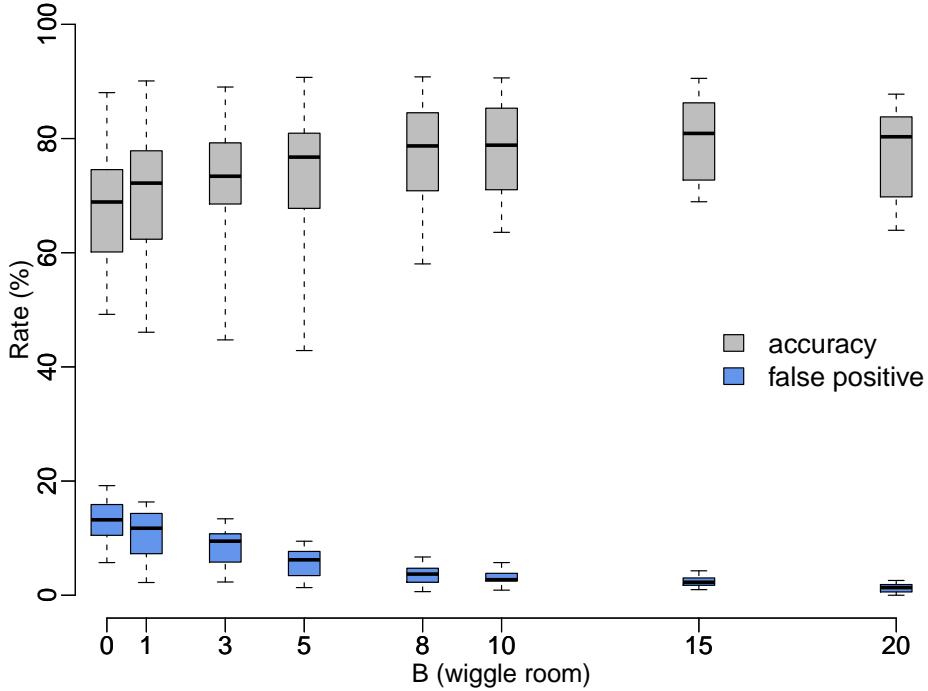


Figure 3-9: Box plot of sensitivity of the ESVM classifier to wiggle room B , as measured in terms of accuracy (gray) and false positive rates (blue).

verage false positive rates tend to decrease with B . These trends can be explained as follows: increased wiggle room for any given weather scenario makes it more likely that the route will be open (and the classifier learns that increased maneuverability makes routes more likely to persist), which decreases the number of potential false positives. At the same time, the imbalance between open and blocked routes results in increased accuracy when more routes are classified as open. The results show smaller variance as B increases, since the wiggle room removes some of the randomness in the spatial prediction.

It is worth noting that the improved classification performance with larger wiggle room must be traded off against a diminished capability for fine-grained planning. For example, a smaller wiggle room could take priority along an Area Navigation (RNAV) route. In particular, RNAV has enabled the introduction of air traffic routes along which aircraft are not constrained to fly from beacon to beacon. Instead, these routes increase lateral freedom and allow aircraft to fly any route within a network of

beacons. This has enabled performance-based navigation such as Required Navigation Performance (RNP) routes, which require among other things that aircraft flying the route calculate their 3D position to within k nautical miles of their true position. For low values of k , it would be necessary to specify appropriately low values of wiggle room B .

3.5.4 Results for Weighted Random Forest classifier

The performance of the WRF classifier is similar to that of ESVM, as it can learn from the features of the weather forecast to predict blocked routes, thereby improving recall over the raw weather forecast, but at a cost to overall accuracy. The detailed classification metrics are omitted to avoid repetition (they are similar but inferior to ESVM). However, we do explore the effect of the WRF weight parameter, which explicitly penalizes misidentified blocked routes, providing an illustration of the tradeoff between FP rate and accuracy.

The direct control of this tradeoff is very useful in practice, as a decision maker such as a traffic manager could use it to select an operating point, choosing between higher accuracy on one hand, which would mean a more aggressive strategy and hence higher throughput, and a lower false positive rate on the other, which would be a more conservative strategy with a potential capacity loss.

Figure 3-10 depicts this relationship across four planning horizons, with R_I , R_O , and B fixed at 10, 65, and 8 km, respectively. A diagonal trend is evident between the FP rate and the accuracy rate of the WRF for each planning horizon. The label on each point (where each point is the mean of 10 iterations) contains the weight used in the training function. Points associated with a lower weight tend to be in the top right (higher FP rate and accuracy), while points associated with a higher weight tend to be in the bottom left (lower FP rate and accuracy), for each planning horizon.

The figure also depicts the changes in accuracy and FP rates across the planning horizons: at the shorter planning horizons, the classifier can attain higher accuracy rates and lower FP rates (due to the greater reliability of the weather forecast), while

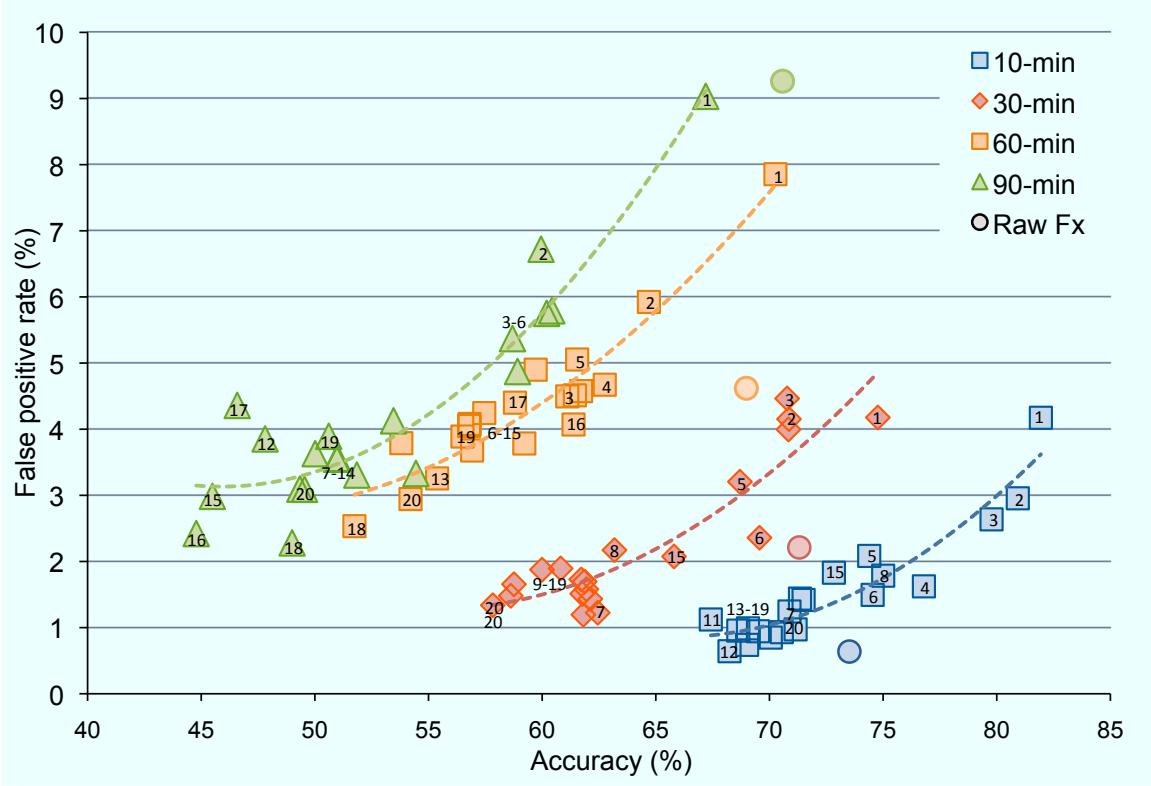


Figure 3-10: Comparison of false positive and accuracy rates of WRF for each planning horizon, as a function of weight (penalty against false positives).

at the longer planning horizons, the absolute improvement in FP rate is greater, but at a correspondingly larger cost to accuracy, than at shorter planning horizons.

Finally, Figure 3-10 depicts the FP and accuracy statistics for the raw weather forecast (Fx, as defined in Section 3.5.1). These are denoted by circles, as a point of comparison. At the 10-minute planning horizon, Fx dominates WRF , while the situation is reversed by 100-minutes, when Fx is dominated by WRF at all values of the weight. This reinforces the notion, also observed for ESVM , that the advantages of classification are realized for planning horizons longer than 10-minutes and increase with length of planning horizon.

3.5.5 Two more classifiers

Several additional classifiers were trained on the route blockage dataset, in order to validate the results of ESVM and WRF , and compare with other, non-ensemble,

classification methods. Since ESVM and WRF outperformed these methods in terms of maximizing recall, we only include brief descriptions and summaries of results.

A Support Vector Machine (SVM) was trained using two types of datasets. The first was a simple (imbalanced) partition of Data2007 into a training and test set. The second took this training set and oversampled the minority (blocked) class to produce a balanced training set. An SVM with an RBF kernal was trained on each dataset using 5-fold cross-validation to optimize for recall. A separate classifier was trained on many subsets of features, where feature combinations were selected based on mutual information and by balancing different feature types (features related to the weather grid such as theoretical capacity, and features related to the specific route such as mean VIL along the route). However, both data sets resulted in classifiers with very high FP rates (as compared to ESVM and WRF), though they also had higher accuracy rates than Fx.

In addition, a decision tree was trained on an imbalanced data set. In order to maximize the recall rate, a weighted loss function was used just like for the WRF. Even with a high penalty for miss-classifying the blocked class, the resulting classifier had very high FP rates. This method also posted higher accuracy rates than Fx.

Overall, all non-ensemble methods tested failed to effectively learn from the features set to detect false positives.

3.6 Probabilistic prediction of route availability

Thus far we have evaluated the performance of the ESVM classifier in providing a deterministic binary prediction of route blockage. The usefulness of this prediction would be improved if it were probabilistic, as a probabilistic prediction of route blockage would enable the selection of the most robust route through terminal airspace. This section extends the binary classifier into a probabilistic one, and validates the resulting predictions. Such a probabilistic forecast will then allow for the optimization of terminal-area fixes and airspace structure, which will be described in Chapter 4.

The ESVM classifier has the following natural mapping into a probabilistic prediction. The classifier consists of N (possibly dependent) ensemble members, each

of which are trained to give a separate SVM prediction of route blockage based on the features of a potential route. This binary prediction can be turned into a probabilistic prediction using logistic regression; the logistic function $f(z) = \frac{e^z}{1+e^z}$ is fit to the decision values of the classifier using maximum likelihood (Hastie et al., 2003, chap. 4.4). The e1071 package for R used in classification computes this automatically (Dimitriadou et al., 2009), giving a set of N probabilities predicting blockage for a given route. We set the ensemble prediction to the mean of these N probabilities. This fusion strategy (equivalent to the sum of experts) was chosen in part because it is a simple and effective fusion strategy that outputs a probability rather than a binary value (Alkoot and Kittler, 1999). Moreover, the mean prediction was found to behave similarly to the majority vote (the fusion strategy typically used in ensemble methods, and used for ESVM). That is, the two fusion strategies disagreed on fewer than 0.5% of datapoints, where a disagreement is defined as a case when the majority vote predicts a route is blocked while the mean vote is larger than 0.5, or vice versa.

Let ESVM-P refer to the ensemble of SVM classifiers modified, as just described, to give probabilistic predictions of route availability using the mean of individual ensemble predictions. Figure 3-11 validates the resulting predictions of ESVM-P, by comparing the prediction given by the classifier against the empirical fraction of open routes, among those route that are given similar predictions.

More concretely, classifier predictions are divided into 20 bins of size 0.05 each. The set of routes for a given set of parameters (terminal model parameters, flight direction, and planning horizon) are then evaluated using ESVM-P, and each is given a probability that the route will be available. The fraction of open routes in each bin represents the empirical probability that a route will be open given its classification probability. The confidence intervals in the figure represent the Agresti-Coull confidence intervals corresponding to each fraction. The black trend line is a smoothing of these empirical probabilities, performed by taking the average of each five-bin window, weighted in proportion of the number of data points in each bin, exactly as described in Section 3.2.1. This smoothed line represents the actual probability that a route will be open, given the classification probability. The gray line is the

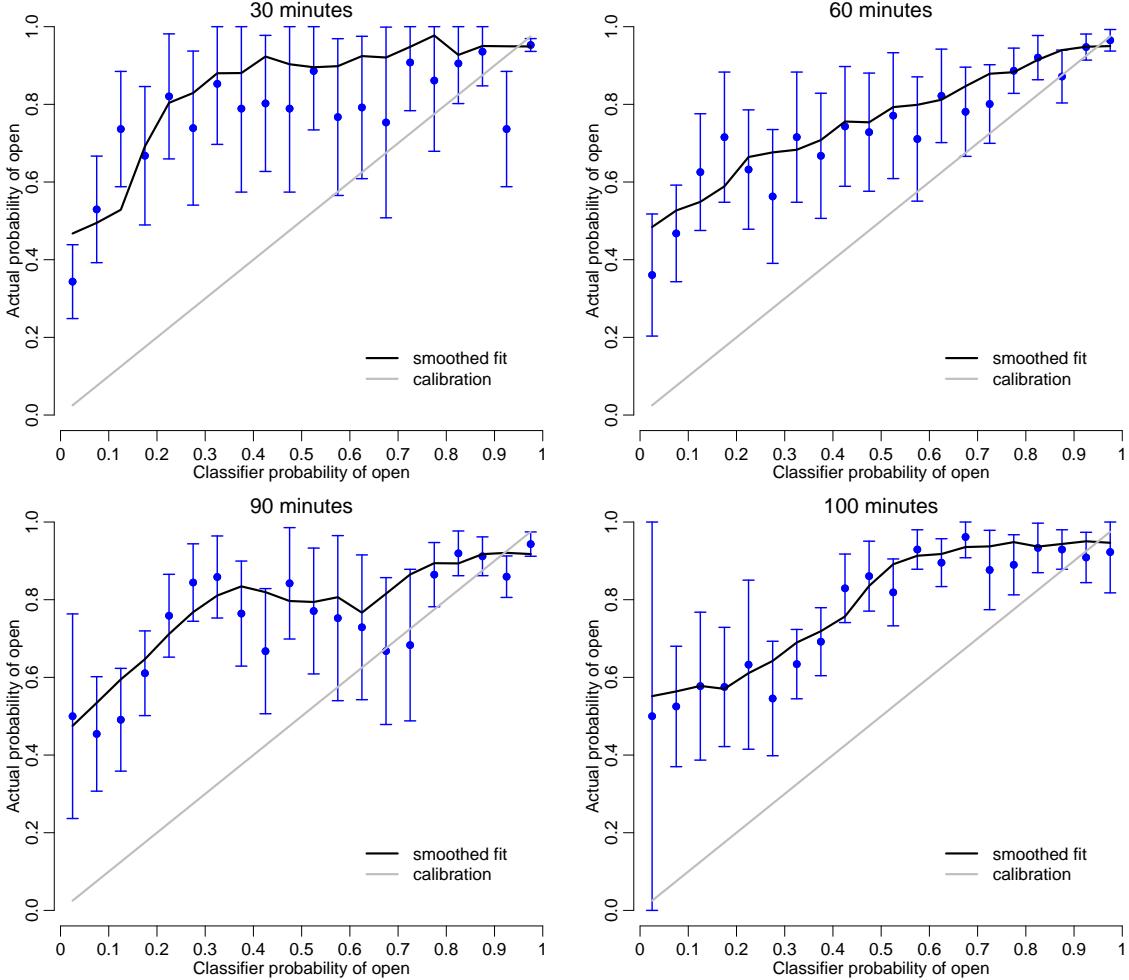


Figure 3-11: Validation of classifier’s probabilistic prediction of route availability, for departures at increasing planning horizons, compared to the calibration line $x = y$.

calibration line at $x = y$. Note that a classifier that were perfectly calibrated would have a smoothed fit that matched this line.

At each time horizon in the figure, the smoothed probability line falls above the calibration line, indicating that the classifier’s probabilistic estimate tends to be a conservative estimate of true probability. This trend is shown for specific parameter settings and departures here, but is representative of other values of the parameters, and arrivals as well. The conservative nature of the fit can be explained by the classification training function. The training places more weight on avoiding false positives than on accuracy (which is dominated by being correct on the open routes), thereby

lowering the confidence on open routes. For this reason, we use the raw classifier probabilities (corresponding to the $x = y$ calibration line) to perform optimization in the next section. It is important to note that any increasing fit of the probability data points would give similar optimization results, and would typically differ only in probability values.

Also noteworthy are the contrasting distributions of data seen in Figure 3-11, indicated by the uneven and varying confidence interval lengths. At the shorter planning horizons, the histogram data is concentrated at the last few bins. Since there is less data (and larger confidence intervals) around the midpoint 0.50, this indicates consensus in the prediction. In contrast, the longer planning horizon of 100-minutes (bottom right) has its shortest confidence intervals concentrated closer to the 0.50 probability point, indicating the presence of lower confidence and lack of consensus amongst the ensemble members, resulting in lower-probability predictions. This trend is seen across a range of parameter settings, and could be explained by the inverse relationship between forecast accuracy and planning horizon.

Having thus validated the probabilistic forecasts of route availability, we can refer to ESVM-P as a **model of route robustness**, because it gives a measure of which routes are likely to withstand uncertainties in the weather forecast and ultimately be open in the weather that materializes.

3.7 Conclusion

This chapter used techniques from machine learning to develop a **model of route robustness**. The model predicts, given the weather forecast, the probability that a route through terminal airspace will be open.

The chapter began with the introduction of eleven features of the weather forecast that were identified for their potential to predict route blockage. The individual features were shown to give reasonable (binary) predictions of blockage using the empirical conditional probability of blockage given the feature value, and were also evaluated using mutual information. They were next combined and used to train several classification algorithms, of which the the ensemble of SVM (ESVM) classifier

outperformed the others in terms of maximum recall (a measure of how well the classifier performed on blocked examples), when tested against a set of actual weather scenarios. In addition, ESVM was shown to outperform the raw CIWS weather forecast in terms of recall. A tradeoff was demonstrated between the recall and accuracy of the classifiers. In the final part of the chapter, the binary ESVM predictions were mapped into probabilities, and the resulting prediction probabilities were validated and shown to be conservative predictions of route blockage. That is, if the classifier predicts that route r will be open with probability 0.8, we can expect that r will be open *at least* 80% of the time.

Our approach to using weather forecasts to predict route blockage, and evaluating the resulting classifier against actual weather, is an important first step towards the realistic integration of weather forecasts with traffic flow management algorithms and decision support tools. The remainder of the thesis builds on top of this route robustness model, and introduces new models and algorithms with the potential to increase airspace capacity, improve throughput, and reduce delays during adverse weather conditions.

Chapter 4

Dynamic Reconfiguration of Terminal Airspace

In this chapter we incorporate the route robustness model developed in Chapters 3 into air traffic management decision making. We focus on the concept of Dynamic Airspace Configuration (DAC), and investigate methods to make changes to terminal airspace structure in the presence of uncertain weather conditions, with the goal of increasing airspace capacity.

The central questions investigated are: given a weather forecast, how can the terminal-area be restructured to minimize disruptions to scheduled airspace usage?, and 2) can minor adjustments be made to existing airspace structure (for instance, by moving an airspace route or sector boundary) in order to avoid or mitigate the effects of blocked airspace?

This chapter is organized as follows. We begin by motivating the operational concept of adaptable airspace, and describe key differences, in terms of goals and constraints, between enroute and terminal DAC. Next, we introduce and evaluate an algorithm for optimally placing terminal fixes and routes, without making any other changes to airspace structure. We build upon this model by gently relaxing the boundaries of terminal sectors, and develop an integer programming formulation to select optimal fixes and routes, as well as sector boundary placements. All the proposed algorithms are evaluated using real weather scenarios.

4.1 Motivation for dynamic terminal airspace

As described in Chapter 1, Section 1.2.3, DAC algorithms strive to restructure the NAS in ways that allow air traffic control to better manage aircraft flows. Although past research has largely focused on enroute airspace in clear weather conditions, the principle of better matching airspace structure to ambient conditions also has the potential to benefit airport terminal-areas, which are the bottlenecks for NAS traffic flow. In this section, we motivate the benefits of DAC in terminal airspace, and describe the unique challenges that present themselves in the terminal setting.

4.1.1 Terminal airspace structure and air traffic constraints

Figure 4-1 shows the terminal airspace structure for ATL with and without traffic overlaid. All traffic data is taken from the Enhanced Traffic Management System (ETMS). Figure 4-1(a) depicts the terminal sectors and routes, with the four STARs in green, and the corresponding waypoints indicated by green triangles. SID fixes are indicated by red triangles. Airspace at ATL is shown to have a four corner post configuration, as is typical of non-metroplex airports in which one airport is the dominant player in surrounding airspace. That is, airspace is divided into four arrival sectors alternating with four departure sectors, each containing an arrival or

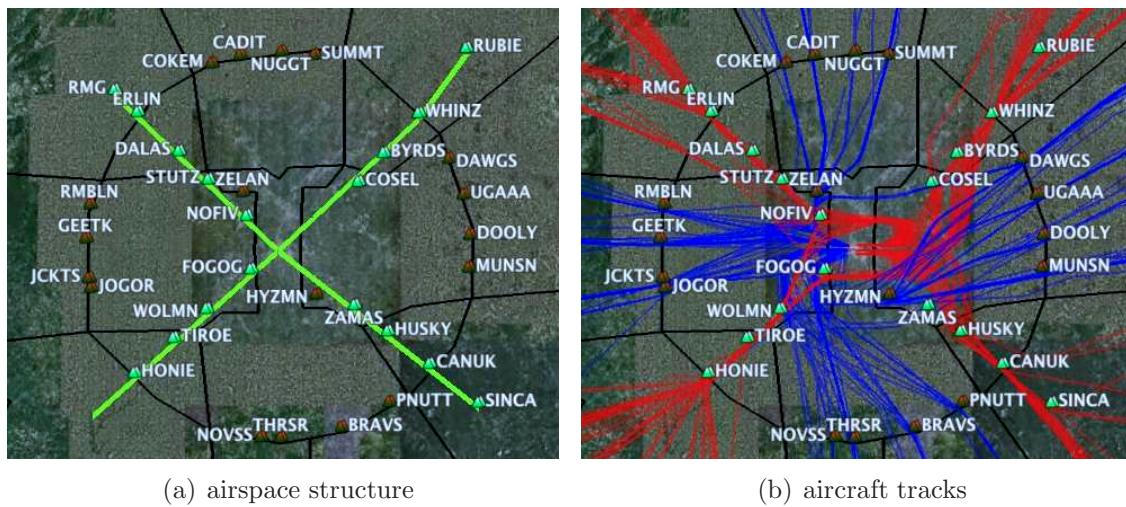


Figure 4-1: Terminal airspace structure for ATL, typical of other four corner-post airports. © Google 2010, Image U.S. Geological Survey, USDA Farm Service Agency.

departure gate along with one or more fixes per gate. Note that each departure gate contains four departure fixes. The TRACON boundary is approximately 75 km away from the airport, although the STARs begin farther out than that, at about 100 km. Figure 4-1(b) shows air traffic overlaid onto the airspace structure, and illustrates the designed separation between arrivals (red) and departures (blue) in the outer section of the terminal, and the merges that occur closer to ATL.

Although traffic is organized and well-structured during normal operating conditions, the situation becomes much more complex when the terminal airspace is affected by convective weather. Such scenarios are encountered, for example, when a STAR is blocked by weather. The next section illustrates the disruptions to nominal traffic patterns during convective weather events.

4.1.2 Terminal airspace operations during convective weather

This section shows visualizations of traffic operations when the terminal airspace is impacted by convective weather, and discusses the operational challenges and potential inefficiencies.

Figure 4-2 compares traffic patterns on a nominal day with clear weather conditions against three separate scenarios in which blocks of airspace and waypoints along standard routes are blocked by convective weather. Each visual contains a snapshot of five minutes of traffic, with arrivals in red, departures in blue, and airspace fixes as black diamonds. Each track contains a bubble at its final position to indicate the direction of flight.

In Figure 4-2(a), traffic travels directly over terminal waypoints and strictly follows the corresponding STARs and SIDs. Arrival traffic is present at each of the four STARs and departure traffic uses all departure fixes. This changes in the three weather-affected scenarios that follow.

In Figure 4-2(b) we see that both southern STARs are unused, even though the southwest one is weather-free. This could be due to upstream weather in enroute airspace not shown on the zoomed-in image, or could be due to forecast error which had predicted greater impact on the southwest STAR. There are also substantial

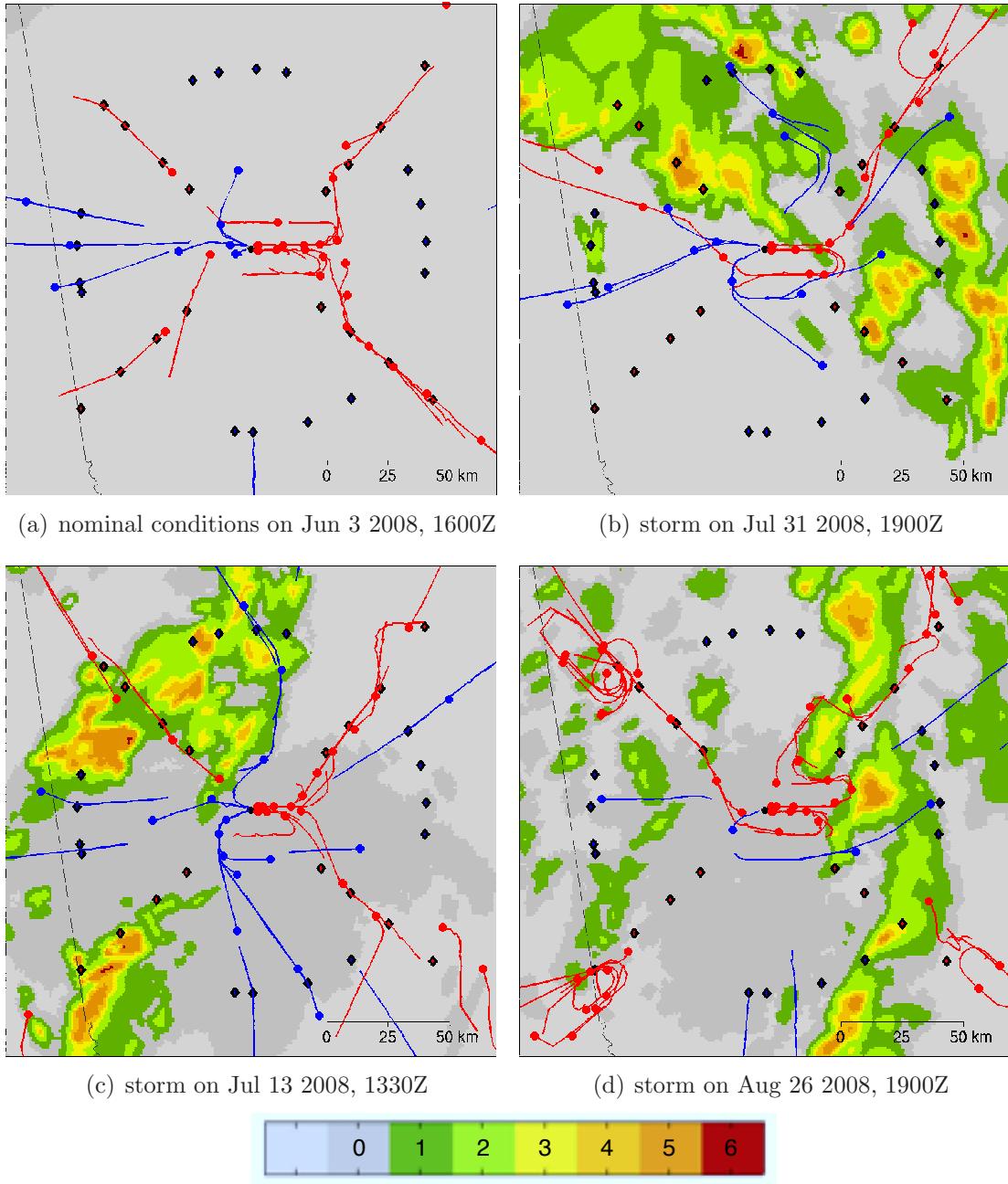


Figure 4-2: Comparison of air traffic flow near ATL during nominal and convective weather conditions.

deviations along the northwest STAR, well into departure airspace, and there are dramatic deviations of northbound departures.

Figure 4-2(c) shows similar behavior. The southwestern STAR has no traffic although there is a gap in the weather in the adjacent airspace. The northwest STAR has a steady stream of arrivals entering through a very narrow break in the weather.

Departures are open in all directions, although northbound departures are deviating around weather cells.

Finally, Figure 4-2(d) shows widespread holding of arrivals at the TRACON boundary. The southwest fix in particular is interesting because the arrival route is clear yet aircraft continue to hold. In contrast, the northeast STAR is in use despite large weather cells nearby, causing aircraft to deviate through a narrow gap in the weather, before entering the downwind leg phase of their approach. Departures seem to be slowed down in all directions.

To improve our understanding of terminal air traffic operations during thunderstorms, we interviewed several air traffic controllers in the United States and Europe. Controllers mentioned that flight deviations are typically ad hoc and done on an aircraft-by-aircraft basis. Specifically, an aircraft will follow its filed flight plan and if the pilot sees a weather cell that looks hazardous, he/she will request a deviation. ATC will respond by trying to accommodate that deviation, and may suggest the same deviation to later flights. During weather events when arrivals are in danger of deviating, departure operations are often slowed down by the TRACON in order to ensure separation (which may have to be temporal), and to limit controller complexity. Appendix A contains details of the air traffic controller interviews.

In summary, despite rigid airspace and route structure, once an aircraft is in the air with a filed flight plan, air traffic controllers do allow deviations around weather cells when requested by pilots. In effect, the current operational environment *already has* some flexibility in changing routes. Disruptions from nominal operations include aircraft crossing into adjacent sectors, deviating from standard arrival and departure routes, and holding when airspace is seemingly available. These disruptions increase the complexity of air traffic control operations and are often accompanied by a decrease in airspace throughout. Clearly, there exist tactical opportunities to recover capacity currently lost due to stormy conditions.

4.1.3 Comparison of terminal and enroute Dynamic Airspace Configuration

Although researchers have studied algorithms for increasing airspace capacity through DAC, previous research has primarily focused on enroute airspace. In this section we discuss the substantial differences in the constraints and objectives of a good sectorization and airspace design, for enroute versus terminal airspace.

Typically, DAC algorithms for enroute airspace select a set of convex and connected sectors which minimize controller complexity constraints (Delahaye and Puechmorel, 2006; Yousefi, 2005; Basu et al., 2008). Aircraft flying through a given enroute sector may cross in several different directions, and there may be several jetroute intersections, sometimes merging more than two traffic flows.

In contrast, aircraft trajectories in the terminal-area are less complex in certain ways. They can be modeled by line segments, without turns, from the terminal fix to the airport (or vice versa in the case of departures), with all turning, merging, and crossing confined to an inner circle very close to the airport. This simplifies the convexity and connectivity constraints of the sectorization problem, and allows airspace sectors to be constrained to pie slices.

Besides these inner merges, aircraft crossings are rarely allowed in the terminal-area, and arrivals and departures are kept in separate airspace to minimize complexity and maintain safety.

The typical enroute sectorization constraint of balanced workload between sectors is not quite relevant for terminal airspace. During any given time interval (say, 30 minutes), there is an inherent imbalance of arriving and departing traffic in terminal airspace, as a result of the banking of operations. This means that the load balancing of controller workload among all sectors is not an objective (although it may be desirable to spread the arrival demand across all *arrival* sectors). However, controller workload is still an important factor in the terminal-area, and can be decreased, for instance, by limiting the deviations from existing airspace structure.

A more appropriate objective for the terminal-area is that of meeting demand

by (for example) expanding sectors when arrival demand is larger than departure demand, or by moving sectors or routes during periods of weather activity. In the face of weather, a predicted storm cell may render an entire sector (or more) impenetrable by pilots, and it is desirable to mitigate the complexity and capacity hit of such situations.

The remainder of this chapter focuses on algorithms for terminal DAC which consider these unique constraints and objectives of terminal airspace.

4.2 Terminal-area DAC model setup

Motivated by existing inefficiencies in air traffic operations, there are several ways in which the terminal airspace can be restructured to optimize capacity in the presence of hazardous weather, with varying degrees of complexity:

- 1 move standard routes and corresponding fixes, keeping sector boundaries fixed,
- 2 move standard routes, corresponding fixes, and allow for renegotiation of sector boundaries, and
- 3 design airspace *from scratch*.

These strategies are ordered in terms of increasing changes to existing operations, and hence increasing complexity to traffic controllers (as compared to current practice).

In this chapter we begin with approach 1 and study the potential benefits of even a small amount of operational flexibility. We fix all sector boundaries and move fix locations and corresponding routes within each sector so as to minimize the probability that each route is blocked by hazardous weather. Keeping the sector boundaries fixed limits the additional complexity and air traffic controller workload that arise from redefining sector boundaries. Approach 2 is explored later in this chapter, and Approach 3 is studied in Chapter 5.

4.2.1 Terminal airspace sectorization model

This section introduces the terminal-area model used in this chapter, which is based on the one used for the route blockage model in earlier chapters.

Consider the model depicted in Figure 4-3, which represents the terminal airspace T using two concentric circles, as seen previously: an outer circle C_O of radius R_O , and an inner circle C_I of radius R_I . C_O represents the points at which arriving (departing) aircraft enter (exit) the terminal airspace. The inner circle C_I represents the points at which aircraft start their final approach into the airport and perform any merges or other maneuvers.

The dashed gray lines represent the division of T into a set S of m sectors, where each $s \in S$ contains a fix, whose position and direction (either arrival or departure) are indicated by the placement of the gray arrow. The solid line in the southeast arrival sector indicates the route that aircraft take from the outer fix to the airport, in that particular sector. Note that these routes, as well as all sector boundaries, lie along radii of the circle C_O , and are of length $R_O - R_I$.

4.2.2 Route robustness model

We use the model of route robustness developed in Chapter 3 and finalized in Section 3.6 as an input to the terminal DAC algorithms. Thus, for a route r through the terminal airspace, we have a probability \hat{p}_r that the route will be open (with some wiggle room) when the actual weather materializes.

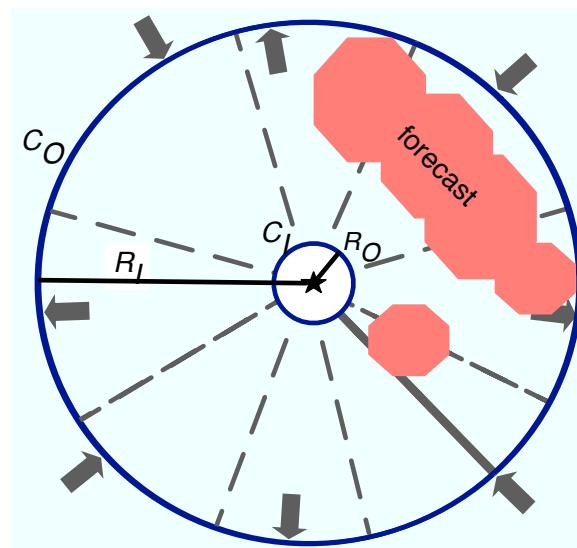


Figure 4-3: Model of terminal airspace with standard sectors and fixes

4.3 Model 1: Dynamic terminal routes

In this section, we consider the problem of selecting one route and outer fix for each sector in terminal airspace at tactical planning horizons, while keeping sector boundaries fixed. The goal is to select routes which will be flyable by pilots in actual weather conditions.

The setup in Section 4.2 gives rise to a natural and simple algorithm for moving terminal routes and associated fixes in the face of disruptions from convective weather: select the maximum probability route and associated fix for each sector. We next give details of this approach and analyze operational gains.

4.3.1 Algorithm Description

Consider the following algorithm, which given a planning horizon t_0 and a weather scenario, selects terminal routes and fixes within each sector independently:

For each sector s :

1. Generate a set of potential routes and corresponding fixes by sampling straight-line routes at incrementally increasing angles from the airport. Each potential route begins at R_I and ends at R_O ; the intersection of the route with C_O defines the associated fix.
2. For each potential route r , evaluate \hat{p}_r , the probability that r will be open given the weather scenario, planning horizon t_0 , and route direction (either arrival or departure, depending on s).
3. Output the route r^* with maximum probability of being open (as long as the probability exceeds 0.50). Ties are broken by picking the route that is closest to the standard route. If all probabilities corresponding to a given sector are less than 0.50, declare s blocked.

The algorithm is depicted visually for one departure sector in Figure 4-4. As shown, the SID route and corresponding standard departure fix may be replaced by a route and associated fix that is more likely to be open, chosen from a set of potential fixes.

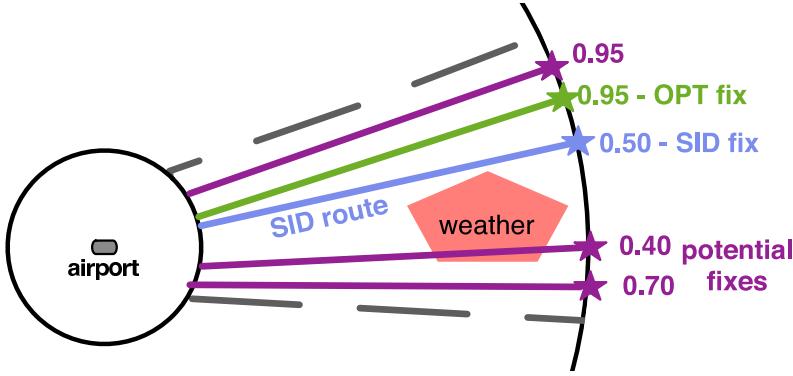


Figure 4-4: Selection of an optimal route and associated fix for one sector, as a function of the weather forecast and standard fix location.

Since sectors are optimized independently, and since there is an implicit B km of wiggle room in the actual route (and fix) flown, the algorithm as described does not necessarily maintain separation between adjacent routes. To ensure strict separation of at least $2B$ km, it is necessary to simultaneously solve across all sectors. Furthermore, the perturbed routes may violate sector boundaries, especially for larger values of B . However, these violations are rare in our experiments; there were no violations of route separation, and a single boundary crossing violation of under 1 km, among the 224 cases studied in the next section. For simplicity we do not consider them further.

4.3.2 Algorithm Analysis

The route movement algorithm was tested on the Data2008 dataset consisting of 28 weather scenarios. The parameters were set to $B = 8$ km, $R_I = 20$ km and $R_O = 100$ km, and the route robustness classifiers were trained on the independent dataset Data2007.

Table 4.1 shows the overall performance of the algorithm for varying planning horizons, and for arrival and departure sectors. Each row corresponds to one planning horizon and direction combination, and represents 112 data points (28 weather scenarios, each with 4 sectors per flight direction). The computed metrics reflect the effectiveness and trade-offs of the dynamic fix movement algorithm.

The first metric reported, movements, refers to the percentage of fixes moved,

t_0 (min)	move- ments (%)	fix classi- fied blocked (%)	fix blocked given classified blocked (%)	potential avoid- able blockage (%)	avoided blockage (%)
arrivals	10	21	18	50	75
	30	20	18	55	55
	60	19	22	32	48
	90	32	24	30	44
	100	34	36	23	48
departures	10	22	21	43	65
	30	29	26	34	59
	60	34	32	25	72
	90	36	32	25	58
	100	48	40	24	51

Table 4.1: Overall results for dynamic route and fix movement algorithm.

and gives a measure of how often the algorithm recommends an alternate fix. This number increases with planning horizon, and tends to be larger for departures than for arrivals. Note that when we refer to a fix (movement, blockage, etc), we are implicitly referring to the fix *and associated route*. We use this convention during the remainder of this analysis, for simplicity.

The second metric, classified fix blockages, refers to the percentage of (original) fixes for which the classifier predicted blockage. This number increases with increasing planning horizon. Of course, a predicted fix blockage does not necessarily mean the fix will actually be blocked, and this situation is captured in the next metric, the percentage of actual blocked fixes given that the fix is predicted to be blocked. Here we find that the longer planning horizons are accompanied by lower values, indicating that the classifier's prediction is less accurate at longer planning horizons.

Potential avoidable blockage shows the percentage of predicted-blocked fixes for which the algorithm recommends an optimal fix (which is predicted to be open). We find that at shorter planning horizons, the potential to avoid blockages is predicted to be greatest, staying well above 50%, meaning that the algorithm gives an alternate routing possibility more than half the time.

Finally, avoided blockage refers to the percentage of predicted-blocked fixes for

which the optimal fix recommended by the algorithm is open in actual weather. This number tends to decrease with planning horizon, and tends to be higher for departures than arrivals. The gap between the last two columns gives a measure of accuracy on predicted-blocked routes, though it does not distinguish between fixes assigned a 0.90 probability of being open and those with 0.60 probability. Clearly the accuracy should depend on these probabilities, and we explore this correlation later.

Table 4.2 provides a closer look at the standard routes and associated fixes that are moved to some optimal route and fix by the algorithm. When a route is moved, there are four possible outcomes: the original route and the optimal route are both open in the observed weather (Open/Open), both are blocked (Blocked/Blocked), the original route is blocked while the optimal is open (Blocked/Open), or the original route is open while the optimal is blocked (Open/Blocked). Ideally, we'd like to minimize Blocked/Blocked and Open/Blocked, the cases where the algorithm recommends an unavailable route.

The table indicates several trends. First, Open/Open accounts for more than 63% of route movements across all categories, while Open/Open and Blocked/Open together account for more than 76% of movements, indicating that the optimal route is usually likely to be at least as good as the original. A movement of a route that turns out to be open may seem undesirable, but since the confidence in the optimal

t_0 (min)	number of movements	original / optimal			
		Open/ Open	Blocked/ Open	Open/ Blocked	Blocked/ Blocked
arrivals	10	24	15	4	0
	30	22	16	2	0
	60	21	16	0	2
	90	36	31	1	0
	100	38	33	0	0
departures	10	25	18	6	1
	30	32	25	3	4
	60	38	31	3	2
	90	40	33	3	4
	100	54	47	2	3

Table 4.2: Analysis of route movements.

route and associated fix is greater than the original (it is associated with a higher probability of being open), the optimal route is the more conservative choice and has higher expected capacity.

There are very few data points in the other three categories, indicating possibly large sampling error, so we only perform modest analysis of these cases. At the 10-minute planning horizon for both arrivals and departures, it is a good decision to move the route, which is consistent with the short-term accuracy of the pixel-based forecast. Thus, tactical decisions to move fixes can be relied on, although more care and validation must be pursued at longer and more strategic planning horizons.

Table 4.3 shows algorithm performance as a function of classifier prediction probability \hat{p} , and provides a validation of the probability estimates associated with each route and fix movement recommendation. Each column lists, for a set of input parameters and for a range of values of \hat{p} , the empirical percentage of optimal fixes that were open in actual weather.¹ For instance, for arrivals at a 90-minute planning horizon, among those optimal fixes (corresponding to actual weather scenarios) that were predicted to be open by the classifier with probability between 0.95 and 1.00,

¹Note that the data points represented by Table 4.3 are not just those from Table 4.2, but include the recommended route from all sectors for which the algorithm finds an open route (all routes r such that $\hat{p}_r > 0.5$).

t_0 (min)	% open given $\hat{p} \in (0.95, 1.00]$	% open given $\hat{p} \in (0.75, 0.95]$	% open given $\hat{p} \in (0.50, 0.75]$
arrivals	10	97.92	-
	30	94.38	-
	60	-	91.46
	90	96.30	88.90
	100	-	95.35
departures	10	96.00	-
	30	97.53	-
	60	98.11	87.50
	90	91.05	88.46
	100	-	92.31
			88.24

Table 4.3: Dynamic route movement as a function of the predicted probability of being open, \hat{p} .

96.30% of them were open in the actual weather that materialized. Blank entries correspond to cells with fewer than 13 data points, which were removed to eliminate cells with sample variance over 0.05.

As was noted in Section 3.6, the longer the forecast planning horizon, the lower the likelihood of high-probability predictions from the classifier. This explains the uneven distribution of data points among the three probability levels studied. The table shows that the percentage of open routes tends to stay within the predicted percentage when there are enough data points. This means that decision-makers can be more confident in fixes that have a high probability of being open. The table also shows that the validation is less accurate with increased planning horizon and with decreased probability interval, as expected based on the classification results.

Thus, the predicted probabilities correlate well with actual rates of route availability, and can be used to inform route movement decisions in marginal weather conditions.

4.3.3 Stability of route selection

In the fix relocation algorithm as described, all recommended route movements and associated fixes are designed for aircraft that begin their flight through the terminal at a specific time t . The algorithm is dynamic in the sense that changing weather conditions are incorporated into that specific journey through the terminal. However, as a weather scenario evolves, the same fix will not necessarily be optimal (or robust) for flights that enter the terminal later, say, at $t + 20$.

In practice, it is necessary to decide *when* to initiate a dynamic fix movement, and to understand the frequency and scale of any subsequent adjustments as weather evolves. This section introduces a dynamic variant of the fix relocation algorithm (DYN from now on), which allows us to analyze the stability of route recommendations.

Given a weather scenario with starting time t , planning horizon t_0 , and fixed parameter settings as before, we run the algorithm presented in Section 4.3 for times $t, t + 5, \dots, t + 55$, using the planning horizon t_0 in each iteration. Let the i 'th time

period be denoted by T_i . We break any ties among potential optimal routes during period T_i by selecting the route closest to the optimum from period T_{i-1} . The first time period uses the standard fix as a tie breaker, as in the original algorithm.

For each sector s , let $\hat{p}_i^*(s)$ be the probability (from the route robustness model) that the optimal fix for sector s is open in period T_i . Let $D_i(s)$ be the displacement in degrees of this optimal fix (in period T_i) from the optimal fix in the previous time period, T_{i-1} . Let n be the number of data points (sectors optimized). We run DYN for all Data2008 weather scenarios ($n = 224$), with parameters set to $t_0 = 60$ minutes, $B = 8$ km, $R_I = 20$ km, and $R_O = 100$ km.

Figure 4-5 shows the fix locations and associated routes recommended by DYN for one departure sector, as a weather scenario evolves over six consecutive periods (i.e., 30 minutes). The top row contains the (dynamic) weather forecast, with the route and fix from the previous time-period in magenta, and the new recommended optimal route and fix in blue. The bottom row contains the actual weather corresponding to each time period, as well as the perturbed route closest to the optimal, in blue. This perturbed route is not necessarily a straight line, and merely indicates feasibility by avoiding weather hazards. In other words, the leftmost images correspond to the route and fix location (in blue) assigned to flights that arrive in the first 5 minutes

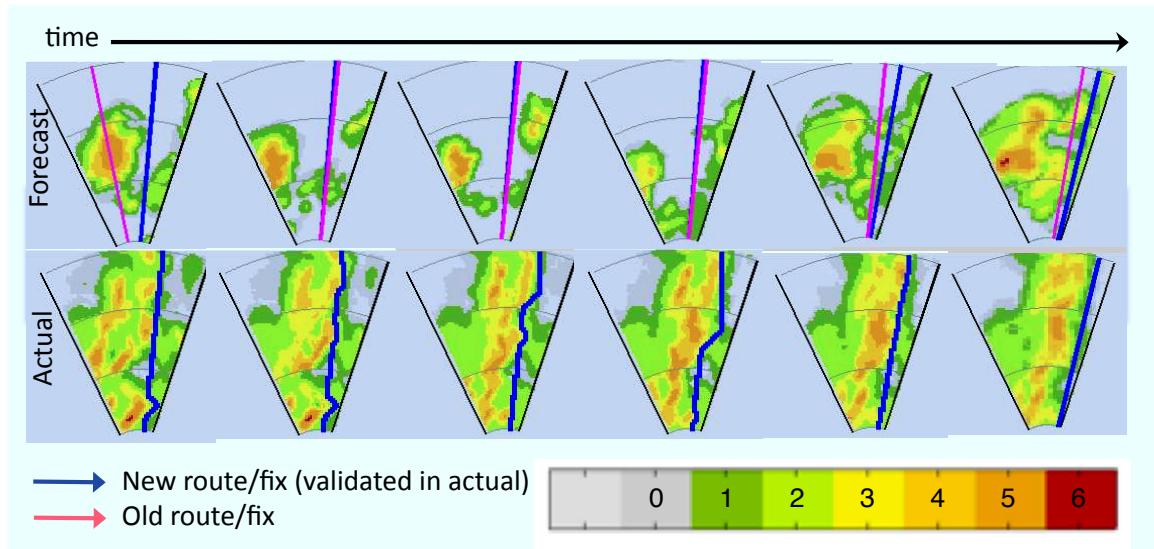


Figure 4-5: Example of dynamic route movement for six consecutive time periods.

of the scenario, the images immediately to the right of that correspond to the fix location and routes for flights that arrive in the next 5 minutes, and so on.

This scenario is illustrative of the typical behavior of a given sector. The first recommended movement (or the first movement after a fix blockage) is the largest, and subsequent movements are much smaller (or nonexistent). To understand the stability of routes under the algorithm more deeply, we next analyze results for the entire dataset.

Table 4.4 summarizes the behavior of DYN between two (arbitrarily chosen) consecutive periods, showing how recommended fixes tend to evolve with weather. Of note are the results for high and low-probability fixes. For a sector whose optimal route has probability over 0.95 (which accounts for 74% of all sectors), the optimal route for the next time period is open 99.4% of the time, and averages a movement of 1.2 degrees (equivalent to 2.4 km at the outer circle). Furthermore, a blocked sector (in which $\hat{p}^* < 0.5$, accounting for 12% of all points) remains blocked 86% of the time during the next time period, and in the case that it opens up, the corresponding route movement averages to 14 degrees. As was demonstrated in the visual example, this confirms that displacements tend to be few and small in most cases, and grow larger after a blockage.

Since consecutive time periods are not expected to be independent (fast-moving storms are likely to be accompanied by more or larger route movements during all 12 time periods), Table 4.4 does not paint the whole picture. Across all twelve time periods, the average sector is blocked for 1.4 periods, and contains 2.4 fix movements.

predicted probability of optimal fix in T_6 : \hat{p}_6^*	n (%)	% of time sector is predicted blocked in T_7	mean $\mu(D_7)$ (deg)	std. dev. $\sigma(D_7)$ (deg)
$0.95 < \hat{p}_6^*$	74	0.6	1.2	4.2
$0.90 < \hat{p}_6^* \leq 0.95$	6	7.7	2.3	6.9
$0.50 < \hat{p}_6^* \leq 0.90$	8	15.8	7.8	14.3
$\hat{p}_6^* \leq 0.50$	12	85.2	14.0	16.0

Table 4.4: Stability of route movements between consecutive time periods T_6 and T_7 .

The average movement size across all fix movements (including the first time period) is 9.5 degrees, equivalent to 16.6 km at the outer terminal boundary C_O . Thus we can conclude that over a typical 1-hour period, fixes tend to remain open, especially once a high-probability fix is identified.

In summary, the dynamic approach to moving routes during convective weather is applicable to TRACON operations for non-metroplex airports, which have the type of simple arrival and departure routes studied. Moving fixes during severe weather scenarios has the potential to increase terminal capacity and to improve controller workload as far as dealing with pilot deviations, since target fixes are more likely to be flyable.

4.4 Model 2: Dynamic terminal routes with renegotiation of sector boundaries

The previous section considered the problem of improving air traffic operations by making dynamic changes to standard terminal routes and fixes when the standard structure is likely to be affected by convective weather. In this section, we build upon the previous approach and consider the additional benefits when we allow for gentle changes to sector boundaries in addition to dynamic changes to standard routes and fixes.

We first develop an integer programming formulation for the problem which assigns airspace sector boundaries, terminal routes, and fixes such that the selected routes are predicted to be open during convective weather activity. The model also limits deviations from existing airspace layout. We then apply the model to real weather scenarios and analyze the results.

4.4.1 Integer programming formulation

The integer program (IP) essentially partitions the terminal airspace into pie slices and assigns sector boundaries and routes to these slices, subject to constraints on the maximum displacement from the original boundaries and the minimum distance

between adjacent routes, as depicted in Figure 4-6. The IP strives to select routes with high probability of being open (any route selected must be forecast to be open) while at the same time limiting deviation from the original airspace layout.

Input

We start with terminal airspace T partitioned into n pie slices, each corresponding to a *potential route* and associated fix (a route is an (undirected) line segment between C_I and C_O , along a radius of R_O). The set of these potential routes (and fixes) is $F := \{1, \dots, n\}$, where the number n of potential routes should be chosen to be large enough to provide many options (say, 360 for a terminal-area), but small enough so that each route is at least 1 km wide at the outer boundary (the granularity of the weather forecast). We are also given a set $S := \{1, \dots, m\}$ of sectors.

Because the terminal-area is circular, route 1 is adjacent to route n geometrically (distance is 1). To deal with this “wrap around” effect in the formulation, we introduce an augmented set of n^+ wedges $F^+ := \{1, \dots, n^+\} \supset F$, and an augmented set of sectors $S^+ := \{1, \dots, m + 1\} \supset S$. The number n^+ of wedges must be at least n plus the maximum number of wedges in a single sector.

The final input is a weather forecast for T corresponding to a specific date and

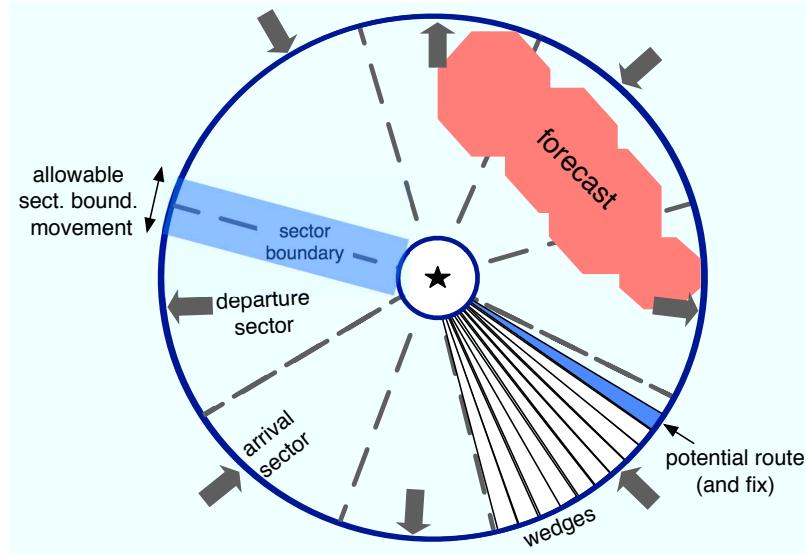


Figure 4-6: Model of terminal airspace with standard sectors divided into wedges and allowable range of sector boundary locations.

planning horizon t_0 .

Parameters

The IP uses the following parameters:

For each $s \in S$ and $i \in F^+$, the weather forecast is given in terms of blockage probability \hat{p}_{si} , the probability that i will be open in s . Note that arrival and departure routes may have different likelihoods of blockage, so \hat{p}_{si} depends on the direction of sector s .

For each $s \in S$ and $i \in F^+$, $d_{s,i}^{\text{rte}}$ is the distance of wedge i to the original route for sector s , while $d_{s,i}^{\text{sect}}$ is the distance of wedge i to the original boundary for sector s . The boundary of sector s always refers to the clockwise-first boundary (or, the boundary with minimum wedge number, mod n), so that sector s extends from its boundary to the boundary of sector $s + 1$. All distances in this formulation are in number of wedges.

Parameter K is the maximum displacement of a sector boundary, parameter L is the minimum distance between any two routes, and M is a large constant. Finally, the parameters α , β , γ , and λ are used to control the weight given to the various objectives (discussed in Section 4.4.1).

Variables

variables are

$$x_{si} = 1 \text{ iff sector } s \text{ is assigned wedge } i$$

$$y_{si} = 1 \text{ iff the boundary of sector } s \text{ is at wedge } i$$

$$z_s = 1 \text{ iff sector } s \text{ is open}$$

where $s \in S^+$ and $i \in F^+$.

Objective

Four main objectives are desirable in the sectorization of terminal airspace:

- 1 maximizing the probability that the selected terminal routes are open,
- 2 limiting the distance between the new sector boundaries and their default locations,
- 3 limiting the distance between the new routes and their default locations, and

4 keeping each sector open if feasible.

These objectives can conflict with each other, since a high-probability route in sector s may only be feasible (depending on the weather forecast) if the boundary of sector s is moved clockwise, rendering sector $s + 1$ blocked (say, if all remaining potential routes are blocked: $\hat{p}_{(s+1)i} \leq 0.5 \forall i \in (s+1)$). Thus, we create a linear combination of these objectives, and explore trade-offs in Section 4.4.2. The overall objective function is therefore to minimize:

$$\sum_{s \in S} \left[\sum_{i \in F^+} \alpha x_{si} \hat{p}_{si} - \beta y_{si} d_{s,i}^{\text{sect}} - \lambda x_{si} d_{s,i}^{\text{rte}} \right] - \gamma (1 - z_s) \quad (4.1)$$

In (4.1), the parameter α is the weight given to the first objective, namely, maximizing the probability that the selected routes are open, β and λ are the penalties for deviation from the default sector and route locations respectively, and γ is the penalty for closing down a sector, along with its associated arrival or departure route.

Constraints

The first two constraints ensure that each sector has exactly one boundary, and that a route is selected for each open sector, respectively.

$$\sum_{i \in F^+} y_{si} = 1 \quad \forall s \in S \quad (4.2)$$

$$\sum_{i \in F^+} x_{si} = z_s \quad \forall s \in S \quad (4.3)$$

The next set of constraints ensure that an optimal route and associated fix f^* for sector s is feasible for s : f^* is contained within s , and f^* is forecast open ($\hat{p}_{sf^*} \geq 0.5$).

$$\sum_{i \in F^+} iy_{si} \leq \sum_{i \in F^+} ix_{si} + M(1 - z_s) \quad \forall s \in S \quad (4.4)$$

$$\sum_{i \in F^+} ix_{si} \leq \sum_{i \in F^+} iy_{(s+1)i} \quad \forall s \in S \quad (4.5)$$

$$0.5z_s \leq \sum_{i \in F^+} x_{si} \hat{p}_{si} \quad \forall s \in S \quad (4.6)$$

The next two constraints ensure that routes are at least L wedges apart, and that sector boundaries are moved no more than K wedges from their default locations, respectively.

$$L \leq \sum_{i \in F^+} ix_{(s+1)i} - \sum_{i \in F^+} ix_{si} + M(1 - z_{s+1}) \quad \forall s \in S \quad (4.7)$$

$$\sum_{i \in F^+} d_{si}^{\text{sect}} y_{si} \leq K \quad \forall s \in S \quad (4.8)$$

Finally, we have constraints to take care of the wrap around effect due to the circular airspace structure by essentially setting sectors 1 and $m + 1$ to be equal. We assume without loss of generality that sector 1 is always defined to have its boundary at wedge $K + 1$, so that its boundary will be kept between 1 and $2K + 1$ by the IP.

$$x_{(m+1)i} = 0 \quad \forall i \in \{1, \dots, n\} \subseteq F^+ \quad (4.9)$$

$$x_{(m+1)i} = x_{1(i-n)} \quad \forall i \in \{(n+1), \dots, n^+\} \subseteq F^+ \quad (4.10)$$

$$y_{(m+1)i} = 0 \quad \forall i \in \{1, \dots, n\} \subseteq F^+ \quad (4.11)$$

$$y_{(m+1)i} = y_{1(i-n)} \quad \forall i \in \{(n+1), \dots, n^+\} \subseteq F^+ \quad (4.12)$$

$$z_{m+1} = z_1 \quad (4.13)$$

Equations (4.9) and (4.10) assign sector $(s+1)$ the same wedge (route) as sector 1 $(\bmod n)$. Equations (4.11) and (4.12) assign sector $(s+1)$ the same sector boundary as sector 1 $(\bmod n)$. Finally, (4.13) ensures that sector $(s+1)$ is open iff sector 1 is open.

4.4.2 Computation of model solution

The IP formulation was tested using data from the same Data2008 dataset as the simpler model discussed in Section 4.3. The stochastic route robustness model was calibrated for a terminal-area with $R_I = 20$ km, $R_O = 100$ km, and $B = 8$ km using the independent Data2007 dataset.

Although the model presented is an integer program, which is NP-hard in general,

the problem size can be kept small in practice, thus eliminating computational issues. As defined, the IP has $2mn + m = O(mn)$ constraints and roughly $7m + 4\frac{n}{m} + 1 = O(m + \frac{m}{n})$ variables. In particular, when $n = 360$ and $m = 8$, the problem size is roughly 5800 variables and 250 constraints. Despite the non-integrality of the formulation (there exist non-integral optimal solutions to the LP relaxation), this turns out to be easily solvable using CPLEX, which consistently solved the problem in under 0.1 seconds on a 2.66GHz processor with 3.8GB of RAM, running 64-bit Fedora 11.

Due to the nature of the multi-objective optimization, we next give a detailed description of results for one setting of the objective function parameters. Afterwards, we look at how the weightings of various objectives affect the sectorization results.

4.4.3 Analysis of results for fixed parameter settings

This section describes results of the model when $(\alpha, \beta, \lambda, \gamma) = (100, 1, 1, 1)$. This parameter setting emphasizes the selection of a robust route (one with high probability of being open), with small penalties for the displacement of routes and sectors from their default positions, as well as a small penalty for a blocked sector.

Figure 4-7 shows diagrams of the sectorization found by the algorithm for two weather scenarios. The magenta arrows correspond to the original (standard) terminal routes (of the indicated direction) passing through the original fixes, while the blue arrows correspond to the new, optimal routes and fixes. The purple arrows represent overlap of magenta and blue, and indicate that the algorithm kept the original route and fix in place.

The top scenario of Figure 4-7 corresponds to weather on July 13 2008 at 1930Z with the forecast corresponding to a 60-minute planning horizon on the left, and actual weather on the right. In this scenario, the algorithm does not move any sectors, but does move fix ROME further away from weather activity, giving it a higher probability of being open. Moreover, two sectors in the southwest are (correctly) declared blocked. All other routes are kept in place, and it turns out these routes are open and clear of convection.

The bottom scenario corresponds to August 26 2008 at 1100Z, with the planning horizon again set to 60 minutes. We see that the sector containing fix ROME is predicted to be blocked. The algorithm makes a small sector boundary adjustment in the northwest sector, illustrating its potential to open up routes and fixes that would otherwise be blocked. The algorithm recommends a move of the sector boundary which results in a route predicted to be open with probability 0.875. Fix CADIT is then moved to the far side of its sector so as to maintain separation. In the observed weather (on the right), this turns out to have been a good decision, as both the new ROME and CADIT fixes (and corresponding routes) are open, while the original

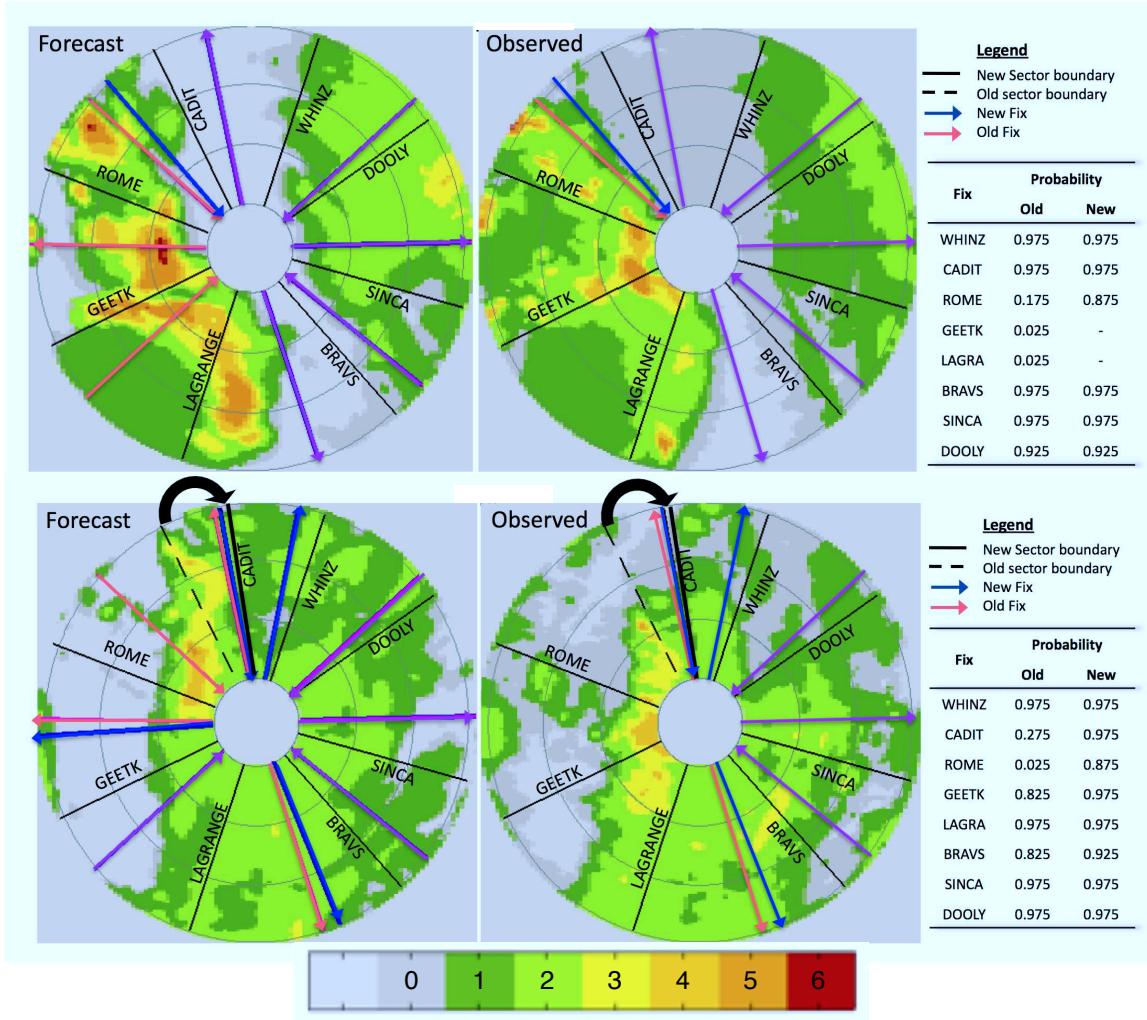


Figure 4-7: Sectorization results for two illustrative weather scenarios, one with a sector boundary shift (bottom), and one without (top).

t_0 (min)	% route move- ments	% sector move- ments	route pred. blocked (%)	route blocked given pred. blocked (%)	potential avoidable blockage (%)	avoided blockage (%)
arrival	10	26	1	19	48	62
	30	28	4	18	55	45
	60	25	5	21	29	54
	90	25	3	24	30	54
departure	10	24	7	21	42	74
	30	33	7	26	31	65
	60	37	4	32	22	71
	90	34	4	31	23	55

Table 4.5: Overall results for terminal route and fix optimization with renegotiation of sector boundaries.

ROME fix is blocked by a large weather cell.

Now that we have a sense of what the algorithm results look like, we concentrate on aggregate results. Table 4.5 shows the overall performance of the algorithm for varying planning horizons, and for arrival and departure sectors. Recall that in this analysis the objective function parameters α , β , λ , and γ are fixed to 100, 1, 1, and 1, respectively. Each row corresponds to one planning horizon and direction (arrival/departure) combination, and represents 112 data points (28 weather scenarios, each with 4 sectors in the appropriate flight direction). The computed metrics reflect the effectiveness and trade-offs of the optimization model.

The first metric reported, route movements, refers to the percentage of routes (and corresponding fixes) moved and gives a measure of how often the algorithm recommends an alternate route. This number tends to be larger for departures than for arrivals, but shows little variance within each direction. The second metric, sector movements, reports the percentage of instances (sector data points) for which a sector boundary was moved. This value is small (under 7% across the board) for this setting of parameters, reflecting the fact that sector movements only occur when there is a large gain in route robustness.

The next metric, forecast route blockages, refers to the percentage of (original) routes which were predicted to be blocked. This number increases with increasing

planning horizon. Of course, a predicted route blockage does not necessarily mean the route will be blocked once weather materializes, and this situation is captured in the next metric, the percentage of actual blocked routes given that the route is forecast to be blocked. Here we find that the longer planning horizons are accompanied by lower values, reflecting the lower forecast accuracy at longer time horizons.

Potential avoidable blockage shows the percentage of predicted-blocked routes for which the algorithm recommends an optimal route (which is predicted to be open). We find that at shorter planning horizons, the potential to avoid blockages is predicted to be greatest. The percentage of avoidable blockages is above 50% in all cases, except for an outlier at arrivals with 30-minute planning horizon, meaning that the algorithm gives an alternate routing possibility more than half the time.

The last column, avoided blockage, refers to the percentage of standard routes (and associated fixes) which are predicted to be blocked *and* for which the algorithm recommends an optimal route which is open in actual weather. It gives an estimate of the potential increase in airspace capacity if the algorithm were implemented. We note that these values are not monotonic in planning horizon. This is likely due to the existence of a “sweet spot” at mid-level planning horizons. That is, at shorter planning horizons the forecast is already quite good and not much can be gained from optimization, while at longer planning horizons there is more error in predictions of open routes by the algorithm (and probabilities of blockage for optimal routes tend to be lower). The sweet spot is at planning horizons in between. Finally, note that this table does not distinguish between routes assigned a 0.9 probability of being open and those with 0.6 probability. The accuracy should clearly depend on these probabilities, and this correlation is explored later, in Table 4.7.

Table 4.6 provides a closer look at the routes that are moved to some optimal location by the algorithm. When a route (and its associated fix) is moved, there are four possible outcomes: the original route and the optimal route are both open in the observed weather (OO), both are blocked (BB), the original route is blocked while the optimal is open (BO), or the original is open while the optimal is blocked (OB). Ideally, we would want that the cases where the algorithm makes a mistake in moving

t_0 (min)	# route move- ments	Original / Optimal				Open/Open + Blocked/Blocked (%)	
		Open/ Open	Blocked/ Open	Open/ Blocked	Blocked/ Blocked		
arrival	10	29	18	6	0	5	83
	30	31	23	4	0	4	87
	60	28	22	1	2	3	82
	90	28	22	1	2	3	82
departure	10	27	17	9	1	0	96
	30	37	29	5	2	1	92
	60	41	34	2	2	3	88
	90	38	34	1	1	2	92

Table 4.6: Analysis of route movements for model 2.

a route, BB and OB, be few in number, while BO (especially) and OO be many.

The table indicates several trends. First, OO accounts for more than 62% of route movements across all categories, while OO and BO together account for more than 82% of route movements, indicating that the optimal route is usually likely be at least as good as the original. A movement of a route that turns out to be open may seem undesirable, but the confidence in the optimal route and associated fix is greater than the original, making it the more conservative and robust choice. There are very few data points in the other three categories, indicating possibly large sampling error, so we only perform modest analysis of these cases. Nevertheless, at the 10-minute planning horizon for both arrivals and departures, and at the 30-minute horizon for departures, it is a good decision to move the route. This is consistent with the validation (in Chapter 2) of the short-term accuracy of $1 \text{ km} \times 1 \text{ km}$ pixel-based forecasts. Thus, tactical decisions to move routes can be relied on, although more care and validation must be employed at longer and more strategic planning horizons.

Table 4.7 shows algorithm performance as a function of prediction probability. The columns give the empirical percentage of open routes, given a range of blockage probabilities (from the classifier) for those routes. The standard error is also reported in parentheses. Blank entries correspond to cells with fewer than 10 data points, which were removed to eliminate cells with standard error greater than 0.10.

t_0 (min)		% open given $\hat{p} \in (0.95, 1.00]$	% open given $\hat{p} \in (0.75, 0.95]$	% open given $\hat{p} \in (0.50, 0.75]$
arrival	10	99.03 (0.01)	-	-
	30	95.96 (0.02)	-	-
	60	91.67 (0.03)	-	-
	90	100.00 (0.00)	89.47 (0.04)	84.62 (0.10)
departure	10	99.02 (0.01)	-	-
	30	95.83 (0.02)	90.00 (0.09)	-
	60	95.29 (0.02)	93.75 (0.06)	-
	90	95.24 (0.03)	92.59 (0.05)	-

Table 4.7: Dynamic route movement of model 2 as a function of the predicted probability of being open, \hat{p} .

The uneven spread of data points among the three probability levels is an artifact of the behavior of the underlying weather model, which is less likely to assign high-probability predictions as the planning horizon increases. The table shows that the percentage of open routes tends to stay within the predicted percentage when there are enough data points. The table also shows that the validation is less accurate with increased planning horizon and with decreased probability interval, as expected based on the behavior of the weather forecast model. Thus, the predicted probabilities correlate well with actual rates of route availability, and can be used to inform route movement decisions in marginal weather conditions.

4.4.4 Analysis of results as objective function varies

This section discusses results of the optimization as the parameters in the objective function vary.

Figure 4-8 shows how four key result statistics are affected as each of the four objective function parameters is varied while the others are fixed. When fixed, parameters are set to values studied in Section 4.4.3: $\alpha = 100$, and $\beta = \lambda = \gamma = 1$. The results focus on departures at a 60-minute horizon, and all other model parameters are unchanged from Section 4.4.3.

The figure shows that parameters α (preferring increases in route robustness prob-

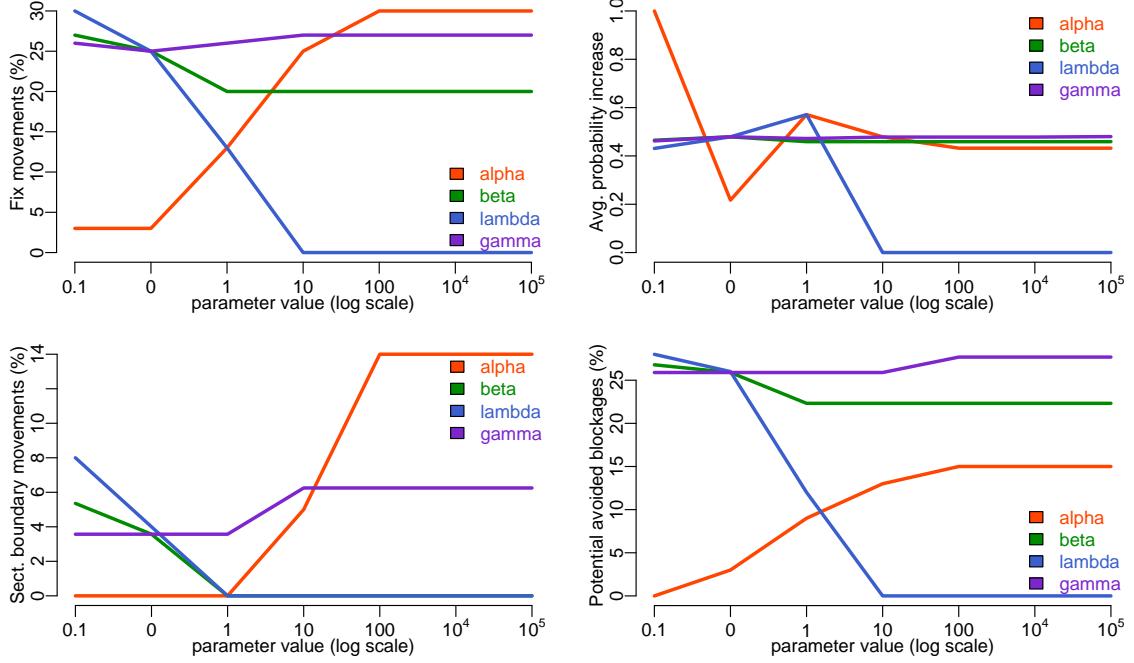


Figure 4-8: Sectorization results as a function of objective function parameters

ability) and λ (penalty on distance of route movement) have the largest effect on the percentage of route movements (top left), the percentage of sector boundary movements (bottom left), and percentage of potential avoided blockages (bottom right). These parameters often work against each other, and a clear trade-off in results is evident in the plots.

In contrast, as γ increases, there is a very modest effect of under 3% on these first three statistics. This can be explained by the relatively small number of total sector boundary movements available – when a route is moved, there is often a high-probability route within the original sector boundaries, making a boundary movement unnecessary. The penalty β on the distance of a sector boundary movement also shows modest effects on these three statistics, for similar reasons.

The top-right plot shows average increase in probability for a route movement, which is surprisingly invariable to the four parameter values. Low values of α show the largest effect, but in general this statistic stays around 0.45. This indicates that when a route is moved, on average there is a high gain in robustness. We note that the lack of variability in this plot could be due to the choice of the fixed parameters.

Overall, we see that the parameters have expected results in key result statistics, and decision-makers can tune these to reflect their preferences.

4.5 Conclusion

This chapter has focused on the development and evaluation of two models for making dynamic changes to terminal airspace structure in order to improve throughput during thunderstorms. Both the simple initial model, which limited structural changes to moving the standard terminal arrival and departure routes, as well as the more sophisticated integer programming model, which made changes to sector boundaries as well as standard routes, were shown to have the potential to avoid periods of standard route blockage and to improve the robustness of the terminal structure.

An interesting follow-on problem is to investigate the benefits of sectorization “from scratch”, similar in spirit to research performed for enroute airspace, where the current sectors are typically not considered as constraints. Furthermore, the models as presented have not considered arrival and departure demand, which certainly has bearing on which routes are more important for traffic managers to keep open. Modeling these could lead to a more complete understanding of how much weather-related delay is avoidable. The next chapter takes the ideas and successes of the models presented, and develops realistic 3D route recommendations given the forecast outlook and demand for airspace.

Chapter 5

3D routing in terminal airspace

Previous chapters modeled aircraft trajectories in two dimensions and made several assumptions about the structure of routes, including the assumptions that routes are straight and angled incident to the airport.

In this chapter we model more realistic aircraft, airport, and traffic control constraints and develop algorithms for identifying three dimensional (3D) routes through terminal airspace. The modeling in this chapter is a culmination of previous work, and captures (directional) airspace demand, aircraft dynamics (including turn angle, climb and descent rates, route smoothness, and route length), 3D deconfliction of arrival and departure routes, avoidance of obstacles in 3D (including other airports in a metroplex), and weather forecasts.

We begin by considering the problem of designing airspace routes with the listed constraints, but without accounting for weather. In other words, we consider the problem of designing terminal airspace from scratch. We develop a facility location-based integer programming formulation for the problem of selecting distance-optimal airspace fixes. These fixes are aligned with the traffic flowing in and out of the airport. We then describe a modified A* algorithm which identifies 3D routes between fix-runway pairs subject to the constraints listed. We discuss experimental results for the Chicago airspace.

In the second half of the chapter, we move to the context of tactical planning up to 90 minutes ahead of inclement weather and consider the problem of designing

3D conflict-free routes that are likely to be robust to inaccuracies in the weather forecast, taking into account the same set of constraints. We build upon previous chapters by incorporating the route robustness model with the A* algorithm to yield more realistic route recommendations. The result is a proof-of-concept illustration of how robust 3D routes through convective weather may be determined.

5.1 Constraints for realistic terminal airspace trajectories

In this section we list the high-level characteristics and constraints that describe standard routes through terminal airspace. These constraints fall into two categories: *aircraft constraints* which are specific to individual aircraft, and the *external constraints* necessarily for safe and environmentally-friendly air traffic operations.

The aircraft-specific constraints are:

1. 3D routes must take into account the fleet mix and its corresponding range of ascent and descent rates.
2. Runway choice for an aircraft depends on landing equipment, aircraft size, and weight; not all aircraft can be assigned to a given runway.
3. Routes must be fairly “smooth” and have few turns.

The external constraints include:

4. Vertical and horizontal flow separation.
5. Noise restrictions in the form of vertical minima over cities or neighborhoods.
6. Runway capacities.
7. Existence of a downwind leg buffer for merging.

Note that several of the above constraints are actually more like objectives rather than true constraints. For example, a route with one turn has better ride quality than a route with two, but the number of turns is not a physical limitation of the aircraft. Several other constraints are unspecific only because the details can vary. For example, in today’s airspace aircraft are separated vertically by 1000 feet, and horizontally by 2.5 nmi in the TRACON. However, this depends on radar accuracy,

and the requirements are likely to change with the future adoption of RNAV and other technologies. Specifics will be detailed later in the chapter when algorithm parameters are set.

Now that we have described the high-level constraints of realistic airspace design, we introduce the problem of designing terminal airspace structure and routes subject to these requirements.

5.2 Blank slate design of terminal airspace

Currently, STAR and SID routes are designed manually based on the airport layout, existing Navaid infrastructure, nearby constraints (including cities and other airports) and expected traffic patterns. This is a continuing process that is adjusted over time in a gradual, ad hoc manner. In this section we consider the problem of automatically designing conflict-free terminal airspace routes from scratch.

This direction of research could support the development of new or dynamic 3D airspace routes, which is a goal of research for the NextGen and SESAR programs for modernizing airspace. In particular, it fits within the NextGen concepts of Trajectory-Based Operations (TBO) and dynamic resource and airspace management (Joint Planning and Development Office, 2007). TBO would change the current management of high-density airspace into a system in which exact knowledge of aircraft positions could increase opportunities for controllers to improve throughput (by decreasing separation, adding more routes to airspace, etc.). The concept of dynamic resource management would allow for entire airspace flows to change dynamically in the face of changing demand and weather conditions. This would be made possible by improved Flight Management Systems (FMS) which could follow exact 4D routes. Understanding the design of terminal routes and generating those automatically based on demand and conditions could be a step towards achieving these goals.

5.2.1 Previous Research

The problem of designing conflict-free trajectories for a set of flights has been studied in the research literature. Researchers have focused on the problem of solving

conflicts in air traffic in 4 dimensions (3D + time) and in enroute airspace. The problem has been found to be extremely computationally intensive, and most research uses meta-heuristics to tackle the problem. Approaches have included genetic algorithms (Durand et al., 1996), ant colony optimization (Durand and Alliot, 2009), and more recently an approach based on light propagation (Dougui et al., 2010).

These approaches have modeled some of the constraints listed, including vertical and horizontal flow separation as well as aspects of the physics of flight. However, our focus is flow-based separation for a range of altitude profiles rather than time based separation for individual aircraft. This is more relevant for terminal airspace, which is extremely high density while having relatively simple flows which converge at the airport.

5.2.2 Problem Statement

Given a terminal airspace region, a set of airport locations, and demand for airspace, the goal is to *design a set of 3D routes through airspace (from the TRACON boundary to the runway) which adhere to airspace and controller constraints (1-7) while being optimal* in some sense. In our application, we focus on demand-weighted distance optimality, i.e., that the total distance traveled by all aircraft should be minimized.

5.2.3 Solution Approach

The problem as described does not assume any structure on the *outer fixes*, the points at which aircraft are handed off between the TRACON and center. If we were to assume that these outer fixes are given (say, as the standard STAR and SID fixes), the subsequent problem of designing 3D conflict-free routes subject to constraints (1-7) would be computationally hard. If the state space were discretized and there were no separation or smoothness constraints, it could be modeled as a multicommodity flow, which is NP-hard to solve. However, with the addition of altitude profiles, the problem can no longer be readily modeled as a traditional network flow. This is one reason that meta-heuristics have been used to tackle related problems in previous research.

For these reasons, we separate the problem of selecting optimal outer fixes from the problem of identifying terminal routes through airspace, and solve the two problems sequentially. First, we model the optimal outer fix placement problem and determine placements using an integer programming model. Second, given these fix placements, we identify terminal routes sequentially using airspace demand as input, with a modified A* algorithm. These methods are described in the following two sections, and results for Chicago airspace are discussed later in the chapter.

5.2.4 IP formulation for selection of optimal terminal fixes

Given terminal-area demand in the form of a set of O-D pairs, we model the problem of selecting optimal outer fixes for an airport \mathcal{A} as a facility location problem. Specifically, we open two types of facilities: arrival fixes and departure fixes, and assign demand to each facility so as to minimize the demand-weighted distance traveled. The model aims to align the outer fix positions with major flow directions, under the assumption that the great-circle distance route can be flown from origin to destination.

The linear integer programming formulation, illustrated in Figure 5-1, is as follows. Consider a circle C_O which defines the outer boundary of the TRACON for airport \mathcal{A} . Partition the terminal boundary into n segments, each of which is a potential fix location with an associated nonnegative demand (for both arrivals and departures). For each flight between airports \mathcal{A} and \mathcal{B} , there is a unit demand point at the node closest to the intersection of C_O with the great-circle $\mathcal{A} - \mathcal{B}$ route. The problem is to *select a subset of m fixes of each flight direction so as to minimize the total distance of deviations flights make from their great-circle route when flying into or out of airport \mathcal{A} .* To ensure separation of flows we add a constraint on the minimum distance L between each pair of arrival and departure fixes.

More formally, let $N = \{1, 2, \dots, n\}$ be a set of nodes. Demand from some direction $d \in D$ originates from each node $j \in N$, and is denoted by b_j^d , where $D := \{\text{arr}, \text{dep}\}$ (in this application there are two direction types: arrival and departures). There is a set of potential fixes F , and each $i \in F$ can be “opened” as either an

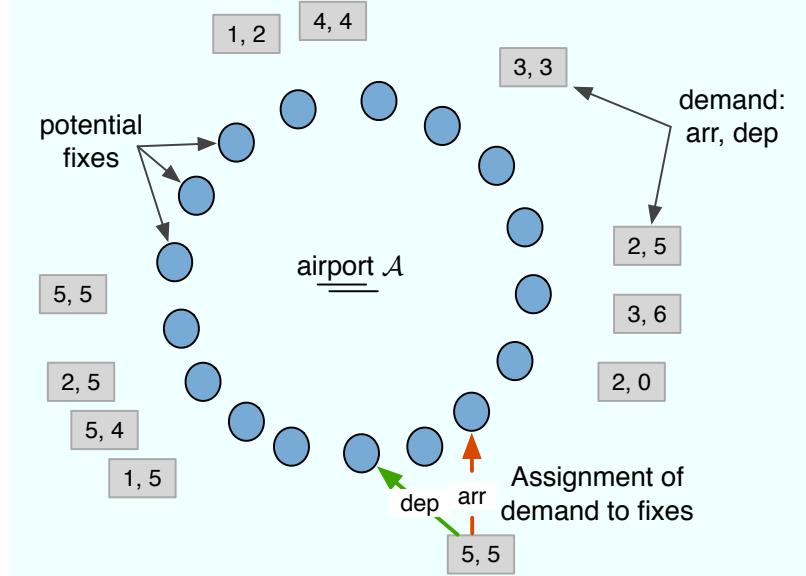


Figure 5-1: Diagram of demand-based optimal outer fix selection model

arrival or departure fix.

Let the cost of assigning a potential demand point b_j^d to fix $i \in F$ be c_{ij}^d . This cost is equal to the demand b_j^d times the distance between $j \in N$ and potential fix $i \in F$, and is denoted by s_{ij} .

The variables are:

$$y_i^d = 1 \text{ iff fix } i \in F \text{ is open for direction } d \in D$$

$$x_{ij}^d = 1 \text{ iff demand } j \in N \text{ of direction } d \in D \text{ is assigned to fix } i \in F$$

The formulation is:

$$\min \sum_{d \in D, i \in F, j \in N} c_{ij}^d x_{ij}^d \quad (5.1)$$

$$\text{s.t. } \sum_{i \in F} x_{ij}^d = 1 \quad \forall j \in N, \forall d \in D \quad (5.2)$$

$$\sum_{i \in F} y_i^d \leq m \quad \forall d \in D \quad (5.3)$$

$$\sum_{j \in N} x_{ij}^d \leq M y_i^d \quad \forall i \in F, \forall d \in D \quad (5.4)$$

$$M(2 - y_{i_1}^{arr} - y_{i_2}^{dep}) + s_{i_1 i_2}(y_{i_1}^{arr} + y_{i_2}^{dep}) \geq 2L \quad \forall i_1, i_2 \in F \quad (5.5)$$

The objective function Equation 5.1 is simply the total cost (distance flown) by aircraft. Note that other objectives could be incorporated into this cost function, including wind-optimal routes which reflect average wind conditions, and distance-optimal routes between standard jet routes rather than origin airport.

Equation 5.2 assigns each demand point to exactly one fix of the appropriate direction. Equation 5.3 bounds the number of fixes selected for each direction by m . In a more general setting, m can be replaced by a separate m_d for each direction $d \in D$. Equation 5.4 ensures that each demand point of a given direction is assigned to an open fix of the same direction; if any demand of direction d is assigned to fix i , then fix i must be open. Finally, Equation 5.5 separates the two fix types by at least L km, and in particular ensures that arrivals and departures are not assigned to the same fix. M is a large constant.

Once we have selected a set of optimal outer fixes, we use them as input to the conflict-free trajectory generation algorithm, described next.

5.2.5 A^*-1 : algorithm for 3D conflict-free route identification

This section describes the standard A^* algorithm and introduces modifications necessary for our application.

Given a graph $G = (N, A)$ with nodes N , arcs A , and arc costs $c_a \geq 0 \ \forall a \in A$, Dijkstra's shortest path algorithm finds the shortest-length path from a given source $s \in A$ to a given sink $t \in A$, in time $O(|N|^2)$ (Ahuja et al., 1993). Nodes are expanded one by one, where the node $x \in N$ with the lowest cost $d(s, x)$ from the source is selected at the start of each iteration, out of all unexamined nodes.

The A^* algorithm builds on this base algorithm and introduces a cost-to-go heuristic that tends to decrease the number of nodes that are examined in practice, although the theoretical run time does not change (Hart et al., 1968; Dechter and Pearl, 1985). This cost-to-go $h(x)$ from node $x \in N$ to the sink t must be a lower bound on the cost of the actual shortest $x - t$ path. Nodes are expanded one at a time as in Dijkstra's algorithm, but the node chosen at each iteration is the node x with minimum $f(x) := d(s, x) + h(x)$, the lowest combined distance from the source and estimated

cost-to-go. This has been proven to give an admissible $s - x$ path at each iteration, yielding a shortest $s - t$ path upon algorithm completion. If the cost-to-go is chosen well, A* tends to be very fast in practice. Algorithm 1 has pseudocode for the basic A* algorithm. Subprocedures are described with more detailed pseudocode in Appendix B.2.

Algorithm 1 Find 3D Route

```

procedure A*( $s, t$ )            $\triangleright$  source and sink node, plus network parameters
    CLOSED  $\leftarrow \emptyset$ 
    OPEN  $\leftarrow \{s\}$ 
     $f[s] \leftarrow \text{heuristicDistanceToSink}(s)$ 
    while OPEN  $\neq \emptyset$  do
         $x \leftarrow \text{argmin}_{y \in \text{OPEN}} f[y]$ 
        OPEN  $\leftarrow \text{OPEN} \setminus \{x\}$ 
        CLOSED  $\leftarrow \text{CLOSED} \cup \{x\}$ 
        if  $x == t$  then
            return RECONSTRUCT_PATH( $x$ )
        end if
        for all  $y \in \text{CHILDREN}(x)$  do
            if  $y \in \text{CLOSED}$  then
                continue
            end if
            if  $y \notin \text{OPEN}$  then
                 $h[y] \leftarrow \text{heuristicDistanceToSink}(y)$ 
                 $f[y] \leftarrow \infty$ 
            end if
             $f[y] \leftarrow \min(f[y], g[x] + c(x, y) + h[y])$ 
        end for
    end while
    return INFEASIBLE
end procedure

```

We adopt the basic A* algorithm to sequentially identify 3D terminal routes. The node state space has 5 dimensions: x, y, z , the heading θ , and the total heading γ . There is a source node s with some initial heading, and a sink node t . The altitude z represents an altitude range centered at z . This altitude range is governed by the altitude profile for the particular route, which is a nondecreasing function $A(\cdot)$ of the shortest path from the source to each node.

We first defined these terms. For each node $u = (x_u, y_u, z_u, \theta_u, \gamma_u)$, there exist k

arcs (u, v) such that $v = (x_v, y_v, z_v, \theta_v, \gamma_v)$, $\|(x_u, y_v) - (x_v, y_v)\| = 1$ km, the altitude is $z_v = A(z_u + 1)$ and $\theta_v \in \{\theta_u - 2\theta_\Delta, \theta_u - \theta_\Delta, \theta_u, \theta_u + \theta_\Delta, \theta_u + 2\theta_\Delta\}$. The total heading γ_v reflects any heading changes: $\gamma_v = \gamma_u + |\theta_v - \theta_u|$. Parameter θ_Δ is the incremental change in heading possible between nodes. In other words, nodes are expanded in k directions adjacent to u , and the altitude is adjusted to reflect the altitude profile.

The set of nodes as described is very large: if the state space were discretized for a 200 x 200 km terminal airspace, with the first three parameters rounded to the hundredths digit, and with 72 possible values for heading and total heading (for $\theta_\Delta = 5^\circ$), there would be more than 2^{14} nodes and k times as many arcs, making it intractable to enumerate the entire graph.

The A* algorithm is well-suited for this setting because nodes are expanded as needed at each iteration. Moreover, the Euclidean distance can be used very effectively as cost-to-go $h(x)$ (and is used in subprocedure heuristicDistanceToSink in the pseudocode) because it always provides a lower bound on the shortest path distance.

We make several adjustments to A* to incorporate constraints (1-7). The constraint on heading changes while expanding nodes ensures that turns are smooth and do not contain large angles. The shortest path requirement results in relatively short routes. To ensure that these routes do not spiral upwards in order to avoid obstacles, nodes are deleted when their total heading γ exceeds some constant H (we use $H = 360^\circ$). The downwind leg buffer, vertical minima for noise restrictions, and separation of flows are enforced as obstacles such that no node can be expanded to intersect with an obstacle. A data structure of obstacles is checked when neighbor nodes are expanded.

This enhanced algorithm will be referred to as A^*-1 from now on.

5.3 Results for blank slate terminal design

We test our two-phase approach to designing terminal airspace routes using Chicago airspace as a case study. This terminal was selected because it is a metroplex, it sees a significant amount of NAS traffic, and it is heavily affected by convective weather. Its two major airports Chicago O'Hare International (ORD) and Chicago Midway

(MDW) are positioned only 25 km apart. ORD is the 2nd busiest airport in the United States with an average 2400 operations per day in 2010, while MDW averaged roughly 30% of that, with 670 flights per day (OPSNET, 2011).

This section begins by describing Chicago airspace and the airport configurations, altitude profiles, and demand used in our study. We then use this input to evaluate our route design algorithms.

5.3.1 Chicago airspace configuration

We focus on a single airport configuration at ORD: 22R, 28 | 22L, 2L, 32R (arrivals on runways 22R and 28, and departures on runways 22L, 32L, and 32R). According to the FAA's Aviation System Performance Metrics (ASPM) Data, this was the 4th most common configuration at ORD in 2008, and the 2nd most common eastern arrival configuration during the same time period (FAA, 2010). At MDW, we focus on the 4R | 4R configuration, the 2nd most common configuration during 2008.

This pair of configurations was in operation on June 4 2008, between 15:00 and 19:00 local time (CST). Figure 5-2 shows a visualization of traffic for this time period¹ with the main STAR and SID routes at ORD overlayed. The bottom of Figure 5-2 shows close-ups of traffic at ORD and MDW to highlight this particular configuration.

5.3.2 Demand data

Demand is a key consideration when determining optimal routes through airspace, as the placement of terminal fixes should line up with major traffic flows. In this section we use ETMS data to generate demand scenarios for our problem at each of the main Chicago airports.

We compute two empirical demand data sets for use in this chapter:

Demand1: **Demand for a single runway configuration.** For each O-D pair serviced by the airport between 15:00 and 19:00 CST on June 4 2008, we compute arrival and departure demand counts separately. Each OD-pair is then grouped by azimuth with respect to ORD, giving a count of arrival

¹Traffic data is from the Enhanced Traffic Management System (ETMS)

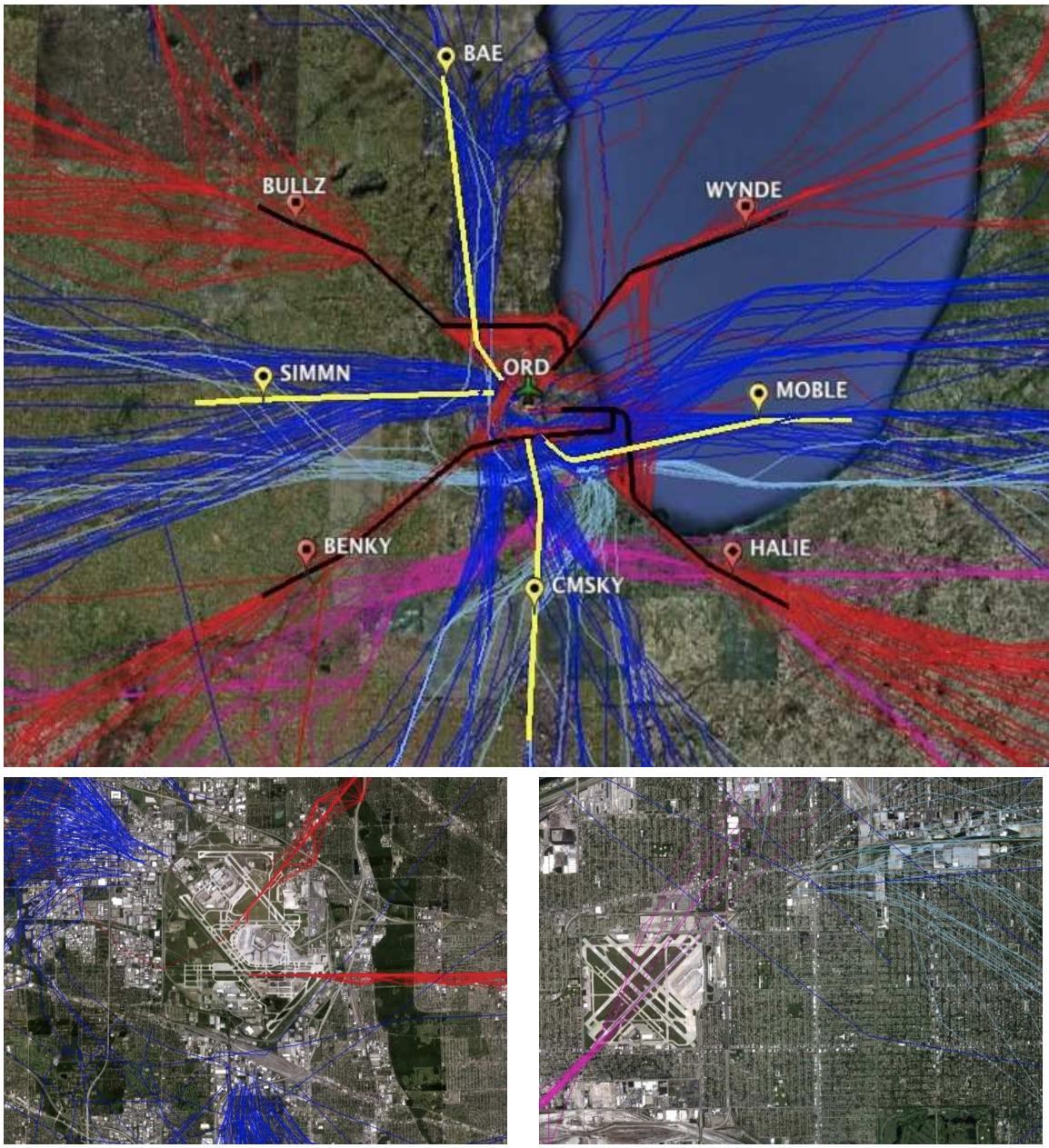


Figure 5-2: Air traffic over Chicago airspace on June 4 2008 (top), with close-ups of ORD (bottom left) and MDW (bottom right). © Google 2010, Image U.S. Geological Survey, USDA Farm Service Agency.

and departure demand at each azimuth.

Demand2: Demand for an extended time interval that reflects typical flow patterns at an airport. The dataset is computed identically as above, but for operations on the entire day of June 4 2008.

The short-term demand scenario Demand1 will be relevant later in the chapter for

making tactical changes to arrival and departure routes during convective weather. Each O-D demand point will be assigned to the nearest STAR or SID route in the current airspace structure, to determine true demand, and to determine priorities when making decisions about dynamic route movements. The reason we do not simply check empirical counts along standard routes is that these counts may have been subject to load balancing or other air traffic control procedures, artificially increasing demand along certain standard routes.

The long-term average demand at an airport, Demand2, is relevant to the blank slate design of terminal airspace, and will be used for the results later in this section.

5.3.3 Altitude profiles

The final input data needed for designing 3D routes through airspace is a description of altitude profiles for ascending and descending aircraft. We estimate ascent and descent profiles empirically using one day of operations for ORD.

When the total distance to ORD is plotted against flight level (FL) for all flights, we find that the altitude profiles across all flights are roughly cone-shaped with linear boundaries. Using this observation, we estimate the ascent and descent profiles by fitting a cone to the data, a cone which contains $\alpha\%$ of tracks with $\alpha \in [0, 100]$. Ideally α is close to 100%.

Formally, we're given aircraft positions and altitudes for a given time period as a set of points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where each (x_i, y_i) is a (flight distance from ORD, flight level) pair. We would like a set of two lines $x = a_1x + b$ and $x = a_2x + b$, with the same y-intercept b (so that they form a cone), which contains $\alpha\%$ of the traffic points. These lines are estimated from points (x_i, y_i) which satisfy $x_i \leq D_{\max}$ km, where D_{\max} is a cutoff on distance from ORD used to estimate the altitude profile.

We estimate \hat{b} , \hat{a}_1 , and \hat{a}_2 using the following procedure:

1. Take all data points (x, y) with $x \leq D_{\max}$. (For arrivals we use $D_{\max} = 30$ km, for departures $D_{\max} = 20$ km.)
2. Let \hat{b} be the y-intercept of the linear regression of these points.

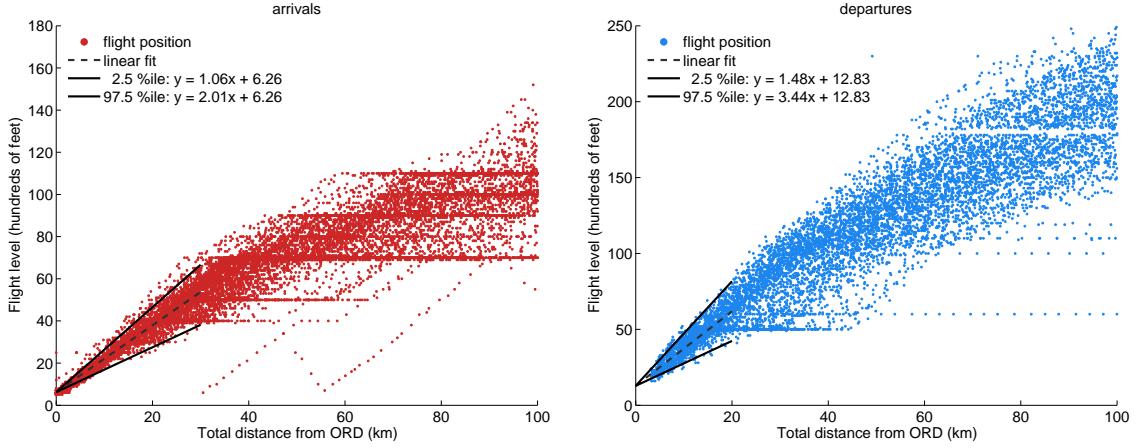


Figure 5-3: Altitude profiles for arrivals (left) and departures (right), estimated empirically using flight tracks for ORD.

3. Let $z := (y - \hat{b})/x$. This gives the slope for each pair of points.
4. Sort z , and set \hat{a}_1 and \hat{a}_2 to the $\frac{\alpha}{2}$ - and $(1 - \frac{\alpha}{2})$ -percentiles of z , respectively.

Figure 5-3 shows the resulting $\alpha = 95\%$ altitude profiles for arrivals and departures, corresponding to ORD traffic on June 4 2008. The arrivals profile on the left end shows a linear relationship between distance to ORD and flight level, though there are a range of altitudes corresponding to different aircraft types and operational practices. In particular, the horizontal signals in the figure indicate that the altitude profiles are influenced by STAR routes rather than purely aircraft abilities. However, for distances under 30 km, there is a clear conic shape, and we use this (as L) to estimate descent profiles. After 60 km from the airport, altitudes are relatively flat. Therefore in our model, we extend the cone to 60 km, and then assume altitude flattens out between 60 and 100 km. For departures (right), the climb is steady, and we extend the cone through to 100 km, with a maximum flight level of 250.

Table 5.1 gives altitude profiles for varying values of α for arrivals and departures. Naturally, the cone becomes tighter around the linear fit at lower values of α .

There are two potential shortcomings to the altitude profiles as computed. In real-world operations, each demand point (for example, an outer departure fix) may serve a different fleet mix. This can be accounted for by computing separate altitude profiles for each such demand point to reflect these differences. Second, by using empirical

α	arrivals	departures
80	$y = 1.28x + 6.26$	$y = 1.85x + 12.83$
	$y = 1.78x + 6.26$	$y = 3.09x + 12.83$
90	$y = 1.18x + 6.26$	$y = 1.65x + 12.83$
	$y = 1.89x + 6.26$	$y = 3.28x + 12.83$
95	$y = 1.06x + 6.26$	$y = 1.48x + 12.83$
	$y = 2.01x + 6.26$	$y = 3.44x + 12.83$

Table 5.1: Estimated altitude profiles for arrivals and departures for varying confidence intervals

tracks to estimate the altitude cones, the resulting altitude profiles are a function of existing STAR and SID routes and their corresponding altitude constraints. It would also be possible to estimate theoretical altitude profiles using the Base of Aircraft Data (BADA), which could result in altitude profiles which reflect optimal performance within a fleet mix.

5.3.4 Results: optimal terminal fixes

The integer program for identifying optimal outer fixes (described in Section 5.2.4) was run using the Demand2 dataset for ORD. This section describes the computational results.

The model parameters were set to $m = 4$, so that 4 arrival and 4 departures routes were selected (for ease of comparing against a four corner post configuration)², and $L = 20$, thus limiting the required separation between arrival and departure fixes to 20 km.

The resulting IP formulation had over 250,000 variables and constraints. The computational effort to solve the problem was significant; CPLEX solved the instance to optimality in just under 4 hours on a 2.66 GHz processor with 4GB of RAM. An integral solution within 5% of optimum was found in under 1 minute. We did not attempt to improve this solution time because our particular application does

²This value of m is reasonable for arrivals, as airports typically have 4 arrival fixes arranged at the four corner posts of the airspace. However, departures are different. Airports typically have 4 departure gates, but each gate contains 2-4 fixes. Moreover, departures tend to fan out and not pass directly over the fixes filed on their flight plan. For simplicity we set m to 4 for departures in this case study.

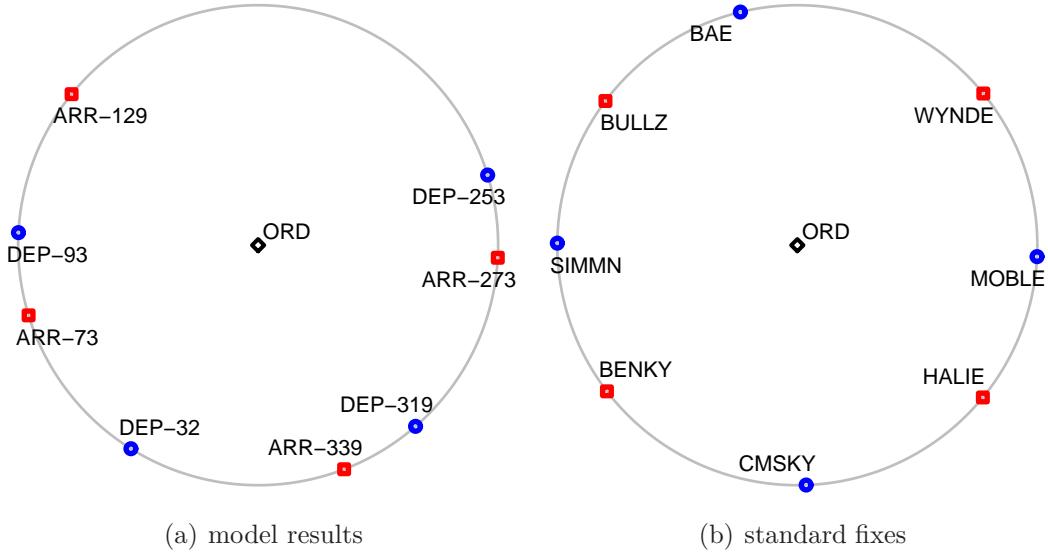


Figure 5-4: Comparison of outer fixes selected by IP approach (left) with the STAR and SID fixes at ORD (right).

not require real-time solutions, but this would be an interesting problem for future research.

Figure 5-4(a) shows optimal fix positions (100 km away from ORD), with arrival fixes indicated with red squares, and departure fixes indicated with blue circles. Note that fixes alternate between arrival and departures, just like in the standard corner post configuration of ORD depicted in Figure 5-4(b), though they are positioned more heavily to the south rather than spread out evenly around C_O , in order to be aligned with demand (there is not as much demand coming into ORD from the north).

To evaluate the quality of this solution, we compare it against the standard configuration as follows. For the Demand2 dataset, we compute the great-circle distances flown for both the optimal and the standard fixes, such that each demand point is assigned to fly via its closest fix (of the appropriate direction). The standard fixes used for ORD are the arrival fixes BENKY, BULLZ, WYNDE, and HALIE, and the departure fixes SIMMN, CMSKY, MOBLE, and BAE. We note that this comparison is not entirely fair to departures because we only use four departure fixes rather than the full set at ORD.

For arrivals, the savings of optimal over standard fixes is 4,328 km, equivalent

to 0.26% of total distance traveled across 1,248 flights. For departures the savings was much lower at 768 km, or 0.05% of total distance across 1,267 flights. The low percentages are not surprising and reflect the relatively long O-D routes, whose distance is not significantly affected by the displacement of their target along a 100 km circle outside of Chicago. However, the total distance saved can become worthwhile when savings accumulate over entire years and multiple airports.

5.3.5 Results: conflict-free 3D routes

This section presents the results of running the A^*-1 algorithm on Chicago airspace.

We first describe the input and parameter settings. Let F^* be the set of optimal outer fixes identified by the IP in the previous section, and let F_{STD} be the set of standard outer fixes for the configuration in dataset Demand2. The outer fixes F^* are the sinks for our algorithm. Each sink is paired with a source corresponding to a runway, which is selected as follows. First, each fix $f \in F^*$ is paired with the closest standard fix $f' \in F_{\text{STD}}$ (this is done manually by looking at Figure 5-4). Among all runways in the Demand2 configuration at ORD, select the runway $r_{f'}^*$ which sees the greatest empirical demand fed by f' . For example, if f' is an arrival fix, $r_{f'}^*$ is the most common runway used by the STAR corresponding to f' . The source s is then defined to be a final approach fix 10 km away from the runway threshold for arrivals, and the point at which we start to see departure tracks for departures, about 1.5 km from the departure runway threshold. With this choice of source points, we can align the altitude profiles with each s . The heading γ_s for the source corresponds to the heading of empirical $f' - s$ tracks when they cross s . Let S denote the resulting set of sources (and note that different fixes can be routed to the same source). The demand associated with each $s - f$ pair is the Demand2 demand for f , denoted d_f .

We now have eight source-sink pairings $s - f$ (corresponding to inner- and outer-fixes) for which to identify routes.

The ascent and descent profiles used were those developed in Section 5.3.3 with $\alpha = 80\%$. To ensure smooth routes, nodes are expanded by 1 km in each of $k := 5$ directions, in increments of $\theta_\Delta := 5^\circ$. Thus the maximum turn angle is 15° and all arc

costs are 1. The maximum amount of turning along a route is $H := 360^\circ$. The vertical and horizontal separation assumed are 1000 ft and 2 km, respectively. For simplicity in this experiment, we do not introduce obstacles to ensure room for a downwind leg and merging operations for arrivals, nor do we incorporate vertical minima over cities or neighboring airports. The cost-to-go $h(u)$ used is the Euclidean distance between node u and sink f plus an additional distance to account for any heading change that must take place to travel along the $u - f$ line segment.

For each fix-runway pair, we identify a route through terminal airspace sequentially, using the following procedure:

1. Sort the $s - f$ pairs by decreasing demand d_f , with arrivals first, followed by departures
2. Let $\text{Obstacles}(s) = \emptyset \quad \forall s \in S$
3. For each $s - f$ pair (in order):
 4. Run A^*-1 to identify the shortest-path route r_{s-f}^* , avoiding $\text{Obstacles}(s)$
 5. Update $\text{Obstacles}(s') \quad \forall s' \in S, s' \neq s$ to reflect horizontal and vertical separation around r_{s-f}^*

Note that once a route is identified, it is added to the list of obstacles for all other routes, except for those routes which share a source (and therefore target the same runway in the same direction). This is to accommodate necessary merges of routes that share a runway resource, and tends to produce natural merge and split points.

Figure 5-5 shows results of the algorithm. The solution time was on the order of seconds for 7 of the routes, and just under 4 hours for the final route added, F238-DEP. The computational burden grew substantially in the case of F238-DEP because much of the state space was examined due to obstacles blocking the (obstacle-free) shortest path. Although real time route solutions are not necessary for this application, they will be important in the next section when weather is integrated into the algorithm so as to improve tactical decision making during thunderstorms. We will discuss further heuristic improvements to the algorithm runtime there³.

³The statespace pruning step will be shown to improve runtime substantially. For comparison, the algorithm identifies an optimal route for F238-DEP in under 3 minutes.

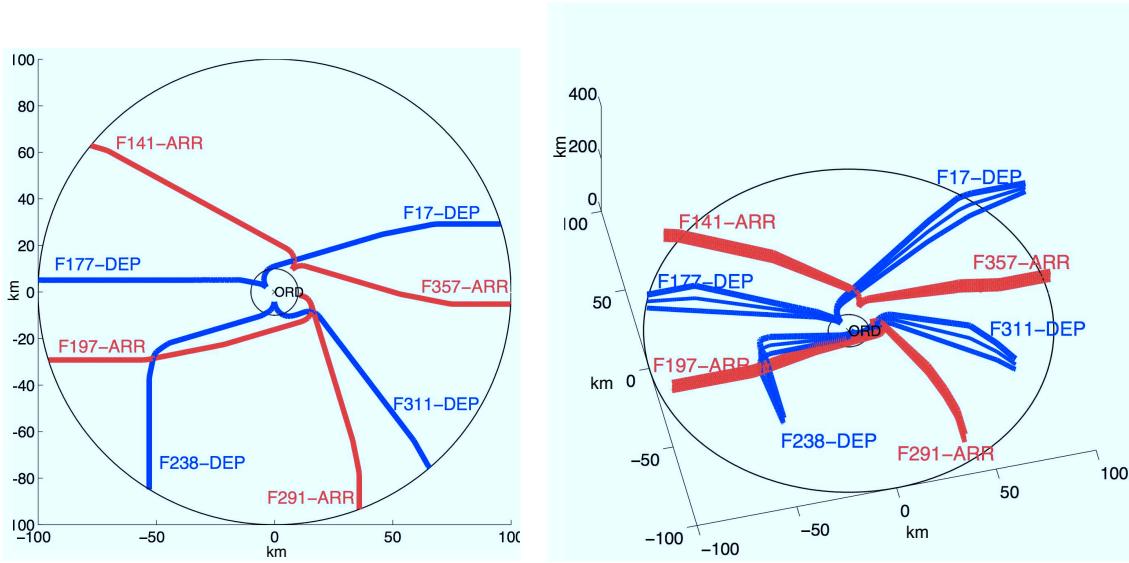


Figure 5-5: 3D conflict-free terminal routes, at different angles

Overall the terminal routes identified are smooth and adhere to the stated constraints. Despite the sequential (and hence not necessarily globally optimal) approach, we have made a first step into the automated design of realistic 3D terminal airspace trajectories. Simulations and human factors analysis would be necessary to fully evaluate these routes against existing STARs and SIDs.

5.4 3D routes through convective weather

This section incorporates the algorithm for realistic 3D routes through airspace into an approach for decision support during convective weather events.

The main idea is as follows. The STAR and SID routes are evaluated using the route robustness model t_0 minutes ahead of operations. New 3D conflict-free routes are identified to replace those predicted to be blocked. Note that the weather forecast used is still 2D, but any route recommendations based on the forecasts are 3D, with guaranteed horizontal and vertical separation of flows. Since the forecast of VIL represents vertically integrated liquid, and since flights are climbing or descending in the terminal-area, it is reasonable to use this 2D forecast for 3D planning.

5.4.1 Algorithm

Let R_{STD} be the set of standard routes, where each $r \in R_{\text{STD}}$ specifies a route between source node s_r and sink node t_r . The algorithm is the following.

1. For each $r \in R_{\text{STD}}$, obtain the probability \hat{p}_r that r will be blocked in the true weather, given the forecast available t_0 minutes in advance of operations.
2. For all routes r with $\hat{p}_r > 0.5$: fix r in place as an obstacle.
3. For all routes r with $\hat{p}_r \leq 0.5$, ordered by demand, arrivals first:
 4. Run A^*-2 to obtain a new route r' with sink t'_r within L km of t_r along C_O
 5. Fix r' as an obstacle

Note that the procedure A^*-2 in step 4 refers to a further modified version of the A^* algorithm, described in the next section. The key modifications include improved runtime and the incorporation of weather.

When a route r is fixed as an obstacle in Steps 2 and 5, all nodes within the horizontal and vertical separation distance of r are removed from the network for routes r' that do not share a source t_r with r .

5.4.2 A^*-2 : Further modifications to the A^* algorithm

This section describes A^*-2 , a version of the A^* algorithm for identifying realistic 3D routes through airspace introduced in Section 5.3.5, with the several modifications.

The first modification is a change to the cost function of each arc, in order to incorporate weather into the cost function. When a new route is identified, we would like it to be robust to the error in the weather forecast. However, although the route robustness model can evaluate each of a set of routes, it is not designed to identify an optimal feasible solution (unless all feasible routes are evaluated). Instead of using the predictor directly, we take knowledge gained during the development of the route robustness model to develop a cost function that penalizes nodes likely to have hazardous weather. We create a heuristic cost function at each node which combines the features which have been shown to be good predictors of blockage.

More precisely, the previous cost function for arc $(u, v) \in A$ was the Euclidean

distance between u and v , $c(u, v) = ||u - v||$. We define the new cost function to be $c'(u, v) = ||u - v|| + f_{\text{forecast}}(v)$, a function of the forecast grid overlayed onto our network. For simplicity, we use a linear combination of three simple features: the maximum weather level in the B km neighborhood of v , the weather level at v , and the distance to L3+ weather from v . Let $f_1(v), f_2(v), f_3(v)$ refer to these respective values for node v . Then we use $f_{\text{forecast}}(v) = \max(0, f_1(v) - 2) + M \max(0, f_2(v) - 2) + 0.1 \max(0, 10 - f_3(v))$.

The second modification is the addition of a supersink so that nodes along C_O and within L km of the original sink r_t are all feasible sinks. This way the original route can move to avoid weather hazards. This change to the network merely requires replacing the check $x == t$ in Algorithm 1, line 9, (which checks if the current node x is equal to the sink node t) with the subprocedure IS_SINK that checks whether the described condition is satisfied.

The final modification is a speed improvement. We improve the performance of the algorithm significantly by pruning the state space regularly. After every 10,000 nodes from the OpenSet have been examined, procedure PRUNE is run, which removes from OpenSet any nodes that are dominated by close neighbors. This essentially decreases the size of the statespace while avoiding larger rounding errors (which would occur if the statespace were less granular). The procedure PRUNE is described in greater detail with pseudocode in Appendix B.

5.4.3 Experimental Results

This section presents results for the tactical planning of 3D routes through weather at a 60 minute planning horizon.

We use as input the Chicago airspace structure shown in Figure 5-2, along with the Demand1 demand scenario. For simplicity, we apply the weather scenarios from ATL to this traffic, so that the route robustness model does not have to be retrained on Chicago weather patterns.

Initial experimental results are shown in Figure 5-6, with the weather forecast overlayed with terminal routes for three separate scenarios. Unchanged standard

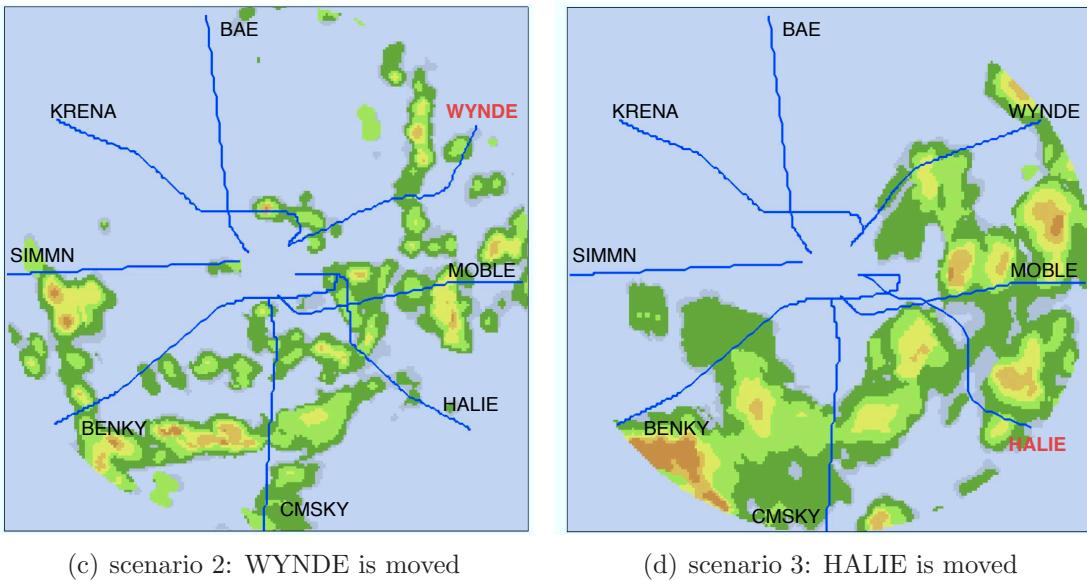
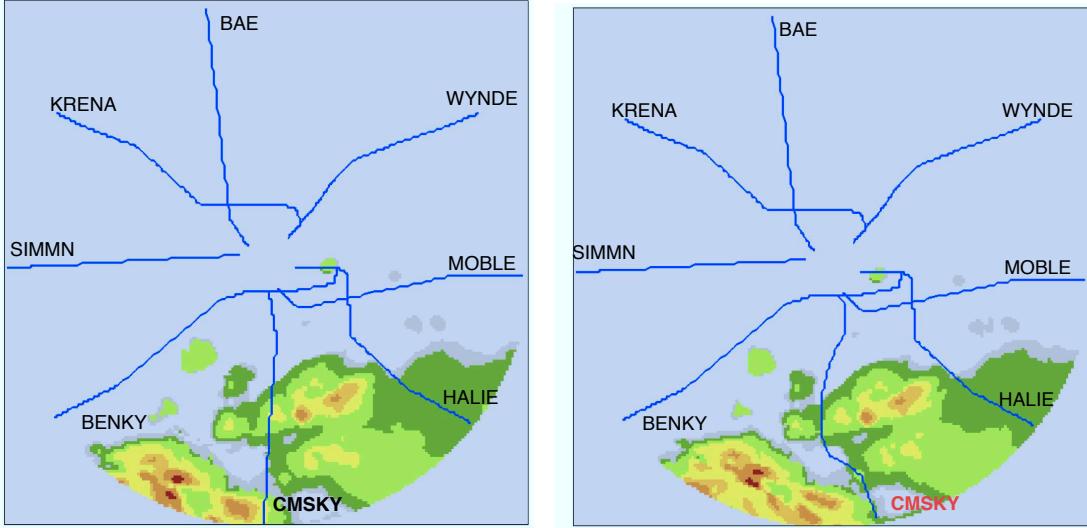


Figure 5-6: Experimental results for three weather scenarios. Red routes are conflict free routes identified by the A^*-2 algorithm.

routes are indicated with black text, and altered routes are indicated with red bold text. In Figure 5-6(a) the forecast and standard routes are shown for scenario 1. Convective weather is forecast to be present only in the southern portion of the terminal, and CMSKY is the only route predicted to be blocked by the route robustness model. Figure 5-6(b) shows the results of A^*-2 , which identifies a new route that maneuvers around forecast weather cells, and replaces the standard CMSKY STAR.

Figures 5-6(c) 5-6(d) show algorithm results for two more scenarios, which move the standard WYNDE and HALIE routes, respectively.

5.4.4 Caveats and future directions

Our approach to 3D terminal airspace planning is a proof-of-concept. Further enhancements and validations are required to fully evaluate the approach. In addition, there are several caveats that should be addressed.

The first caveat is that the standard routes evaluated with the route robustness model contain turns, including the downwind leg for arrivals. The blockage predictor, however, was trained solely on straight routes (line segments), and should be re-trained on routes with turns. In previous work we found little change in prediction performance when turns were included in the feature set (Michalek and Balakrishnan, 2009a), so we have reason to believe the predictions are still reasonable.

A related issue is the assumed aircraft speed profiles, which are unchanged from previous chapters, and assume a straight route flown at a constant speed. However, the altitude profiles used correspond to empirical aircraft speed, which changes with distance to the airport (arrivals slow down and departures speed up). Since the CIWS grid has very little movement in weather cells between 5-minute increments, this is not expected to affect the results.

The second caveat is related to merge operations very close to the airport. The downwind leg of arrivals should largely be static in the A^*-2 algorithm. In addition, when multiple routes with the same source split off (or merge), they should not cross again. This is violated in Figure 5-6(d) by the BENKY and HALIE routes, which share a source, then separate near the airport (HALIE skips the downwind leg), and cross again. Enhanced handling of merges should be included in future versions of the algorithm.

Finally, the algorithm as described returns a route of minimum cost, but does not then evaluate the route's likelihood of being open. This validation should be performed.

5.4.5 Conclusion

Realistic terminal airspace planning in the face of weather can be of enormous use to air traffic control; it can improve tactical decision-making and thereby reduce delay during weather events. The ad hoc decision-making present in many airport terminals during convective weather events can be improved with robust predictions of blockage and 3D conflict-free route recommendations that traffic controllers can trust. While work is needed to train and equip ATC and pilots with the ability to implement dynamic airspace changes, we believe that it can decrease delay *and* decrease airspace complexity during weather events.

Chapter 6

Conclusions

The focus of this thesis has been on the integration of state-of-the-art aviation weather forecasts into air traffic management decision-making. We conclude with a summary of the thesis and its major contributions, and a discussion of potential directions for future research.

6.1 Review of Thesis

The thesis began in Chapter 1 with a review of research literature related to air traffic flow management, dynamic airspace configuration, convective weather forecasts, and airspace capacity forecasts. We saw that while there existed much research in each of the areas, there was a gap between the assumptions made in ATFM algorithms and the form and reliability of existing aviation weather products.

Chapter 2 introduced the state-of-the-art CIWS convective weather forecast developed by MIT Lincoln Laboratory. As with other aviation weather forecasts, there were significant spatial and temporal errors when comparing the forecast VIL at each individual pixel against the VIL that actually materialized. We argued that while this may be the case, the CIWS forecast could still be useful from a traffic flow management perspective, as local errors are not necessarily critical for planning.

Motivated by this argument, we introduced a route-based approach to viewing and evaluating weather forecasts, and showed how the resulting forecasts posted much

better skill scores, while at the same time being useful for TFM.

Chapter 3 took the route-based approach one step further and presented an approach for identifying routes that are likely to be robust to the inaccuracies of convective weather forecasts. A set of features of weather forecasts were evaluated for high correlation with route blockage in observed weather. These features were then utilized in classification algorithms based on machine learning techniques to predict, given a set of potential routes and a weather forecast, which routes were likely to be blocked in the observed weather. The performance of the proposed classifiers was evaluated and compared to the raw forecast predictions, using several metrics including the false positive rate and the overall accuracy. It was shown that classifiers can be optimized to minimize the FP rate (important for aviation applications), and the trade-offs between overall accuracy and the FP rate were illustrated.

Chapter 4 applied the route robustness model to the problem of dynamically restructuring terminal airspace. Two models were proposed. The first was an algorithm that simply selected the most robust (high probability) route in each sector, keeping existing sector boundaries fixed. The second was an integer programming formulation for the problem of selecting an optimal route in each sector while gently moving sector boundaries, subject to constraints on the deviation of sector boundaries and routes from the standard positions. Both approaches were evaluated against 28 weather scenarios at ATL from summer 2008 (the route robustness model was trained on a separate 2007 dataset). Both models were shown to increase airspace throughput during convective weather. In particular, the IP model made available 13% more routes that were forecast blocked during these scenarios.

The stability of recommended routes in the face of changing weather conditions was also evaluated by tracking algorithm results for 12 consecutive time periods, each 5 minutes apart. The routes output by the algorithm were found to be highly stable; once a high-probability route was identified, it tended not to move during the subsequent time period, with an average displacement of only 1° . Moreover, across all scenarios evaluated in this way, an optimal route was available 88% of the time.

Chapter 5 relaxed several assumptions made in the terminal airspace model used

earlier in the thesis, with the goal of identifying realistic 3D routes through terminal airspace that could be used to route aircraft during convective events. The chapter began by studying the problem of designing terminal airspace routes from scratch, in the sense that no assumptions were made about today's airspace structure. Given O-D demand for a given period of time, optimal fixes were selected using an integer programming formulation. For each of these fixes and an appropriate runway (dependent on the specific runway configuration in operations), an algorithm based on the A* shortest path algorithm was developed to identify a 3D conflict-free route through the terminal. The algorithm identified routes which adhered to the following pilot, airport, and air traffic controller constraints: horizontal and vertical separation, smooth with a realistic turn radius, included an altitude profile representative of the fleet mix flying the route, and avoided obstacles such as cities and airspace used by nearby airports.

The second part of the chapter incorporated this 3D route identification algorithm to tactical route planning during convective weather, using Chicago airspace as a case study. The main STAR and SID routes at ORD were evaluated against the weather forecast using the route robustness model. For those routes predicted to be blocked, new routes were identified (sequentially in order of demand) using the A* algorithm (which was further modified to penalize for features of the weather forecast).

6.2 Future Directions

This section highlights a few directions for future research.

Incorporation of controller workload

The proposed methods for modifying existing airspace structure have focused on proofs-of-concept and identifying potential benefits of relaxed airspace configuration. Although we have had informal discussions with former air traffic controllers and airspace operators about the assumptions made, we have not performed a study of the implications of this research on controller complexity. Further research into human factors and controller workload involving human-in-the-loop experiments is necessary to study how changes to terminal airspace structure would affect operations and

safety.

Improving model of pilot deviation in terminal airspace

In this research we have defined level 3+ VIL to be hazardous, as pilots tend not to fly through it in enroute airspace. In reality, the situation is more complex, especially in terminal airspace. There exist cases of pilots flying through level 3 weather, and cases of pilots who seem to be avoiding level 2 weather¹. In terminal airspace, multiple factors are believed to affect pilot deviation, including demand, how much delay a flight is already experiencing, and VIL. It would be interesting to incorporate a better model of pilot deviation into this work.

Improving the Tactical 3D planning approach

Chapter 5 introduced an approach for modeling realistic aircraft, airport, and ATC constraints to identify 3D routes. More experiments of this approach are warranted to further prototype the idea for use in decision support. The modeling of merges and downwind leg buffers should be explicitly considered.

Global rather than sequential route identification

The 3D routing approach from Chapter 5 identified conflict-free routes sequentially. Although each subsequent route is the shortest path given constraints set by the routes discovered earlier, this does not necessarily result in a globally optimal solution in which the combined route length is minimized. Although simultaneously solving for all routes is expected to be computationally difficult, it definitely warrants further research.

Better incorporation of demand

Demand is an important factor in making decisions about whether or not to prioritize traffic on a given route. We incorporated demand into our model in Chapter 5 by simply changing the order in which we identified new routes, thereby giving precedence to routes with higher demand. Demand could be incorporated more fully into the model in other ways, such as by setting capacities on individual routes and modeling the spill-over of traffic onto lower-demand routes.

¹This is based on conversations with the Aviation Weather Sensing Group at MIT Lincoln Laboratory

This thesis has shown that it is possible to incorporate weather forecasts with reliable and realistic tactical air traffic management algorithms and decision support tools. The proposed methods were shown to have the potential to increase airspace capacity during convective weather events. With the aircraft equipage and technological advancements of the NextGen air transportation system, the opportunity and potential for traffic management decision support will increase further, and the research directions outlined could significantly reduce this future congestion and delay.

Appendix A

Interviews with Air Traffic Controllers

In the process of conducting research for this thesis, we have had conversations with several air traffic controllers. The informal interviews took place during July 2010, and focused on the existing flexibility of airspace structure and air traffic operations, and on the potential for dynamic airspace concepts during weather events. This appendix contains a record of the conversations. Thank you to Jean Anouilh, Bertrand Foucher, and Leo Prusak for taking their time to answer our questions.¹

ATC during convective weather

During weather events, ATC is tactical. One aircraft requests a heading change, and ATC vectors that aircraft to the airport; same for next aircraft. Once one aircraft asks to deviate, ATC has a better understanding of the weather situation and can tell later aircraft to make the same deviation and go the same way.

If arriving aircraft are moved from their nominal routes or fixes, the departures are usually slowed down by ATC to create a buffer for the increased coordination necessary to maintain separation.

To illustrate the difficulty in forecasting weather accurately and managing traffic during weather events, consider the following example from one morning of operations

¹Disclaimer: this appendix does not contain a transcript of the interviews, but rather an interpretation by the author of this thesis (based on notes) of controllers' opinions.

in New York. There was a large storm 50 miles west of La Guardia Airport (LGA) which was moving east and which would pass just north of the airport. Some arrivals were deviating. However, although the departure SID to the southeast was “open” according to the airport, the departing aircraft were sitting on the ground, unwilling to go.

Flexibility of STAR and SID routes

The TRACON manages traffic from the Center boundary until a point at which all streams are merged. At this point, control is passed to the airport tower. It is possible for aircraft to make lateral changes to their STAR routes and fly a new route around a weather hazard, but it is important that this new route feeds into the same inner fix. The streams into an airport are complicated, and should be managed with the runway location in mind.

There are no strict rules in the terminal area. Anything goes, and flights are not required to follow specific routes. In the case of a weather event, the pilot usually has the best perspective and knows best what the weather is like, so for example, may request to deviate to the left around a storm. The controller on the other hand, may not know if the left or right is better. This happens all the time.

Currently, the movement of metering fixes is done occasionally during weather events. However, the blockage of a fix is typically hard to plan for.

For these reason, any changes to approach routes are typically made tactically. To give another example, consider two streams coming southbound, which then split and each land on one runway (one splits east, one splits west). If the eastern stream is blocked, the terminal will call an upstream center and ask them to re-route that flow in some way, so that they enter the TRACON at a different point. The terminal control manages the new stream.

Since streams do a lot of winding around near the airport, separation is often done by altitude. This is especially true in a metroplex like the CDG terminal.

Arrivals planning to use a given runway (say, if the runway orientation is fixed and there is only one runway choice) can enter the terminal from any direction. It

is always possible to re-orient these aircraft as needed. But, since there are many other routes (in a busy TRACON), the standard ones are used. Moving these standard routes to accommodate one new route which must be oriented in some way would be very complicated for controllers.

When asked if it could be easier for the controller to automatically recommend alternate, deconflicted routes, one controller mentioned that though in principle it could be easier for the controller, it is very tricky to predict weather.

Flexibility of sector boundaries

Making small movements to a sector boundary along with changes to standard arrival and departure routes would be **technically feasible** and makes sense from an efficiency and operational perspective. However, it would necessitate some small changes to current ATC practises. Furthermore, there is yet to be a solid solution proposed for the problem. If a “playbook” of potential routes existed, that would require a route infrastructure in place, as the controller working the airspace would need knowledge of the new fix.

The challenge is that forecasts are imprecise, and they really do not predict where weather will be exactly. So even if a fix is moved, it will likely be blocked anyway once weather materializes. Thus the concept is likely to degenerate to current operations, in which it is up to the pilot to deviate, and in which ATC gives ad hoc vectors.

Changes in sector direction

Changing the direction of a sector, for example by changing a departure sector to an arrival sector to accommodate an arrival push, is possible. It would not be too complex for controllers.

Appendix B

Pseudocode for 3D route planning

This appendix contains detailed pseudocode for the algorithms run for 3D planning in Chapter 5.

B.1 Details of the A* algorithm

This section provides more details for subprocedures of the A* algorithm (Algorithm 1) in Chapter 5.2.5, as implemented for the 3D route planning application.

CHILDREN

The procedure CHILDREN returns a list of child nodes for current node u . These child nodes are close neighbors of u , and represent lateral steps of distance STEPSIZE, a corresponding increase in altitude, and potentially a small displacement in heading, from u . In our experiments, STEPSIZE = 1 km. This section gives a detailed definition of the child nodes.

Let $u = (x_u, y_u, z_u, \theta_u, \gamma_u)$. Let parameter DEG_STEP be the heading stepsize for each child node, and let parameter NSTEPS be the number of steps of size DEG_STEP to take in each direction. The result of the CHILDREN procedure is upto $2\text{NSTEPS} + 1$ child nodes.

Given these paramters, we define the children of u as follows. The node $u_k = (x_{u_k}, y_{u_k}, z_{u_k}, \theta_{u_k}, \gamma_{u_k})$ is a child of u with heading $\theta_{u_k} = \theta_u + k\text{DEG_STEP}$, for $k =$

$-NSTEPS, \dots, -1, 0, 1, \dots, NSTEPS$. With this heading θ_{u_k} , the lateral position (x_{u_k}, y_{u_k}) of node u_k is set to coordinates of distance STEPSIZE km from u , in the new heading θ_{u_k} ; the altitude z_{u_k} is a function of the total distance flown so far, and can be computed from z_u using the altitude profiles from Section 5.3.3. Finally, the total heading change γ_{u_k} is set to $\gamma_u + k*\text{DEG_STEP}$.

Once each candidate child u_k is computed, it is discarded if either $\gamma_u > H$ ($H = 360^\circ$ was used in this thesis) or if u_k overlaps with restricted airspace. To determine the latter condition, a data structure of restricted airspace is maintained. This data structure is indexed by 3D position (rounded to the nearest integer), and each 3D cell contains an ordered list of restricted altitudes. This data structure is checked to determine if u_k is feasible.

RECONSTRUCT_PATH

The procedure RECONSTRUCT_PATH is run once the sink node is reached, and follows back-links from that sink node to the source node. This is done by keeping a data structure CAME_FROM in memory such that CAME_FROM[u] points to the node v along the shortest $u - v$ path. In our implementation, CAME_FROM is a hash table. The line of code “CAME_FROM[u] = v” is used to add a back-link to u at the point that the node v is added to the OpenQueue or updated in OpenQueue. Note that v is updated whenever v is still in OpenQueue and the cost associated with v is decreased. Specifically, this command is added following Algorithm 1, line 18.

B.2 Pseudocode for A^*-2 : IS_SINK and PRUNE

This section contains pseudocode corresponding to the A^*-2 algorithm from Chapter 5, Section 5.4.2.

IS_SINK

The procedure IS_SINK is a subprocedure of the A^*-2 algorithm that checks if the sink node has been reached. Specifically, for any predefined sink node t along the outer circle of the terminal C_O , the algorithm terminates once it encounters any node

t' which also intersects C_O and is within L km of the original sink t . The value of L used in experimental runs in Section 5.4.3 was 20 km.

The pseudocode is in Algorithm 2.

Algorithm 2 check if node u is a sink

```

procedure IS_SINK( $u, t$ )            $\triangleright$  current node  $u$ ; designated sink node  $t$ 
    radius $_u \leftarrow \|u\|$             $\triangleright$  lateral distance from airport ( $\sqrt{x_u^2 + y_u^2}$ )
    radius $_t \leftarrow \|t\|$ 
    deltaDist  $\leftarrow \|u - t\|$ 
    if  $|radius_u - radius_t| < 2$  AND deltaDist  $\leq L$  then
        return TRUE
    else
        return FALSE
    end if
end procedure

```

PRUNE

The procedure PRUNE is run at regular intervals as a speed improvement to the A^*-2 algorithm. Essentially, the procedure removes nodes u from OpenSet that are *close* to some other node v which *dominates* u .

Node $u = (x_u, y_u, z_u, \theta_u, \gamma_u)$ is *close* to node $v = (x_v, y_v, z_v, \theta_v, \gamma_v)$ if their 3D positions are equal when rounded to the nearest integer, and if their headings θ are equal once divided by 15 and rounded down. That is, when $\text{round}(x_u) = \text{round}(x_v)$, $\text{round}(y_u) = \text{round}(y_v)$, $\text{round}(z_u) = \text{round}(z_v)$, and $\text{floor}(\frac{\theta_u}{15}) = \text{floor}(\frac{\theta_v}{15})$. Furthermore, u *dominates* v if u is close to v and the cost associated with u is strictly within 1 unit of the cost associated with v (i.e. $f(u) - f(v) < 1$) and the total angle associated with u is at least 30 degrees less than the one for v (i.e. $\gamma_u \leq \gamma_v + 30$). Note that nodes which have large differences in heading are not close enough to each other for domination to be considered. Moreover, nodes which have substantially fewer accumulated turns are discarded for close-by nodes with similar cost and fewer total turns.

The pseudocode is in Algorithm 3.

Algorithm 3 decrease size of OpenSet by removing dominated nodes

```
procedure PRUNE(openQueue)      ▷ priority queue of non-dominated nodes
    prunedList ← ∅
    while openQueue.size() > 0 do
        [costu, u] ← openQueue.pop()
        prunedu ← u.getPrunedCoords()      ▷ increase granularity of state space
        if prunedList.hasKey(prunedu) then
            closeNodes ← prunedList.get(prunedu)
            for [costv, v] ∈ closeNodes do          ▷ nodes close to u
                diffCost ← costu - costv
                diffTotalAngle ← v.getTotalAngle() - u.getTotalAngle()
                if diffTotalAngle > 30 AND diffCost < 1 then
                    uIsAdmissible ← TRUE
                    closeNodes.pop(prunedv)      ▷ v is dominated by u; remove v
                else if diffTotalAngle > 30 AND diffCost >= 1 then
                    uIsAdmissible ← TRUE          ▷ keep both u and v
                else if diffTotalAngle ≤ 30 then
                    uIsAdmissible ← FALSE       ▷ u is dominated by v; skip u
                end if
            end for
            if uIsAdmissible then
                closeNodes.append([costu, u])      ▷ keep u
                prunedList[prunedu] ← closeNodes
            end if
        end if
    end while
    return prunedList      ▷ return the updated, pruned openQueue
end procedure
```

Bibliography

- A. Agresti and B. A. Coull. Approximate Is Better than “Exact” for Interval Estimation of Binomial Proportions. *The American Statistician*, vol. 52(2):pp. 119–126, 1998
- R. K. Ahuja, T. L. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993
- F. M. Alkoot and J. Kittler. Experimental Evaluation of Expert Fusion Strategies. *Pattern Recogn. Lett.*, vol. 20(11-13):pp. 1361–1369, 1999
- A. Basu, J. S. B. Mitchell, and G. Sabhnani. Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing. In *Proceedings of the 9th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 75–89. 2008
- P. Belobaba, A. Odoni, and C. Barnhart. *The Global Airline Industry*. John Wiley & Sons, 2005
- S. Benjamin, R. Bleck, J. Brown, K. Brundage, D. Devenyi, G. Grell, D. Kim, G. Manikin, T. Schlatter, B. Schwartz, T. Smirnova, T. Smith, and S. Weygandt. Mesoscale Weather Prediction with the RUC Hybrid Isentropic-Sigma Coordinate Model and Data Assimilation System. In *Symposium on the 50th Anniversary of Operational Numerical Weather Prediction*. NOAA Research/Forecast Systems Laboratory, 2004
- D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and Applications of Robust Optimization. 2007. Preprint
- D. Bertsimas and A. Odoni. A Critical Survey of Optimization Models for Tactical and Strategic Aspects of Air Traffic Flow Management. Technical report, NASA Ames Research Center, 1997
- D. Bertsimas and S. S. Patterson. The Traffic Flow Management Rerouting Problem in Air Traffic Control: A Dynamic Network Flow Approach. *Transportation Science*, vol. 34:pp. 239–255, 2000
- B. G. Brown, R. B. Bullock, C. A. Davis, J. H. Gotway, M. B. Chapman, A. Takacs, E. Gilleland, K. Manning, and J. L. Mahoney. New Verification Approaches

For Convective Weather Forecasts. In *11th Conference on Aviation, Range, and Aerospace Meteorology (ARAM)*. 2004

L. D. Brown, T. T. Cai, and A. DasGupta. Interval Estimation for a Binomial Proportion. *Statistical Science*, vol. 16(2):pp. 101–133, 2001

Bureau of Transportation Statistics. <http://www.transtats.bts.gov/>, 2010a. Retrieved October 26, 2010

Bureau of Transportation Statistics. Understanding the Reporting of Causes of Flight Delays and Cancellations. <http://www.bts.gov/help/aviation/html/understanding.html>, 2010b. Retrieved October 26, 2010

G. B. Chatterji and B. Gyarfas. Convective Weather Forecast Quality Metrics for Air Traffic Management Decision-Making. In *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*. AIAA, 2006

C. Chen, A. Liaw, and L. Breiman. Using Random Forest to Learn Imbalanced Data. Technical report, Department of Statistics, UC Berkeley, 2004. <http://www.stat.berkeley.edu/users/chenchao/666.pdf>

R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, vol. 32:pp. 505–536, 1985

D. Delahaye and S. Puechmorel. 3D Airspace Sectoring by Evolutionary Computation: Real-World Applications. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. ACM, New York, NY, USA, 2006

R. DeLaura and S. Allan. Route Selection Decision Support in Convective Weather: A Case Study. In *USA/Europe Air Traffic Management R&D Seminar*. 2003

R. A. DeLaura, M. Robinson, M. L. Pawlak, and J. E. Evans. Modeling Convective Weather Avoidance in Enroute Airspace. In *13th Conference on Aviation, Range, and Aerospace Meteorology (ARAM)*. American Meteorological Society, 2008

E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, , and A. Weingessel. *e1071: Misc Functions of the Department of Statistics, TU Wien*, 2009. R package version 1.5-19

N. Dougui, D. Delahaye, and M. Mongeau. A New Method for Generating Optimal Conflict Free 4D Trajectory. In *4th International Conference on Research in Air Transportation*. 2010

N. Durand and J.-M. Alliot. Ant Colony Optimization for Air Traffic Conflict Resolution. In *Eighth USA/Europe Air Traffic Management R&D Seminar, Napa, CA*. 2009

N. Durand, J.-M. Alliot, and J. Noailles. Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the 1996 ACM symposium on Applied Computing*, SAC '96, pp. 289–298. ACM, 1996

- Y. El Alj. *Estimating the true extent of air traffic delays*. Master's thesis, Massachusetts Institute of Technology, 2003
- J. Evans, M. Weber, M. Wolfson, D. Clark, and O. Newell. Roadmap for Weather Integration into Traffic Flow Management Modernization. Technical report, MIT Lincoln Laboratory, 2009
- FAA. FAA Aviation System Performance Metrics Data. <http://aspm.faa.gov/>, 2010. Retrieved December 10, 2010
- Federal Aviation Administration. FAA Aerospace Forecast, Fiscal Years 2009-2025. http://www.faa.gov/data_research/aviation/aerospace_forecasts/2009-2025/, 2009. Retrieved October 26, 2010
- P. Hart, N. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, vol. 4(2):pp. 100–107, 1968
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, corrected edn., 2003
- J. V. Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental Perspectives on Learning from Imbalanced Data. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pp. 935–942. ACM, New York, NY, USA, 2007
- IATA. IATA Financial Forecast. <http://www.iata.org/whatwedo/Documents/economics/Industry-Outlook-Sep-10.pdf>, 2010
- Joint Planning and Development Office. Next Generation Air Transportation System Integrated Plan. Technical report, 2004. http://www.jpdo.gov/library/NGATS_v1_1204r.pdf
- Joint Planning and Development Office. Concept of Operations for the Next Generation Air Transportation System, Version 2.0. Technical report, 2007. http://www.jpdo.gov/library/NextGen_v2.0.pdf
- Joint Planning and Development Office. ATM-Weather Integration Plan. Technical report, 2009
- M. P. Kay, J. L. Mahoney, and J. E. Hart. An Analysis of Collaborative Convective Forecast Product Performance for the 2005 Convective Season. In *12th Conference on Aviation Range and Aerospace Meteorology (ARAM)*. American Meteorological Society, 2006
- A. Klein, P. Kopardekar, M. D. Rodgers, and H. Kaing. “Airspace Playbook”: Dynamic Airspace Reallocation Coordinated with the National Severe Weather Playbook. In *7th AIAA Aviation Technology, Integration and Operations Conference*. 2007

- P. Kopardekar, K. Bilimoria, and B. Sridhar. Initial Concepts for Dynamic Airspace Configuration. In *7th AIAA Aviation Technology, Integration and Operations Conference*. 2009
- J. Krozel, C. Lee, and J. S. B. Mitchell. Turn-Constrained Route Planning for Avoiding Hazardous Weather. *Air Traffic Control Quarterly*, vol. 14(2):pp. 159–182, 2006
- J. Krozel, J. S. B. Mitchell, V. Polishchuk, and J. Prete. Capacity Estimation for Airspaces with Convective Weather Constraints. In *AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, South Carolina*. 2007
- K. Leiden, J. Kamienski, and P. Kopardekar. Initial Implications of Automation on Dynamic Airspace Configuration. In *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*. AIAA, 2007
- P.-C. B. Liu, M. Hansen, and A. Mukherjee. Scenario-based Air Traffic Flow Management: From Theory to Practice. *Transportation Research Part B: Methodological*, vol. 42(7-8):pp. 685–702, 2008
- Y. Liu, A. An, and X. Huang. Boosting Prediction Accuracy on Imbalanced Datasets with SVM Ensembles. In *Lecture Notes in Computer Science*, vol. 3918. Springer Berlin / Heidelberg, 2006
- J. L. Mahoney, S. Seseske, J. E. Hart, M. Kay, and B. Brown. Defining Observation Fields for Verification of Spacial Forecasts of Convection: Part 2. In *11th Conference on Aviation, Range, and Aerospace Meteorology (ARAM)*. 2004
- P. K. Menon, G. D. Sweriduk, T. Lam, G. M. Diaz, and K. Bilimoria. Computer-Aided Eulerian Air Traffic Flow Modeling and Predictive Control. *AIAA Journal of Guidance, Control and Dynamics*, vol. 29:pp. 12–19, 2006
- D. Michalek and H. Balakrishnan. Building a Stochastic Terminal Airspace Capacity Forecast from Convective Weather Forecasts. In *Proceedings of the Aviation, Range and Aerospace Meteorology Special Symposium on Weather-Air Traffic Management Integration, American Meteorological Society 89th Annual Meeting*. 2009a
- D. Michalek and H. Balakrishnan. Identification of Robust Routes using Convective Weather Forecasts. In *Eighth USA/Europe Air Traffic Management R&D Seminar, Napa, CA*. 2009b
- D. Michalek and H. Balakrishnan. Dynamic Reconfiguration of Terminal Airspace during Convective Weather. In *Proceedings of the 49th IEEE Conference on Decision and Control*. 2010
- J. S. Mitchell. On Maximum Flows in Polyhedral Domains. In *SCG '88: Proceedings of the Fourth Annual Symposium on Computational Geometry*, pp. 341–351. ACM, New York, NY, USA, 1988

J. S. B. Mitchell, V. Polishchuk, and J. Krozel. Airspace Throughput Analysis Considering Stochastic Weather. In *AIAA Guidance, Navigation, and Control Conference*. 2006

T. Mitchell. *Machine Learning*. McGraw Hill, 1997

NEXTOR. Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the United States. Technical report, The National Center of Excellence For Aviation Operations Research, 2010. http://www.isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf

A. Nilim and L. E. Ghaoui. Algorithms for Air Traffic Flow Management Under Stochastic Environments. In *Proceedings of the 2004 American Control Conference*, vol. 4, pp. 3429–3434. 2004

NOAA Aviation Weather Center. Collaborative Convective Forecast Product Product Description Document. <http://aviationweather.gov/products/ccfp/docs/pdd-ccfp.pdf>, 2010. Retrieved January 8, 2011

OPSNET. FAA Operations and Performance Data. <http://aspm.faa.gov/>, 2010. Retrieved October 26, 2010

OPSNET. FAA Operations and Performance Data. <http://aspm.faa.gov/>, 2011. Retrieved March 4, 2011

D. M. Pfeil and H. Balakrishnan. Identification of Robust Terminal-Area Routes in Convective Weather. *Transportation Science*, to appear in 2011

J. Prete and J. S. B. Mitchell. Safe Routing of Multiple Aircraft Flows in the Presence of Time-Varying Weather Data. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2004

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2008. <http://www.R-project.org>

SESAR Consortium. D1 - Air Transport Framework: The Current Situation. <http://www.sesarju.eu/sites/default/files/documents/reports/DLM-0602-001-03-00.pdf>, 2006

S. A. Seseske, M. P. Kay, S. Madine, J. E. Hart, J. L. Mahoney, and B. Brown. *2006: Quality Assessment Report: National Convective Weather Forecast 2 (NCWF-2)*, 2006. Submitted to FAA Aviation Weather Technology Transfer (AWTT) Technical Review Panel

K. Sheth, B. Sridhar, and D. Mulfinger. Impact of Probabilistic Weather Forecasts on Flight Routing Decisions. In *AIAA Aviation Technology, Integration and Operations Conference*. 2006

- L. Song, D. Greenbaum, and C. Wanke. The Impact of Severe Weather on Sector Capacity. In *Eighth USA/Europe Air Traffic Management Research and Development Seminar*. 2009
- D. Sun, S. D. Yang, I. Strub, A. M. Bayen, B. Sridhar, and K. Sheth. Eulerian Trilogy. In *Proceedings of the AIAA Conference on Guidance, Navigation and Control*. 2006
- T. M. Therneau and B. Atkinson. *rpart: Recursive Partitioning*, 2008. R package version 3.1-42. R port by Brian Ripley.
- S. S. Weygandt and S. G. Benjamin. RUC Model-Based Convective Probability Forecasts. In *11th Conference on Aviation, Range, and Aerospace Meteorology (ARAM)*. 2004
- S. S. Weygandt, S. G. Benjamin, T. G. Smirnova, J. M. Brown, and K. Brundage. Hourly Convective Probability Forecasts and Experimental High-Resolution Predictions Based on the Radar Reflectivity Assimilating RUC Model. In *13th Conference on Aviation, Range and Aerospace Meteorology*. American Meteorological Society, 2008
- M. Wolfson, B. Forman, K. Calden, W. Dupree, R. Johnson, R. Boldi, C. Wilson, P. Bieringer, E. Mann, and J. Morgan. Tactical 0-2 Hour Convective Weather Forecasts for FAA. In *11th Conference on Aviation, Range and Aerospace Meteorology*. 2004
- A. Yousefi. *Optimum Airspace Design with Air Traffic Controller Workload-Based Partitioning*. Ph.D. thesis, George Mason University, 2005