Analysis and Uncertainty of Airport Pushback Rate Control Policies

by

Patrick Kough McFarlane

B.S., University of Notre Dame (2014)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

Author ..... Signature redacted
Department of Aeronautics and Astronautics
December 17, 2015

Certified by ..... Signature redacted
Hamsa Balakrishnan
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ..... Signature redacted
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Analysis and Uncertainty of Airport Pushback Rate Control Policies

by

Patrick Kough McFarlane

Submitted to the Department of Aeronautics and Astronautics
on December 17, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis analyzes the effects of two algorithms that control the departure of aircraft at congested airports, with an emphasis on the uncertainty of the underlying processes. These algorithms, N-control and dynamic programming, belong to a broader class of control policies called Pushback Rate Control (PRC) policies that calculate a pushback rate for departing aircraft based on the state of the airport surface congestion. During times of congestion, these algorithms limit the amount of aircraft on the airport surface while maintaining departure throughput. This reduces the taxi-out time of aircraft, resulting in reduced fuel burn and emissions. This thesis introduces the policies and simulates their performance at LaGuardia Airport while varying two policy parameters, the length of the prediction interval and the number of prediction intervals, under several types of uncertainty, including the departure schedule and arrival rate. As will be shown, each policy results in significant taxi-out time reductions, saving airlines at least 60,000 minutes of taxiing over a 2-month period with the traditional 15-minute time window simulations. However, when accounting for the uncertainty in the algorithm inputs or the variation of policy parameters, the performance of both PRC policies degrades. By accounting for the variation of policy parameters and the different sources of uncertainty that affect airport surface management, the main contribution of this thesis provides a realistic analysis of PRC policies.

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

Acknowledgments

This research was funded by the Federal Aviation Administration.

I would like to express my sincerest gratitude to my academic and thesis advisor, Professor Hamsa Balakrishnan. Since my first day at MIT, Professor Balakrishnan has made me feel truly welcome. In addition, her passion for the research we have been doing has transformed me into a better student and researcher. I have greatly enjoyed researching airport surface management, and Professor Balakrishnan has taught me a great deal about statistical modeling, stochastic processes, and the aviation industry. I am very thankful for all that she has done for me.

Next, I would like to thank my family. It is difficult to articulate the positive influence of my family on my life. Rather than citing specific examples, I believe all of the love and efforts of my family constantly surround me and support me in all of my pursuits, especially this degree. Research is a nonlinear function of progress, frustration, time, and reward. My mother and father, Deborah and Scott McFarlane, have been my foundation throughout my entire life, while my brothers, Ryan and Sean McFarlane, have been my guideposts. For their unwavering support, I am eternally grateful.

Lastly, I would be remiss if I did not mention my colleagues in ICAT. Thank you for all of your help and guidance with classes and research. I feel lucky to have learned so much from you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2014, airlines in the United States operated 7.7 million departures [12] from the 77 Aviation System Performance Metrics (ASPM) airports and consumed 10.3 billion gallons of fuel [13]. With such a high level of traffic, congestion can occur, resulting in delays. Specifically, in the taxi-out phase of flight, the average delay across the entire United States is greater than 5 minutes, with LaGuardia Airport in New York leading the nation with an average taxi-out delay of more than 12 minutes [5]. During congestion, these taxi-out delays result from aircraft waiting in the departure queue. Because the engines are running in the departure queue, these delays have additional costs in terms of fuel burn and emissions. A widely noted 2008 study by the Joint Economic committee found that delays in 2007 resulted in 7.07 million metric tons of CO_2 emissions from 740 million gallons of fuel consumed, costing an estimated $1.6 billion [6]. These high costs to both the economy and the environment stem from all sources of delays, some of which, such as weather and unforeseen maintenance issues, are unavoidable. However, delays resulting from the length of the departure queue can be reduced or redistributed through the control of departing aircraft. While the control of the departure process can limit congestion, careful consideration must be given to the uncertainty surrounding both airport operations and the control algorithms. The benefits of limiting airport surface congestion must be weighed against the costs of changing airport operations. Also, the accuracy of control algorithms depends on the accuracy of the data required. The goal of this thesis is the development

and application of control algorithms that mitigate congestion, while maintaining a focus on the uncertainty of the overall process.

1.1 Motivation

Congestion occurs when the number of aircraft on the airport surface exceeds an amount necessary to maintain the departure throughput. The additional aircraft provide no benefit in terms of departure capacity, so their presence on the surface simply adds to the cost of congestion. Airport surface management represents one approach to address the problem of congestion.

Airport surface management controls the aircraft operating at an airport so as to reduce congestion while maintaining departure and arrival throughput. Several control algorithms have been developed to mitigate congestion by controlling the rate at which departing aircraft push back from their gates. These algorithms are called Pushback Rate Control (PRC) policies. Two of these policies, N-control and dynamic programming, will be thoroughly examined in this thesis. PRC policies work within the existing airline schedules by holding certain flights during times of congestion.

While PRC policies seem to be straightforward, many constraints and limitations surround the implementation of any airport surface management policy. One subset of these constraints includes the effects of PRC policies on airport operations. Currently, once aircraft receive authorization from the control tower, departures push back from their gates, regardless of the state of the airport surface. Aircraft are usually ready to push back relatively close to their scheduled departure time, which allows for push-back and maintenance crews to be adequately assigned to flights. Holding an aircraft through PRC policies has the potential to interfere with the crew schedules, which could cause further delays. Airlines must be capable of the dynamic scheduling of aircraft crews that accounts for the possibility of held aircraft due to congestion. Also, holding aircraft at their gates increases the chance of gate conflicts. Gate conflicts occur when an arrival lands at an airport and a departure is still at the gate to which the arrival is scheduled. Because gates are occupied by departures for a longer period

of time, gate conflicts are more likely to occur during PRC policy implementation. Because gate conflicts add to airport surface congestion and increased traffic, these are highly undesirable. The last airport operations constraint is the increase of the workload of air traffic controllers. Because controllers must implement the policy, the controllers must learn new procedures in addition to their current duties. The effects of PRC policies on airlines, gate conflicts, and controllers must be considered.

In addition to the operational constraints, the PRC policies require real-time and forecasted data from arrivals and departures. The surface traffic level, future departures, future arrivals, airport capacity, and weather all represent variables required by the PRC algorithms to accurately calculate a departure pushback rate. While the surface traffic level can be observed, future operations, capacity, and weather must be predicted. These predictions represent another source of uncertainty. If the algorithm inputs cannot be predicted with an acceptable degree of accuracy, the departure pushback rate of the PRC policies is not accurate. This could result in poor surface management because a high pushback rate during congestion could lead to increased congestion and more delays, while a low pushback rate during with low congestion could lead to a decrease in departure throughput. Both of these results are undesirable, as one is not effective at mitigating congestion, while the other one "starves" the runway by not maintaining runway utilization.

The last source of uncertainty is the use of PRC policies. PRC policies calculate departure pushback rates that are valid for a certain time window. Historically, the choice of this time window has been 15 minutes. A 15-minute time window for the pushback rate usually allows aircraft pushing back in one 15-minute window to reach the runway by the next 15-minute window. However, there may be reasons to adjust the length of the time window. A shorter time window leads to more accurate departure pushback rates because the rates are updated more frequently and forecasted data are not projected far into the future. However, a shorter time window increases the workload for controllers due to the need to frequently update the departure pushback rate. A longer time window decreases this workload, but at the cost of less accurate departure pushback rates due to the inaccuracy of predictions further

into the future. This tradeoff is an important consideration for controllers, airlines, and airports when implementing PRC policies. In addition to the time window, the time horizon is also of importance for PRC policies. The time horizon is defined as the number of time windows that a PRC policy "looks ahead" and calculates a pushback rate for a time window in the future. Most PRC policies use a time horizon of 0, meaning that the PRC policy updates the pushback rate at the end of each time window. The same tradeoff between accuracy and workload exists for time horizons as well.

Reducing congestion is a strong motivating factor for this research, but the uncertainty involved in achieving that goal is arguably just as important. PRC policies are an effective way to reduce congestion while not decreasing operations or altering airport capacity. However, implementation of PRC policies introduces potential issues in terms of airport operations and policy accuracy. A thorough examination of both PRC policies and the associated uncertainties illustrates a complete picture of the costs and benefits of airport surface management.

1.2 Literature Review

1.2.1 PRC Policies

The two surface management strategies examined in this thesis are the N-control and dynamic programming PRC policies. These policies lead to reductions in fuel burn and emissions, minimizing the impact of surface operations, as analyzed by many studies [21] [22] [16] [10]. Simaiakis et al. develop an N-control simulation for Boston Logan International Airport and reports important metrics to describe the effects of airport surface management [21]. They then go on to calculate those metrics for several major airports and analyze emissions and fuel burn [22]. Ravizza et al. demonstrate the relationship between airport surface movement and fuel burn [16]. Khadilkar examines the control of both departures and arrivals on an individual aircraft basis [10].

Simaiakis [18] [19] lays the foundation for controlling departure processes. He develops an estimation of airport capacity, unimpeded taxi-out times, and a dynamic programming algorithm for Boston Logan International Airport. Many of these techniques will be used in this thesis to develop models for LaGuardia Airport and extend the study of airport surface management to include the variation of policy parameters and operational uncertainty. Simaiakis also notes that PRC policies are flow-based approaches to airport surface management, which means that these policies use virtual queues by holding aircraft at their gates. The use of virtual queues were initially suggested [9] and proposed in separate studies [1]. Feron et al. [9] give a detailed overview of the conceptual departure control process, culminating in the idea of using virtual queues to mitigate congestion. Burgain et al. [1] use virtual queues to minimize a cost function related to passenger wait time.

As for the N-control policy, several studies have developed the theoretical framework [14] [3] [18] [19] and the implementation procedures are demonstrated in a case study at Boston Logan International Airport [17]. Pujet et al. [14] introduce an N-control algorithm while also suggesting the possibility of using dynamic programming to help address some of the uncertainties associated with the N-control policy. Carr et al. [3] develop a software tool to simulate the departure process at airports to support the research of airport surface management. Burgain et al. [2] derive a full-state feedback algorithm to control departures, where the algorithm uses a cost structure based on the number of aircraft taxiing and the non-utilization of the runway. This thesis expounds upon the work of a potential implementation proposed for LaGuardia Airport [11] [20], in which the authors performed an N-control simulation for the airport focusing on the length of gateholding times allowed by the algorithm.

For dynamic programming, a recent paper introduced this method of departure metering [23]. This method accounts for the underlying uncertainty of the airport capacity by modeling the state of the airport surface as a semi-Markov process. The optimal pushback rate is calculated based on a cost of queuing function and the probability of the airport surface being in a given state. However, the policy still relies on obtaining an arrival rate and weather prediction to build the model, which intro-

duces uncertainty. The remaining relevant work with dynamic programming applied to airport operations only focuses on the optimization of aircraft scheduling. Rathinam et al. [15] propose a dynamic programming approach for the departure schedule that finds the optimal pushback schedule for a given amount of departing aircraft. Chandran and Balakrishnan [4] also use a dynamic programming algorithm for the departure schedule, but they account for the uncertainty and random deviations inherent in the departure process. Dell'Olmo and Lulli [7] consider both arrivals and departures with dynamic programming and the tradeoff and interactions between the two different types of flights at an airport.

1.2.2 Variation of Policy Parameters

The structure of the PRC policies includes the time window and time horizon. The length of the time window for which a pushback rate is valid can be changed. Researchers to this point generally set this time window to be 15 minutes, reasoning that the time window should roughly equal the lead time of the system [17] [23]. The lead time of the departure process is the time it takes for the runway to experience a given pushback rate. A runway experiences a pushback rate when the aircraft subjected to that rate begin to arrive at the runway. Rathinam et al. [15] only calculate a departure schedule for a given number of aircraft, while Dell'Olmo and Lulli [7] appear to settle on a time window of 15 minutes for their analysis. Burgain et al. [2] use a sampling time of 1 minute for their full-state feedback algorithm. In terms of time horizons, most studies do not look ahead, meaning that the PRC policy updates the pushback rate at the end of each time window. Longer time horizons would mean that a PRC policy calculates a pushback rate for a time window further into the future.

1.2.3 Uncertainty

While the literature review introduces the state of research concerning airport surface management, the issue of uncertainty remains a crucial problem. The departure

planner provides a introductory discussion of uncertainty and its role in controlling the departure process [9]. The sources of uncertainty cited include weather, airline operations, air traffic operations, and human factors. These sources of uncertainty all have an effect on the data required by PRC policies to calculate departure pushback rates. Simaiakis and Balakrishnan demonstrate the uncertainty in the taxi-out time of aircraft, indicating the taxi-out time is a stochastic process [21]. Others have used this approach as well [2] [14] [3]. Chandran and Balakrishnan [4] address the uncertainty that results from perturbations in the departure and arrival schedules.

1.2.4 Contributions of this thesis

This thesis summarizes the research efforts to understand the effects of policy parameter variation and uncertainty on PRC policies. The main contributions of this thesis are:

1. The analysis of policy parameter variation for the length of time windows and horizons of PRC policies. 15-minute, 30-minute, and 60-minute time windows are examined to weigh the costs and benefits associated with different time window lengths. For time horizons, the analysis explores the use of looking ahead into the future to calculate future departure pushback rates. These "look aheads" will be for 15-minute time windows. For example, the 0 look-ahead is equivalent to the 15-minute time window analysis and 1 look-ahead calculates a departure pushback rate for the 15-minute time window following the current 15-minute time window.

2. The analysis of operational uncertainty stemming from the stochastic departure and arrival rates in a given time period. Scheduled departure times for aircraft are rarely met, as aircraft are often ready before or after that time. Arrivals often arrive before or after their scheduled times as well. PRC policies require accurate arrival rates for a time window to ensure an accurate departure pushback rate. Also, PRC policies rely on sustained demand, which is not always the case at airports.

3. The development of two PRC policies for LaGuardia Airport with particular attention paid to gate conflicts. The simulations of these policies include the potential uncertainties discussed above, providing a more realistic evaluation of PRC policies. This thesis also discusses other possible constraints that may accompany the implementation of airport surface management.

1.3 Thesis Organization

This introduction describes airport surface management and the uncertainties associated with both PRC policies and airport operations. Chapter 2 gives a thorough description of the two PRC policies of interest, N-control and dynamic programming. The algorithm development serves as the foundation for the rest of the thesis. Chapter 3 begins to look at the policy parameter variation of time windows and time horizons. For both PRC policies, the results include a sensitivity analysis for each policy parameter. Chapter 4 considers the operational uncertainties surrounding departure and arrival rates. Finally, Chapter 5 concludes the thesis and suggests future research areas.

Chapter 2

Pushback Rate Control Policies

Pushback Rate Control (PRC) policies belong to a broader class of techniques related to airport surface management. The goal of airport surface management is to control the number of active aircraft on the airport surface so as to reduce congestion. PRC policies accomplish this by mitigating the rate at which departing aircraft push back from their gates during times of congestion. This is in direct contrast to current operations at U.S. airports, which use a first-come, first-served (FCFS) approach for departing aircraft. When a departure is ready for pushback, the controllers allow pushback regardless of the state of the airport surface. Therefore, PRC policies shift the delays incurred by congestion from the airport surface to the departure gate. Instead of waiting in the runway queue with engines on, an aircraft absorbs the delay at the gate in a virtual queue with engines off. This reduces the fuel burn and emissions caused by congestion. However, this also increases the likelihood of a gate conflict in which an arrival lands while a departure is still parked at the arrival's gate. A later section explains the positives and negatives of PRC policies more closely, with particular attention paid to the actual implementation of these policies at airports. This section describes and explains two PRC policies of interest, N-control and dynamic programming.

2.1 N-Control

N-control is a PRC policy that uses the relationship between the number of departure aircraft on the airport surface N and the runway throughput in a particular time window. This relationship is best described through a saturation curve, which is shown in Figure 2-1 for runway configuration 31|4 at LGA under visual meteorological conditions (VMC), or segment (31|4; VMC), for a 15-minute window.

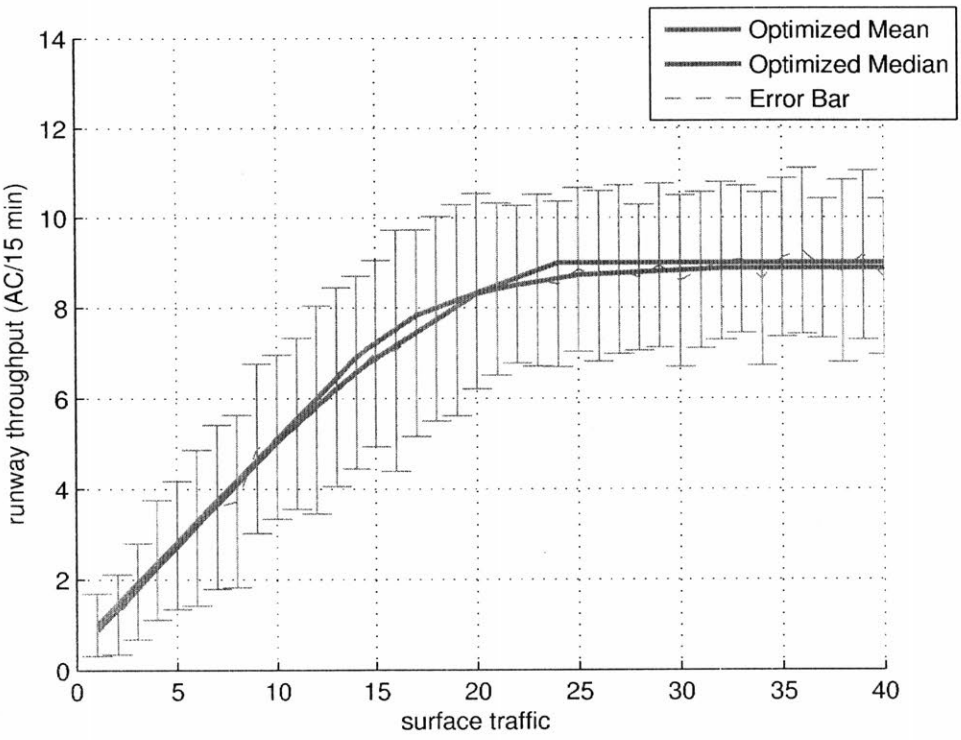

Figure 2-1: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for segment (31|4; VMC) with a 15-minute window.

Figure 2-1 contains n pairs of the departure throughput $T(t)$ and surface traffic $N(t)$ for each 15-minute period at LaGuardia Airport for 2013 for the given segment. The saturation curve is a least-squares regression fit to the data. As described by Simaiakis [19], the n pairs of $N(t)$ and $T(t)$ take the form of $(x_1, y_1), ..., (x_n, y_n)$. The saturation curve is a non-decreasing function $T = f_{mean}(N)$. At each value of N,

$f_{mean}(N)$ is found using simple convex optimization:

$$\min \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \tag{2.1}$$

subject to the following constraints:

$$\hat{y}_i = f_{mean}(u_i), i = 1, ..., n \tag{2.2}$$

$$f_{mean}(i+1) \geq f_{mean}(i), i = 0, ..., (n-1) \tag{2.3}$$

$$f_{mean}(i+1) - f_{mean}(i) \leq f_{mean}(i) - f_{mean}(i-1), i = 0, ..., (n-1) \tag{2.4}$$

While this solution is for the mean regression, the median regression can also be found:

$$\min \sum_{i=1}^{n} |\hat{y}_i - y_i|, \tag{2.5}$$

subject to the following constraints:

$$\hat{y}_i = f_{med}(u_i), i = 1, ..., n \tag{2.6}$$

$$f_{med}(i+1) \geq f_{med}(i), i = 0, ..., (n-1) \tag{2.7}$$

$$f_{med}(i+1) - f_{med}(i) \leq f_{med}(i) - f_{med}(i-1), i = 0, ..., (n-1) \tag{2.8}$$

Figure 2-1 has several interesting features. Note the nearly linear relationship between departure traffic and runway throughput for low levels of departure traffic. This indicates that, during times of low departure traffic, runway throughput benefits from an increase in departure traffic. However, the linear relationship does not extend for higher levels of departure traffic, and runway throughput hits a maximum. In words, more departure traffic leads to less of an increase in runway throughput until a certain point, after which runway throughput does not increase. This point, N^*, is called the saturation point, and it is unique to each segment at an airport. For the N-control policy, an acceptable level of surface traffic N_{ctrl} is chosen around N^*.

The goal of the N-control policy is to maintain surface traffic at or around N^* for a particular segment to maintain maximum departure throughput while decreasing unnecessary surface congestion. The pushback rate R for a time window is calculated using the following equation:

$$R = N_{ctrl} + T_p - N_{cur}, \tag{2.9}$$

where T_p is the predicted throughput for that time window and N_{cur} is the number of departing aircraft taxiing at the beginning of the time window. Calculating predicted throughput is described below. The N-control policy only controls the rate of departure aircraft when $N_{cur} > N_{ctrl}$ at the beginning of a time window. Examining Figure 2-1 reveals a tradeoff in the choice of N_{ctrl}. For $N_{ctrl} < N^*$, the N-control policy will be used more often, but the departure throughput could be less than the maximum departure throughput. Using the N-control policy more leads to more taxi-out time reduction benefits. However, this risks reducing the departure capacity of the airport, which is very undesirable. For $N_{ctrl} > N^*$, the N-control policy will be used less often, meaning congestion levels would be higher and the benefits of the policy decrease. The choice of N_{ctrl} for a particular segment must account for this tradeoff.

Notice the error bars for each value of surface traffic N in Figure 2-1. The saturation curve is the best fit line for each 15-minute period in 2013 at LaGuardia Airport. However, the standard deviation of throughput around the saturation point is about 2 aircraft. This indicates that factors other than surface traffic affect the departure throughput. These factors include weather, arrival aircraft, and human factors. Two of these factors, weather and arrival aircraft, can be used to more accurately predict departure throughput during times of congestion.

With the machine learning technique of regression trees, departure throughput can be predicted during times of congestion ($N_{cur} > N_{ctrl}$). Using empirical data, the regression trees calculate the predicted departure throughput based on the arrival rate and Route Availability Planning Tool (RAPT) for the next time window. RAPT is a

tool used to estimate the location and severity of weather in the area surrounding an airport. RAPT is on a scale of 0 to 3, where 0 indicates no weather and 3 represents very severe convective weather. A RAPT value is given for several areas around the airport, as well as 15 minutes into the past and 15 minutes into the future in 5-minute increments. To get a single value, the RAPT is averaged over all time increments and all areas around the airport. Because the regression trees only predict throughput during times of congestion, they capture the uncertainty present in the maximum departure throughput in Figure 2-1. Figure 2-2 shows an example of a regression tree for LaGuardia Airport.

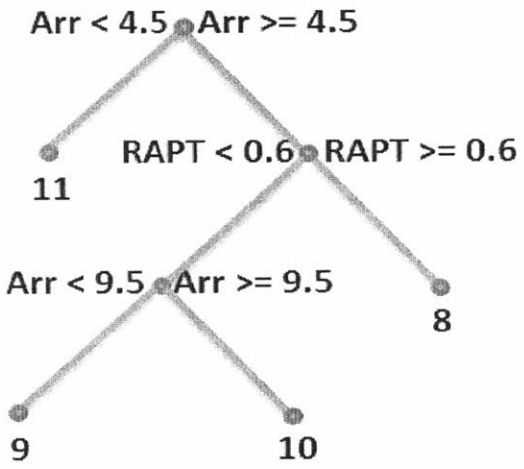

Figure 2-2: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for segment (31|4; VMC) with a 15-minute window. "Arr" indicates the arrival rate for the time window and "RAPT" indicates the single RAPT value for the time window.

With all of the pieces of the N-control policy introduced, the N-control algorithm proceeds in the following manner. At the beginning of a time window of arbitrary length, a controller observes the current departure traffic on the airport surface N_{cur}, the single RAPT value, and the expected arrival rate for the time window. If $N_{cur} > N_{ctrl}$, the airport departure traffic is greater than the acceptable level of traffic, so

the N-control policy controls the pushback rate of departing aircraft for the next time window. If $N_{cur} \leq N_{ctrl}$, the pushback rate of departing aircraft is not controlled and pushbacks are FCFS based on scheduled departure time. If N-control is in effect, a departure pushback rate needs to be calculated. With the RAPT value and arrival rate, the regression tree for a particular segment calculates the predicted throughput for that time window. With this, all of the independent variables of Equation 2.9 are known and the departure pushback rate is found.

For the simulations of the N-control policy, the departure pushback rate creates equally-spaced pushback slots for departures in that time window. Each departure is then mapped to a unique pushback slot in the order of their scheduled departure time. If the number of scheduled departures in a time window exceeds the number of pushback slots, the departures without a pushback slot must wait until the next time window. If the scheduled pushback time for an aircraft is after the pushback slot, the aircraft pushes back at the scheduled departure time. If the scheduled pushback time for an aircraft is before the pushback slot, the aircraft is held at the gate until the pushback slot time arrives. This holding time corresponds to time spent at the gate with the engines off. Under the FCFS policy, the aircraft spends that holding time waiting in the departure queue with the engines on. Therefore, the N-control policy shifts the congestion-induced delay from the departure queue to the gate. This reduces the total taxi-out time of aircraft and, because the engines are off, fuel burn is also decreased. This reveals how the benefits of reduced taxi-out time and fuel burn result from the use of the N-control policy.

For the policy to accurately simulate airport operations, a reliable taxi-out time for each aircraft must be found. The total taxi-out time in the simulations consists of two parts: the unimpeded taxi-out time and time spent waiting in the departure queue. The unimpeded taxi-out time is the time it takes an aircraft to taxi to the departure runway when departure traffic is low or nonexistent. So, the simulation uses this as a proxy for the amount of time it takes an aircraft to reach the departure queue from each terminal at an airport. Of course, taxi-out times depend not only on the terminal from which an aircraft departs, but also the gate from which an aircraft departs.

However, analyzing taxi-out times from each gate is impractical for several reasons. Certain gates may not have taxi-out times in the dataset. For the gates represented in the dataset, the sample size may be too small to get an accurate estimate. For these reasons, the unimpeded taxi-out times from each terminal represent a reasonable approximation for unimpeded taxi-out times for aircraft.

The calculation of unimpeded taxi-out time requires the taxi-out time during different levels of surface traffic. For each flight in the Aviation Specific Performance Metric dataset (ASPM), the taxi-out time is simply the difference between the wheels-off time and pushback time. However, the definition of surface traffic for this application is nontrivial. The number of departing aircraft on the airport surface almost certainly changes from the pushback time of an aircraft and the wheels-off time of that same aircraft. For this reason, adjusted surface traffic N_{adj} serves as the metric describing the state of the departure traffic. N_{adj} for an aircraft is the sum of the number of departing aircraft on the airport surface at pushback time and the number of aircraft that push back after that time but before the original aircraft's wheels-off time. This definition indicates that N_{adj} represents the maximum number of departure aircraft that a departure can expect to encounter on the airport surface. N_{adj} can also be extracted from ASPM data, so the calculation of unimpeded taxi-out time can move forward. Simply using the departure traffic at the time of departure pushback may underestimate the traffic conditions that an aircraft experiences.

Figure 2-3 shows a scatter plot of all flights from Terminals C and D at LGA in April 2014. Notice the concentration of points for lower levels of adjusted traffic. Taxi-out times with a lower level of adjusted traffic appear to have less variability than taxi-out times with higher adjusted traffic. This agrees with intuition. With lower levels of departure traffic, an aircraft taxiing is more likely to achieve the unimpeded taxi-out time. With higher levels of departure traffic, an aircraft taxiing may encounter many impediments on the airport surface.

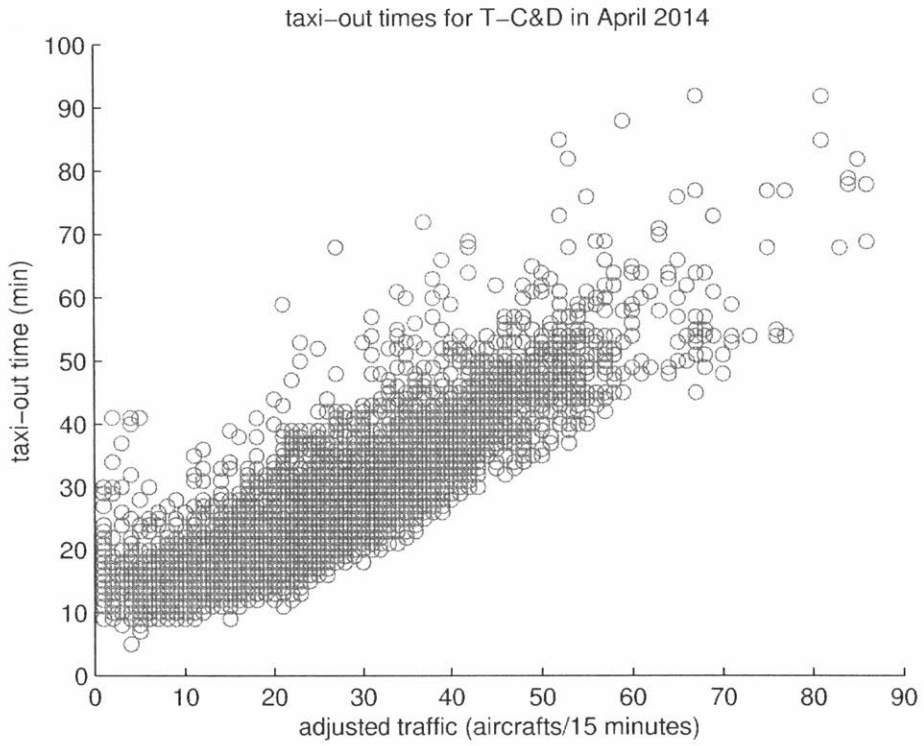

Figure 2-3: Scatter plot with taxi-out time as a function of the adjusted traffic for flights from terminals C and D at LGA. Data from April 2014.

Similar to saturation curves, the scatter plot can be fit using convex optimization regression. Figure 2-3 contains n pairs of the taxi-out time $\tau(t)$ and adjusted surface traffic $N_{adj}(t)$ for each flight from Terminals C and D at LaGuardia Airport for April 2014. The n pairs of $N_{adj}(t)$ and $\tau(t)$ take the form of $(x_1, y_1), ..., (x_n, y_n)$. The curve is a non-decreasing function $\tau = f_{mean}(N_{adj})$. At each value of N_{adj}, $f_{mean}(N_{adj})$ is found using simple convex optimization:

$$\min \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \tag{2.10}$$

subject to the following constraints:

$$\hat{y}_i = f_{mean}(u_i), i = 1, ..., n \tag{2.11}$$

$$f_{mean}(i+1) \geq f_{mean}(i), i = 0, ..., (n-1) \tag{2.12}$$

$$f_{mean}(i+1) - f_{mean}(i) \geq f_{mean}(i) - f_{mean}(i-1), i = 0, ..., (n-1) \tag{2.13}$$

While this solution is for the mean regression, the median regression can also be found:

$$\min \sum_{i=1}^{n} |\hat{y}_i - y_i|, \tag{2.14}$$

subject to the following constraints:

$$\hat{y}_i = f_{med}(u_i), i = 1, ..., n \tag{2.15}$$

$$f_{med}(i+1) \geq f_{med}(i), i = 0, ..., (n-1) \tag{2.16}$$

$$f_{med}(i+1) - f_{med}(i) \geq f_{med}(i) - f_{med}(i-1), i = 0, ..., (n-1) \tag{2.17}$$

Solving for the best fit results in Figure 2-4. The unimpeded taxi-out time is then the value of this fit when $N_{adj} = 0$. This process is repeated for each combination of runway configurations and terminals. Terminals C & D are combined for this calculation due to their proximity. Note, the unimpeded taxi-out time calculation does not include separate results for the different meteorological conditions. While taxi-out times may differ between VMC and IMC, this analysis assumes that the unimpeded taxi-out times for the two meteorological conditions remain the same. The motivation for this assumption mirrors the sample size and reasonable approximation arguments made for excluding individual gates from the unimpeded taxi-out time calculations.

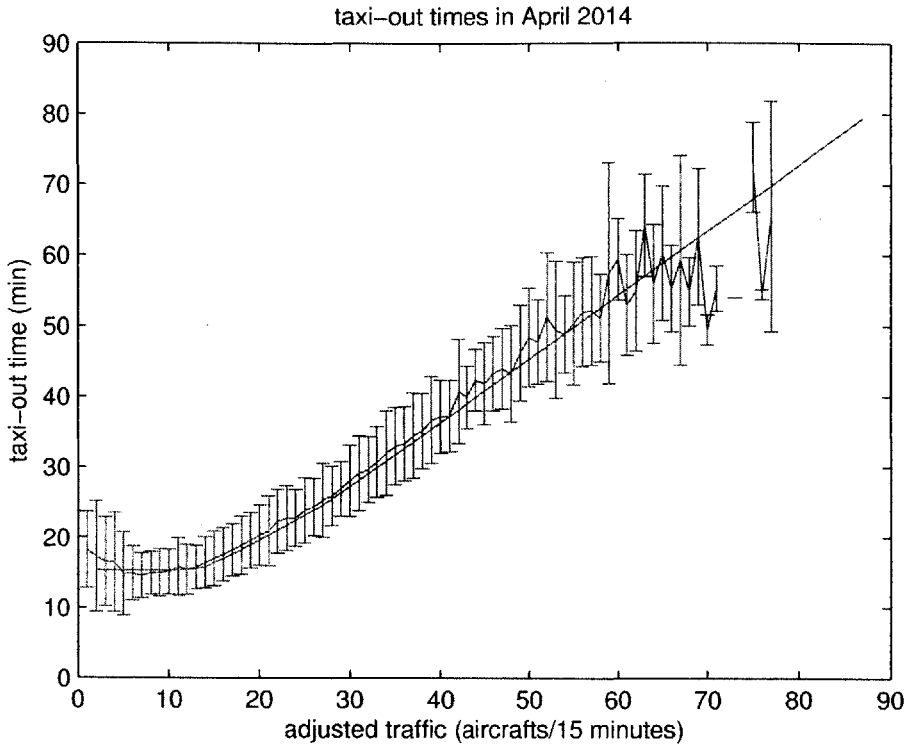

Figure 2-4: Best fit of the empirical data with taxi-out time as a function of the adjusted traffic for flights from terminals C and D at LGA with runway configuration 22|13. Data from April 2014.

Because aircraft spend more time at the gate under the N-control policy, the likelihood of a gate conflict increases. Recall that a gate conflict occurs when an arriving aircraft lands while a departing aircraft still occupies that arrival's gate. Gate conflicts are undesirable as they introduce an additional complexity to the airport surface. Often, the arrival must wait on taxiways or other areas until the departure is ready for pushback and clears the gate area. However, the additional gate conflicts caused by the N-control policy consist of departures ready to depart because they are being held at the gate after their scheduled departure time. To solve the gate conflict issue, if an arrival lands and is headed to a gate with a departure being held by the N-control policy, the departing aircraft is immediately cleared for pushback. This solution ensures that the departure will clear the gate area before the arrival

reaches the gate.

Gate conflicts only account for one of the costs of implementing an airport surface management policy. This section describes the N-control policy without varying the policy parameters or accounting for operational uncertainty. These additional considerations greatly influence the effectiveness of airport surface management.

2.2 Dynamic Programming

The dynamic programming policy models the state of the airport surface as a Markov process with the state described by the number of aircraft taxiing to the runway and the number of aircraft queuing at the runway. By modeling the runway service times as an Erlang distribution[19] with the shape and rate $(k, k\mu)$, the transition probabilities over a time window are found by numerically integrating the Chapman-Kolmogorov equations, which are described below. The runway service time is the time between successive takeoffs on a runway, meaning that a service time is the time it takes the aircraft at the head of the queue to leave the airport surface. Dynamic programming then uses value iteration to find the optimal pushback policy in terms of the costs of queuing and runway utilization.

The dynamic programming policy contrasts with the N-control policy in the following manner. N-control uses a simple equation to maintain departure surface traffic at a predetermined level based on empirical data. Dynamic programming models the runway service time to get a probability distribution of the state of the airport at some point in the future. With this, and a cost of queuing and runway utilization function, dynamic programming finds the departure pushback rate that minimizes costs. While N-control predicts that the state of the airport surface will evolve in a certain manner, dynamic programming considers all of the potential states of the airport and the departure pushback rate accounts for the uncertainty in the evolution of the state of the airport. From this perspective, dynamic programming is a more robust policy than N-control.

The shape and rate $(k, k\mu)$ of the Erlang distribution of runway service times can

be found using the method of moments. Each leaf of the regression trees introduced in Figure 2-2 and described above contains many observations of departure throughput. Because these empirical observations are Erlang distributed, the event of having exactly i services during a time period Δ is a Poisson random variable. The two moments μ_1 and μ_2 are:

$$\mu_1 = \sum_{i=0}^{\infty}(i \cdot \sum_{j=(i-1)k+1}^{(i+1)k-1} \frac{k-|ik-j|}{k} \cdot e^{-k\mu\Delta} \cdot \frac{(k\mu\Delta)^j}{j}), \tag{2.18}$$

$$\mu_2 = \sum_{i=0}^{\infty}(i^2 \cdot \sum_{j=(i-1)k+1}^{(i+1)k-1} \frac{k-|ik-j|}{k} \cdot e^{-k\mu\Delta} \cdot \frac{(k\mu\Delta)^j}{j}) \tag{2.19}$$

The method of moments, with the condition that the shape k must be a natural number, is done in the following manner. First, a numerical solution to Equation 2.18 can be found for different values of k. Then, the error of Equation 2.19 is found for each value of k. These steps repeat until the absolute error increases. Once this happens, the last shape k and rate $k\mu$ describe the Erlang distribution for a given leaf of a regression tree. The shape k and rate $k\mu$ give the lowest absolute error of Equation 2.19. This is done for each leaf of all of the regression trees. For each airport, a solution is found for each shape of the Erlang distribution of service times. These solutions may contain different segments. If so, the rate $k\mu$ is averaged over all segments of the same shape k. This is an approximation, but the rates of the distributions of equal shape do not differ drastically. Therefore, the averaging does not greatly affect the accuracy of the final solution. This improves upon the work of Simaiakis [19], which uses one shape and rate for the entire airport.

At the beginning of the time window, the state (r, q) is observed, where r is the number of aircraft taxiing to the runway and q is the stages-of-work to be completed at the runway. The stages-of-work are the product of the number of aircraft queuing and the shape of the Erlang distribution. The shape and rate have already been calculated using the methods described above. The runway configuration, weather conditions, and arrival rate dictate which shape and rate apply. With the state known at the beginning of the time window (R_0, Q_0) and the runway capacity C, the

following Chapman-Kolmogorov equations are then solved throughout the entire time window of length Δ to get the probability $P_{r,q}(\Delta)$ of the airport surface being in a given state at the end of the time window:

$$\frac{dP_{0,0}}{dt} = k\mu P_{0,1} \tag{2.20}$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} - k\mu P_{0,q}, \qquad 1 \leq q < k \tag{2.21}$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} + \frac{1}{\Delta - t} P_{1,q-k} - k\mu P_{0,q}, \qquad k \leq q < kC \tag{2.22}$$

$$\frac{dP_{0,kC}}{dt} = \frac{1}{\Delta - t} P_{1,k(C-1)} - k\mu P_{0,kC} \tag{2.23}$$

$$\frac{dP_{r,0}}{dt} = k\mu P_{r,1} - \frac{r}{\Delta - t} P_{r,0} \tag{2.24}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} - k\mu P_{r,q} - \frac{r}{\Delta - t} P_{r,q}, \qquad 1 \leq q < k \tag{2.25}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} + \frac{r+1}{\Delta - t} P_{r+1,q-k} - k\mu P_{r,q} - \frac{r}{\Delta - t} P_{r,q}, \qquad k \leq q \leq k(C-1) \tag{2.26}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} + \frac{r+1}{\Delta - t} P_{r+1,q-k} - k\mu P_{r,q}, \qquad k(C-1) < q < kC \tag{2.27}$$

$$\frac{dP_{r,kC}}{dt} = \frac{r+1}{\Delta - t} P_{r+1,k(C-1)} - k\mu P_{r,kC} \tag{2.28}$$

$$\frac{dP_{R_0,0}}{dt} = k\mu P_{R_0,1} - \frac{R_0}{\Delta - t} P_{R_0,0} \tag{2.29}$$

$$\frac{dP_{R_0,q}}{dt} = k\mu P_{R_0,q+1} - (\frac{R_0}{\Delta - t} + k\mu) P_{R_0,q}, \qquad 1 \leq q \leq k(C-1) \tag{2.30}$$

$$\frac{dP_{R_0,q}}{dt} = k\mu P_{R_0,q+1} - k\mu P_{R_0,q}, \qquad k(C-1) < q < kC \tag{2.31}$$

$$\frac{dP_{R_0,kC}}{dt} = -k\mu P_{R_0,kC} \tag{2.32}$$

With these transition probabilities, the costs of releasing a number of aircraft with pushback rate λ can be found, with the assumption that aircraft traveling to the runway queue at the beginning of one time window reach the queue by the start of the next time window. The minimum cost $J^*(r,q)$ at each state is given by Bellman's equation for the infinite horizon problem with discount factor α:

$$J^*(r,q) = \min_{\lambda \in \Delta} \{\overline{c}(r,q) + \alpha \sum_{j=0}^{kC} P_{r,q} J^*(\lambda, j)\} \tag{2.33}$$

where $\overline{c}$ is an average cost of a state over a time period and Δ is the set of all possible pushback rates. This equation can be solved by value iteration. The pushback rate for the time window is then given by the pushback rate that minimizes the cost function.

The cost function must penalize both non-utilization of the runway and lengthy queues. Following the lead of Simaiakis [19], non-utilization of the runway has a constant cost H, while for $q > 0$, the cost is a non-decreasing function of q. This leads to the following equation

$$c(q) = \begin{cases} H & \text{if } q = 0, \\ (\frac{q-k}{k})^2 & \text{if } q > 0. \end{cases} \tag{2.34}$$

H is chosen by an airport to reflect the true cost of losing capacity by not maintaining runway utilization. The equation for $c(q)$ is only a function of queue length and service time shape, so time is not a factor here. Because dynamic programming accounts for all possibilities for the evolution of the airport state, the cost function must be combined with the probability that the runway queue is of a certain length. To add the time component, the vector of these probabilities is

$$\mathbf{p}_q(R_0, Q_0, t) = [\sum_{r=0}^{R_0} P_{r,0}(t), \sum_{r=0}^{R_0} P_{r,1}(t), ..., \sum_{r=0}^{R_0} P_{r,kC}(t)]. \tag{2.35}$$

In words, the above equation states that, given that the state of the airport was (R_0, Q_0) at the beginning of the time window, these are the probabilities that the runway queue consists of q stages-of-work at time t. Now, with the probability of

runway queue length as a function of time, the product of these probabilities and the cost function can be summed over an entire time window Δ to find the expected cost of each state:

$$\bar{c}(R_0, Q_0) = \sum_{i=0}^{10\Delta-1} \frac{1}{10} \mathbf{p}_q(R_0, Q_0, i/10) \cdot c(q). \tag{2.36}$$

Because Equation 2.35 is sampled 10 times a minute, the summation in Equation 2.36 reflects this sampling. With the expected cost over a time window, Equation 2.33 is solved to find the optimal pushback rate.

The solution over all states for a 15-minute time window can be seen in Figure 2-5 for an Erlang distribution of service times with shape $k = 2$ at LaGuardia Airport with a maximum pushback rate of 15 aircraft per 15 minutes.

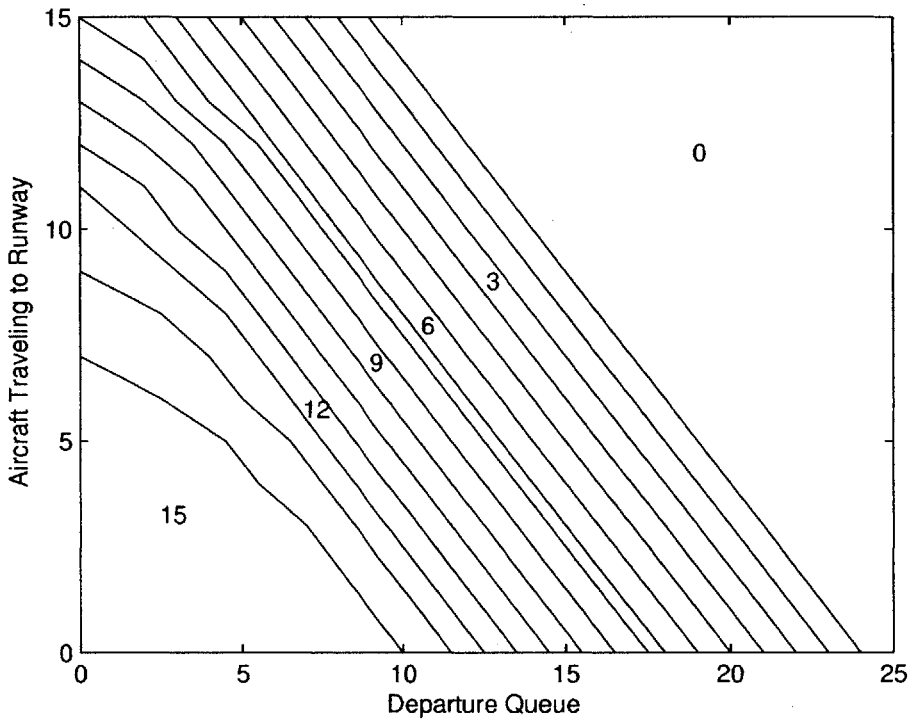

Figure 2-5: Parametric solution of pushback rate with observed values of aircraft taxiing and departure queue length for a 15-minute time window. Each line is the optimal pushback rate for the given state, increasing from 0 to 15 from right to left.

Examining Figure 2-5 provides an intuitive understanding of the underlying mathematics of the dynamic programming algorithm. For a low departure queue Q_0 and departures taxiing R_0, the airport surface is relatively empty and the dynamic programming recommends a maximum pushback rate. For high Q_0 and low R_0, the pushback rate decreases slower with increasing Q_0 when compared to increasing R_0 for high R_0 and low Q_0. This makes sense because, with high Q_0 and low R_0, the queue will likely diminish by the time the recommended pushback rate reaches the runway queue. If R_0 is high and Q_0 is low, the aircraft taxiing to the queue will replenish the queue before the recommended pushback rate reaches the runway. Therefore, the pushback rate does not need to be as high in this circumstance. This is reflected in both the slope and nonlinear characteristics of each parametric solution in Figure 2-5. The intuitive nature of the solution reinforces the methodology of the algorithm. Also, the solution can be summarized in a clean figure that can be used in implementation because it contains all possible evolutions of the airport surface, given R_0 and Q_0. Because the solution is understandable and general, the complexities of the dynamic programming algorithm are masked, making potential implementation possible.

With the pushback rate found, the dynamic programming policy behaves identically to the N-control policy. Equally-spaced pushback slots form to allow for departures to be held at their gates. Again, this increases the likelihood of gate conflicts, and the dynamic programming policy solves this problem in the same manner as the N-control policy. The two policies do differ in execution. While N-control is only in effect during times of congestion $N_{cur} > N_{ctrl}$, the dynamic programming policy is always in effect. Because dynamic programming does not rely on a defined airport congestion threshold, the algorithm can always minimize the cost function, even if the airport is not congested. This difference is evident in the resulting benefits of the two policies as will be seen in the coming chapters.

2.3 Summary

This chapter introduces, derives, and explains two PRC policies, N-control and dynamic programming. N-control aims to maintain departing traffic around an acceptable value N_{ctrl} based on empirical analysis of the relationship between departure throughput and departure traffic. Dynamic programming seeks to minimize a cost function that penalizes both long queues and runway starvation by calculating the probability of the airport being in a certain state at some future time. Each policy is thoroughly explained and derived so as to illustrate clearly the assumptions and methodology used to control departure pushbacks. Figure 2-6 provides an illustration of airport surface management with PRC policies.

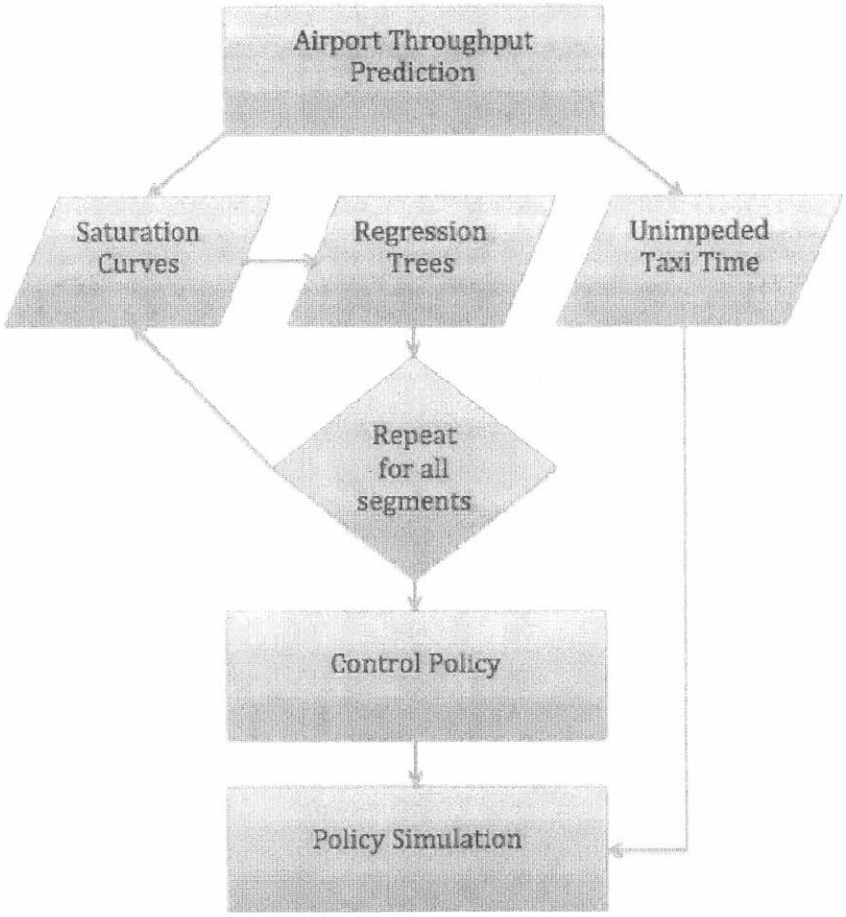

Figure 2-6: Flow chart for airport surface management with PRC policies.

Chapter 3

Variation of Policy Parameters

The PRC policies, N-control and dynamic programming, generate a pushback rate for departures valid for a given time window. Historically, this time window has been set to 15 minutes, but varying this time window can lead to advantages and disadvantages. Also, a pushback rate can be calculated for earlier time windows in the future by changing the time horizon of the policy. Changing the time window or time horizon allows an airport or airline to tailor the PRC policy to specific needs and requirements. Consequently, airports and airlines must understand how varying the policy parameters affects the performance of PRC policies. This section introduces the tradeoffs that arise from varying time windows and horizons. Finally, the results show how the benefits change for different time windows and horizons so that airports and airlines get an accurate grasp of the effects of varying policy parameters.

Figure 3-1 illustrates the difference between time windows and time horizons.

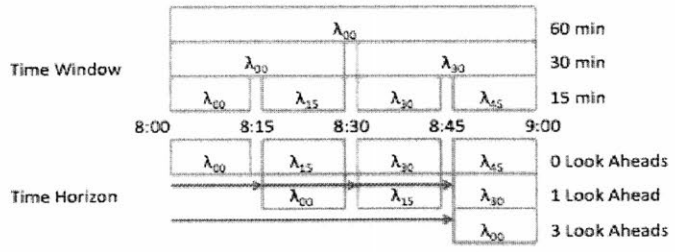

Figure 3-1: Visualization of time windows and time horizons.

First, consider time windows. For 15-minute time windows, the first row above the times in the 8:00 hour shows that four pushback rates govern four 15-minute time windows. These pushback rates are calculated at the beginning of each time window. λ_{00} is calculated at 8:00, λ_{15} is calculated at 8:15, and so on. For 30-minute time windows, the second row above the times in the 8:00 hour shows that two pushback rates govern two 30-minute time windows. These pushback rates are calculated at the beginning of each time window. λ_{00} is calculated at 8:00, λ_{30} is calculated at 8:30. For 60-minute time windows, one pushback rate governs the entire hour, so λ_{00} is calculated at 8:00.

Now, consider time horizons. The time horizon of 0 is exactly the same as the 15-minute time window, with four pushback rates governing four time windows, each calculated at the beginning of a respective time window. For a time horizon of 1, the pushback rate for a particular time window is calculated one time window in advance. The second row below the times in the 8:00 hour has arrows that indicate when a particular pushback rate is calculated. The pushback rate λ_{00} for the 8:15 - 8:30 time window is calculated at 8:00, so the arrow begins when the rate is calculated and leads to the time slot for which this rate is valid. The number of the time horizon indicates the number of time windows in advance that a pushback rate is calculated. Therefore, the time horizon of 3 calculates a pushback rate three time windows in advance. The pushback rate λ_{00} for the 8:45 - 9:00 time window is calculated at 8:00, as shown in the third row below the times in the 8:00 hour. The arrow leads from the time the rate is calculated (8:00) to the time that the rate is valid (8:45).

3.1 Input Data

The data required for the simulation must be pulled from multiple sources. The ASPM dataset provides flight specific metrics such as pushback time, wheels-off time, and wheels-on time. While extremely valuable, the ASPM dataset does not contain a critical piece of information: the gate and terminal assignments of each flight. Gate and terminal assignments allow for the calculation of unimpeded taxi-out time and

allow the policy to monitor gate conflicts. Flightstats.com has this data for each flight, so the PRC policies require the gate and terminal assignments to be integrated with the rest of the ASPM data. The last dataset contains the weather data, RAPT, described previously.

Each simulation contains both a baseline case and a metering case. The baseline case simulates the airport operations by releasing departures from their gates on a FCFS basis based on scheduled departure times. The metering case simulates the airport operations using a PRC policy. The benefits of the policy include the taxi-out time reduction, which is the difference between the taxi-out times in the baseline case and metering case. Taxi-out time reduction contrasts with gateholding time, which is the length of time an aircraft is held at a gate beyond the scheduled departure time due to the PRC policy. Gateholding time is not strictly a cost because aircraft still belong to the virtual queue with engines off. However, occupying the gate causes more gate conflicts, while extended gateholding times can lead to passenger discomfort.

3.2 Time Window

3.2.1 Assessment of the Time Window Length

PRC policies calculate a pushback rate for departing aircraft that is valid for a certain time window. This time window impacts some performance characteristics of the policy, as Simaiakis et al.[23] note briefly. The length of this time window is a tradeoff between: accuracy, ease of implementation, and value added to operators (airlines and controllers). Depending on the main priority of the PRC policy, the time window should be chosen to achieve a certain goal, which may come at the cost of other performance characteristics.

For accuracy, the policies become less accurate as the length of the time window increases. Because the PRC policies calculate the pushback rate at the beginning of the time window based on the state of the airport surface, a longer time window means that the pushback rate is valid for a longer period of time. As time gets farther

away from the beginning of the window, the state of the airport surface changes. This could cause the pushback rate for a time window to be different from the optimal value as time increases from the beginning of the window. Also, the quality of the input data affects the accuracy of the PRC policy. Arrival rates can be predicted somewhat accurately for a 15-minute time window, but this accuracy diminishes for longer time windows. Because of this, the pushback rate could be calculated based on incorrect data.

For implementation, the policies affect the workload of the traffic controller. For shorter time windows, the controller must keep updating the pushback rate more frequently. This involves gathering the input data and the calculation of the rate based on the PRC policy used. This workload decreases with the increase in the length of the time window because the pushback rate is valid for a longer period of time. However, as explained above, this decrease in workload comes at the cost of decreased accuracy.

For the value added by the PRC policy, one must consider both the airlines and the controllers. In terms of airlines, the time window must be long enough for the airlines to plan their operations. However, the benefits of the PRC policy decrease as the time window gets longer due to the decrease in accuracy. Airlines must choose the right time window length that allows for smooth operations planning with sufficient benefits. For controllers, the workload variation has already been considered. In addition, the benefits to traffic control must be substantial enough to justify any increase in workload.

3.2.2 N-Control

In addition to the usual 15-minute time window length simulation, simulations with 30-minute and 60-minute time windows allow for the exploration of the effects of lengthening the time window. Time windows shorter than 15 minutes would likely overload controllers with frequent PRC policy updates, as discussed above. The results of these three simulations with different time window lengths should reflect the tradeoffs of varying time windows.

The tools and data must be reconfigured for the longer time window simulations. The saturation curves change to account for the increase in departure throughput due to extending the time windows. Also, N_{adj} is the metric for departure surface traffic used for the saturation curves. Because the saturation curves change, so must the regression trees. Each tree is rebuilt, accounting for the larger arrival rates and predicted departure throughput. Unimpeded taxi-out time remains the same because the time window has no effect on this metric. With all of the capacity and simulation tools reconfigured, Equation 2.9 still calculates the departure pushback rate and the simulation proceeds exactly as before, except with longer time windows.

The results of the simulations for LaGuardia Airport for July-August 2013 can be seen in Table 3.1. T_m is the taxi-out time reduction in minutes for the metering case compared to the baseline case, while T_p is the percent taxi-out time reduction for the metering case compared to the baseline case. The reduction results are compared to the baseline case in which aircraft push back from their gates as soon as they are ready.

Table 3.1 clearly shows that as the length of the time window increases, the benefits decrease in terms of taxi-out time reduction. The 30-minute time window simulation has 78.6% of the taxi-out time reduction of the 15-minute time window simulation. The 60-minute time window simulation has 54.7% of the taxi-out time reduction of the 15-minute time window simulation. However, each policy has considerable benefits.

In terms of policy fairness, an airport surface management algorithm must treat all airlines equally, meaning that no airline can get a vastly disproportionate share of the benefits. The N-control policy demonstrates fairness by the nearly one-to-one ratio of the percentage share in taxi-out time reduction and the percentage share of departures, or market share. The benefits almost directly correlate with market share. This means that the N-control policy fairly distributes the benefits based on the size of an airline's airport operations. This fairness also exists in the 30-minute and 60-minute simulations, so varying the time window length does not adversely affect policy fairness.

Also, each policy maintains roughly a one-to-one ratio of taxi-out time reduction

to gateholding time. This indicates that a minute of gateholding time will correspond to a minute of taxi-out time reduction during periods of congestion. This relationship is important because airlines do not need to invest a large amount of gateholding to realize a benefit in taxi-out time reduction. The one-to-one ratio here indicates that the N-control policy directly rewards investments of gateholding time with an equal benefit of taxi-out time reduction. Again, this ratio remains for the 30-minute and 60-minute simulations.

Table 3.1: Simulation for each time window policy, separated by airline.

Airline	Policy (min)	T_m	T_p	Reduction % Share	Gateholding Time (min)	Gateholding % Share	Market Share
1	15	24,981	6.6%	39.2%	25,091	39.3%	38.0%
	30	19,693	4.8%	39.3%	19,755	39.4%	
	60	13,720	3.3%	39.3%	13,551	38.9%	
2	15	11,858	6.6%	18.6%	11,831	18.6%	19.8%
	30	9,359	4.8%	18.7%	9,260	18.5%	
	60	6,459	3.2%	18.5%	6,543	18.8%	
3	15	5,995	7.1%	9.4%	6,091	9.6%	8.2%
	30	4,710	5.1%	9.4%	4,688	9.4%	
	60	3,368	3.6%	9.7%	3,429	9.8%	
4	15	2,403	6.4%	3.8%	2,401	3.8%	4.3%
	30	2,015	5.0%	4.0%	2,007	4.0%	
	60	1,326	3.2%	3.8%	1,319	3.8%	
5	15	1,057	6.1%	1.7%	1,067	1.7%	2.0%
	30	837	4.5%	1.7%	876	1.8%	
	60	654	3.4%	1.9%	665	1.9%	
6	15	2,505	5.9%	3.9%	2,463	3.9%	5.1%
	30	2,040	4.4%	4.1%	2,142	4.3%	
	60	1,526	3.2%	4.4%	1,492	4.3%	
7	15	1,432	5.7%	2.3%	1,407	2.2%	3.1%
	30	1,176	4.2%	2.4%	1,139	2.3%	
	60	853	3.0%	2.5%	857	2.5%	
8	15	13,542	7.1%	21.2%	13,424	21.1%	19.6%
	30	10,315	5.0%	20.6%	10,279	20.5%	
	60	6,968	3.3%	20.0%	7,018	20.1%	

3.2.3 Dynamic Programming

The dynamic programming policy can also extend beyond the 15-minute time window simulation to include 30-minute and 60-minute time windows. The Δ in the dynamic programming equations simply changes to match the length of the chosen time window. Then, the Chapman-Kolmogorov equations must be numerically integrated throughout the new time window. With the data and regression trees adjusted for the different time window length, the dynamic programming algorithm proceeds as before. Table 3.2 shows the results of the different time window length simulations.

Table 3.2: Dynamic programming time window simulation results, separated by airline.

Airline	Policy (min)	T_m	T_p	Reduction % Share	Gateholding Time (min)	Gateholding % Share	Market Share
1	15	85,581	22.7%	39.6%	89,032	39.5%	38.0%
	30	53,806	13.1%	39.6%	56,193	39.5%	
	60	38,228	9.1%	40.8%	43,250	40.3%	
2	15	39,788	22.3%	18.4%	41,705	18.5%	19.8%
	30	25,161	12.9%	18.5%	26,315	18.5%	
	60	16,829	8.4%	18.0%	19,620	18.3%	
3	15	20,021	23.6%	9.3%	20,870	9.3%	8.2%
	30	12,699	13.8%	9.3%	13,251	9.3%	
	60	9,086	9.6%	9.7%	10,183	9.5%	
4	15	8,023	21.4%	3.7%	8,377	3.7%	4.3%
	30	4,985	12.3%	3.7%	5,297	3.7%	
	60	3,332	8.0%	3.6%	3,716	3.5%	
5	15	3,797	22.0%	1.8%	3,916	1.7%	2.0%
	30	2,463	13.2%	1.8%	2,534	1.8%	
	60	1,530	8.0%	1.6%	1,835	1.7%	
6	15	8,370	19.6%	3.9%	8,751	3.9%	5.1%
	30	5,284	11.4%	3.9%	5,525	3.9%	
	60	3,350	7.0%	3.6%	3,892	3.6%	
7	15	4,342	17.1%	2.0%	4,550	2.0%	3.1%
	30	2,659	9.6%	2.0%	2,782	2.0%	
	60	1,482	5.2%	1.6%	1,817	1.7%	
8	15	46,198	24.0%	21.4%	47,989	21.3%	19.6%
	30	28,975	13.9%	21.3%	30,282	21.3%	
	60	19,824	9.3%	21.2%	22,947	21.4%	

The dynamic program results in Table 3.2 show that the policy benefits degrade

as the length of the time window increases, following the same trend as the N-control policy results. The 30-minute time window simulation has 62.9% of the taxi-out time reduction of the 15-minute time window simulation. The 60-minute time window simulation has 43.3% of the taxi-out time reduction of the 15-minute time window simulation. The dynamic programming simulations maintain policy fairness by comparing the airline shares of taxi-out time reduction and market share. The one-to-one ratio of taxi-out time reduction and gateholding time is also present in the dynamic programming simulations.

Comparing the results of the N-control policy and the dynamic programming policy simulations highlights some of the differences between the algorithms. The absolute benefits of the dynamic programming policy are much greater than the benefits of the N-control policy. However, this does not immediately indicate that dynamic programming is the better policy. Because the dynamic programming algorithm relies on an infinite horizon solution, the dynamic programming algorithm controls departures at all times, not just during times of congestion like the N-control policy. Therefore, the dynamic programming algorithm meters many more flights than the N-control algorithm. This explains the rather large difference in policy benefits.

To get a better sense of the difference in performance of the two PRC policies, Table 3.3 shows the taxi-out time reduction per metered flight ($\hat{T}_m$) in minutes, as well as the percentage of all flights metered.

Table 3.3: PRC policy variable time window performance comparison (N: N-control, DP: dynamic programming).

Policy	$\hat{T}_m$	% of flights metered	Policy	$\hat{T}_m$	% of flights metered
N (15 min)	10.3	20.3%	DP (15 min)	12.6	56.0%
N (30 min)	10.4	15.8%	DP (30 min)	10.6	42.0%
N (60 min)	9.8	11.6%	DP (60 min)	7.8	39.5%

Table 3.3 reveals that the dynamic programming policy meters many more flights than the N-control policy. For the 15-minute time window, most of the difference in policy performance stems from the additional metering of the dynamic programming policy, although the average taxi-out reduction of the dynamic programming policy is

2 minutes greater than the N-control policy. For the 30-minute time window, almost all of the difference in policy performance is due to the discrepancy in number of flights metered. For the 60-minute time window, the dynamic programming policy actually has a smaller taxi-out time reduction than the N-control policy. However, the additional metering of the dynamic programming policy results in a greater total taxi-out reduction. The N-control policy maintains a fairly consistent $\hat{T}_m$ with increasing time window lengths. The $\hat{T}_m$ for N-control varies by no more than half of a minute. For dynamic programming, the $\hat{T}_m$ decreases by nearly 5 minutes from the 15-minute time window simulation to the 60-minute time window simulation. The N-control policy has more stable policy benefits per metered flight across different time window lengths compared to the dynamic programming policy.

The differences between the N-control and dynamic programming results mirror the differences between the two PRC policies. N-control only meters during times of congestion, while dynamic programming always seeks to minimize the cost function associated with queuing and runway utilization. The results should also communicate to airports and airlines that implementing a dynamic programming algorithm will result in greater benefits, but at the cost of disrupting airport operations with many metered flights. On the other hand, implementing an N-control algorithm meters fewer flights than dynamic programming, but the benefits of the policy also decrease. While dynamic programming might be an ideal solution to a new airport to manage airport surface operations, N-control may be easier to implement at busier airports due to the difference in metering frequency. Airports and airlines must consider all possible effects of implementing each PRC policy, including not only the benefits but also the impact on airport operations.

Portraying dynamic programming as possibly disruptive may be a bit unfair. The dynamic programming algorithm has the goal of ensuring the queue length is not excessive so as to prevent unnecessary fuel burn. The N-control algorithm only monitors the total number of departing aircraft on the airport surface. While the N-control algorithm may allow a long queue with few departures taxiing to the queue, the dynamic programming algorithm penalizes such a situation. By accounting for the

difference between departures taxiing and departures in the queue, dynamic programming has more detailed information than the N-control algorithm uses. The greater detail gives the dynamic programming more situations in which to control the rate of departure pushback.

3.3 Time Horizon

3.3.1 Assessment of the Time Horizon Length

The time horizon length variation has the same tradeoffs associated with the time window length variation. To reiterate, the time horizon is the number of time windows before a given time window that a departure rate is calculated. The above analysis for time window lengths has one departure pushback rate valid for the entire time window. At the end of a time window, a new departure pushback rate is calculated and the policy resets. The time horizon in this instance is zero because the PRC policy does not look ahead to other time windows to calculate other departure pushback rates. For time horizons greater than zero, a unique departure pushback rate is calculated for subsequent time windows. For a time horizon of 1, the policy calculates a departure rate based on the expected state of the variables needed for the time window following the current time window.

For the N-control policy, the expected change in N_{cur} follows directly from an application of conservation of aircraft on the airport surface. If N'_{cur} is the departure surface traffic one time window into the future, then

$$N'_{cur} = N_{cur} - T_p + R, \tag{3.1}$$

where the variables on the right-hand side of Equation 3.1 are for the current time window. The weather and arrival rate can also be predicted with acceptable accuracy multiple time windows into the future, so predicted throughput can also be found. The accuracy of arrival rate prediction will be the subject of an analysis in the next chapter. Therefore, all of the elements of Equation 2.9 are known (or estimated),

resulting in a departure pushback rate for a time window in the time horizon. With the departure pushback rate, the N-control simulation proceeds as usual.

For the dynamic programming policy, both the departures taxiing R_0 and stages-of-work in the departure queue Q_0 at the beginning of a time window change throughout a time horizon. The changes in these variables become straightforward under the assumption that all aircraft taxiing at the beginning of the previous time window reach the departure queue by the beginning of the next time window. If Q'_0 corresponds to the stages-of-work one time window into the future, then

$$Q'_0 = (\frac{Q_0}{k} - T_p + R_0) \times k, \tag{3.2}$$

where the variables on the right-hand side of Equation 3.2 are for the current time window. The departure queue grows by an amount equal to the departures taxiing at the beginning of the previous time window. The departure queue shrinks by the predicted throughput for the previous time window. Again, a conservation of aircraft in the departure queue perspective explains Equation 3.2.

The expected change in R_0 is more straightforward. If the number of aircraft expecting to push back in a time window exceeds the departure pushback rate, the departures taxiing at the beginning of the next time window simply equal the departure pushback rates. If the number of aircraft expecting to push back in a time window does not exceed the departure pushback rate, the departures taxiing at the beginning of the next time window are the number of aircraft expecting to push back in the previous time window. The equations for R'_0, the departures taxiing one time window into the future, are then

$$R'_0 = R \text{ if } R \leq D_r, \tag{3.3}$$

$$R'_0 = D_r \text{ if } R > D_r, \tag{3.4}$$

where R is the pushback rate of the previous time window and D_r is the number of departures ready to push back in the previous time window. With Q'_0 and R'_0

estimated, the dynamic programming simulation proceeds as usual.

3.3.2 N-control

In order to mirror the variable time window analysis, time horizons of length 0, 1, and 3 look ahead 0 minutes, 15 minutes, and 45 minutes into the future, respectively. The time window length in this analysis remains a constant of 15 minutes. This allows a direct comparison with the time window analysis to examine the similarities and differences.

Unlike the variable time window analysis, the tools and data do not need to be altered for the simulation. Because the time windows remain 15 minutes long, the saturation curves and regression trees for the 15-minute time window remain valid. The only alteration of the simulation arises when calculating the expected change of the departure surface traffic N'_{cur}. Inserting this calculation, described by Equation 3.2, allows for the simulation to operate using time horizons. The expected N'_{cur} is used to calculate the pushback rate for a time window of 0, 15, or 45 minutes instead of observing the surface traffic, as is done at the beginning of a time window. The rest of the simulation remains the same.

The results of the time horizon simulations for LGA during the same dates (July - August 2013) can be seen in Table 3.4. Again, the reduction results are compared to the baseline case in which aircraft push back from their gates as soon as they are ready (FCFS).

Table 3.4 clearly shows that as the length of the time horizon increases, the benefits decrease in terms of taxi-out time reduction. However, the decrease in benefits is less than the corresponding decrease in benefits in the time window analysis. The time horizon of 1 simulation has 80.1% of the taxi-out time reduction of the time horizon of 0 simulation. The time horizon of 3 simulation has 53.0% of the taxi-out time reduction of the time horizon of 0 simulation. Each policy maintains considerable benefits.

In extending the time horizon length, the N-control policy maintains both fairness and the one-to-one ratio of taxi-out time reduction to gateholding time. Because the

simulations with longer time horizons exhibit the same qualities as the simulations with longer time windows, the longer time horizons appear to be more favorable than the longer time windows in terms of taxi-out time reduction benefits. Considering the differences between time horizons and time windows, the difference in benefits makes sense. Extending the time window means that one pushback rate is valid for a longer period of time. Also, predictions of weather and arrival rate must also project further into the future. Conversely, extending the time horizon means one pushback rate for each time window in the time horizon, calculated using conservation of departure aircraft on the airport surface. The same problem with weather and arrival rate persists. Because the time horizon analysis updates the departure traffic for each time window, the pushback rate is calculated based on more accurate information compared to the pushback rate for a long time window.

Table 3.4: Simulation for each time horizon policy, separated by airline.

Airline	Horizon	T_m	T_p	Reduction % Share	Gateholding Time (min)	Gateholding % Share	Market Share
1	0	24,981	6.6%	39.2%	25,091	39.3%	38.0%
	1	20,008	5.3%	39.2%	20,137	39.4%	
	3	13,374	3.6%	39.6%	13,387	39.6%	
2	0	11,858	6.6%	18.6%	11,831	18.6%	19.8%
	1	9,526	5.3%	18.7%	9,481	18.6%	
	3	6,418	3.6%	19.0%	6,436	19.1%	
3	0	5,995	7.1%	9.4%	6,091	9.6%	8.2%
	1	4,836	5.7%	9.5%	4,890	9.6%	
	3	3,201	3.8%	9.5%	3,297	9.8%	
4	0	2,403	6.4%	3.8%	2,401	3.8%	4.3%
	1	1,897	5.1%	3.7%	1,853	3.6%	
	3	1,157	3.1%	3.4%	1,135	3.4%	
5	0	1,057	6.1%	1.7%	1,067	1.7%	2.0%
	1	886	5.1%	1.7%	871	1.7%	
	3	527	3.1%	1.6%	520	1.5%	
6	0	2,505	5.9%	3.9%	2,463	3.9%	5.1%
	1	1,942	4.6%	3.8%	1,915	3.8%	
	3	1,273	3.0%	3.8%	1,245	3.7%	
7	0	1,432	5.7%	2.3%	1,407	2.2%	3.1%
	1	1,152	4.6%	2.3%	1,145	2.2%	
	3	758	3.0%	2.2%	740	2.2%	
8	0	13,542	7.1%	21.2%	13,424	21.1%	19.6%
	1	10,842	5.6%	21.2%	10,798	21.1%	
	3	7,079	3.7%	21.0%	7,027	20.8%	

3.3.3 Dynamic Programming

A time horizon analysis also works with the dynamic programming policy. The analysis also uses the time horizons of 0, 1, and 3, like the N-control simulations. The tools and data also do not have to be updated. Only the expected departures taxiing and stages-of-work in the departure queue must be added into the simulations, and this is done in the manner described above. The dynamic programming policy then uses those inputs and proceeds as usual. Table 3.5 shows the results of the variable time horizon simulations.

Table 3.5: Dynamic programming time horizon simulation results, separated by airline.

Airline	Horizon	T_m	T_p	Reduction % Share	Gateholding Time (min)	Gateholding % Share	Market Share
1	0	85,581	22.7%	39.6%	89,032	39.5%	38.0%
	1	75,866	20.1%	39.7%	79,501	39.7%	
	3	59,217	15.7%	39.9%	61,551	39.9%	
2	0	39,788	22.3%	18.4%	41,705	18.5%	19.8%
	1	34,872	19.5%	18.3%	36,781	18.4%	
	3	26,922	15.1%	18.1%	28,125	18.2%	
3	0	20,021	23.6%	9.3%	20,870	9.3%	8.2%
	1	17,917	21.1%	9.4%	18,793	9.4%	
	3	14,275	16.8%	9.6%	14,822	9.6%	
4	0	8,023	21.4%	3.7%	8,377	3.7%	4.3%
	1	6,947	18.6%	3.6%	7,323	3.7%	
	3	5,253	14.0%	3.5%	5,531	3.6%	
5	0	3,797	22.0%	1.8%	3,916	1.7%	2.0%
	1	3,347	19.4%	1.8%	3,479	1.7%	
	3	2,486	14.4%	1.7%	2,567	1.7%	
6	0	8,370	19.6%	3.9%	8,751	3.9%	5.1%
	1	7,451	17.5%	3.9%	7,776	3.9%	
	3	5,814	13.6%	3.9%	6,020	3.9%	
7	0	4,342	17.1%	2.0%	4,550	2.0%	3.1%
	1	3,705	14.6%	1.9%	3,905	2.0%	
	3	2,820	11.1%	1.9%	2,932	1.9%	
8	0	46,198	24.0%	21.4%	47,989	21.3%	19.6%
	1	40,968	21.3%	21.4%	42,796	21.4%	
	3	31,593	16.4%	21.3%	32,728	21.2%	

Like the variable time window analysis with dynamic programming, the policy benefits decrease as the time horizon increases. Table 3.5 shows that the total benefits of the policy with a time horizon of 1 are 88.4% of the total benefits of the policy with a time horizon of 0. Also, the total benefits of the policy with a time horizon of 3 are 68.7% of the total benefits of the policy with a time horizon of 0. Comparing these results to the time window analysis with dynamic programming, extending the time horizon maintains much more of the benefits than extending the length of the time window. The fairness quality of the PRC policies carries over here, as well as the near one-to-one ratio of taxi-out time reduction to gateholding time.

The performance per flight across different time horizons is shown in Table 3.6,

Table 3.6 shows the taxi-out time reduction per metered flight ($\hat{T}_m$) in minutes, as well as the percentage of all flights metered.

Table 3.6: PRC policy variable time horizon performance comparison (N: N-control, DP: dynamic programming).

Policy	$\hat{T}_m$	% of flights metered	Policy	$\hat{T}_m$	% of flights metered
N (0 time horizon)	10.3	20.3%	DP (0 time horizon)	12.6	56.0%
N (1 time horizon)	9.4	17.8%	DP (1 time horizon)	11.9	52.6%
N (3 time horizon)	7.6	14.9%	DP (3 time horizon)	10.3	47.3%

The results in Table 3.6 reveal the reasons behind the longer time horizons maintaining more policy benefits than longer time windows. The taxi-out time reduction per metered flight of longer time horizons decreases nearly 3 minutes for the N-control policy and more than 2 minutes for the dynamic programming policy. Also, the percentage of flights metered does not decrease drastically as the time horizon increases. This was not the case for the variable time window analysis. However, this difference is due to the fact that a long time window simulation may miss times of congestion, but longer time horizon simulations estimate the way in which the surface traffic will evolve. This results in longer time horizon simulations more likely to identify times of congestion within a time horizon. Longer time window simulations do not identify times of congestion within a time window that was not congested at the start of the time window.

3.4 Gate Conflicts

This section uses the N-control policy to explain the issue of gate conflicts. Because PRC policies hold aircraft at their gates past their scheduled departure time, the possibility of increased gate conflicts arises. Gate conflicts occur during normal airport operations, but PRC policies increase the frequency of gate conflicts. This increase can be seen in Figure 3-2. This figure shows the average number of gate conflicts per hour for each airline, with and without metering. Figure 3-3 shows the average number of gate conflicts per day for each airline by day of the week, with and without metering.

While metering clearly increases the number of gate conflicts, the increase is not drastic. Figure 3-3 shows that the number of daily gate conflicts increases anywhere from 0 to 6 per day, depending on the airline and day of the week. The PRC policy simulations handles gate conflicts in the following manner. If a departure occupies a gate when an arrival, scheduled for that gate, lands, the departure immediately pushes back.

Notice the nature of the timing of the gate conflicts in Figure 3-2. For most airlines, a spike in gate conflicts occurs during the morning rush (10 AM - 12 PM) and/or afternoon rush (5 PM - 8 PM). Because these rushes correspond to an increase in airport surface operations, the increase in gate conflicts during these times makes sense. The timing of gate conflicts in Figure 3-3 is also intuitive. Airports usually have less traffic on the weekends, and the gate conflicts at LGA also decrease for nearly each airline on these days compared to the rest of the week. Also, the increase in gate conflicts due to metering is less severe during the weekends.

The increase in gate conflicts from metering also affects airlines proportionally in terms of share of gate conflicts without metering. Examining Figures 3-2 and 3-3 reveals this pattern. Delta Airlines and US Airways have the most gate conflicts in the simulation without metering. The other airlines with low numbers of gate conflicts see less gate conflicts caused by metering. By not penalizing airlines disproportionately in terms of gate conflicts, the N-control policy maintains fairness in another sense.

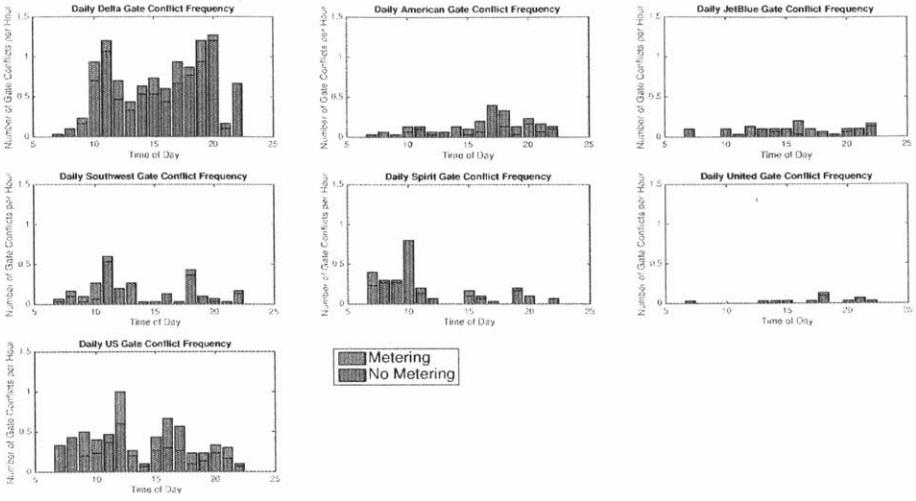

Figure 3-2: Average gate conflict frequency per hour for each airline, with and without metering.

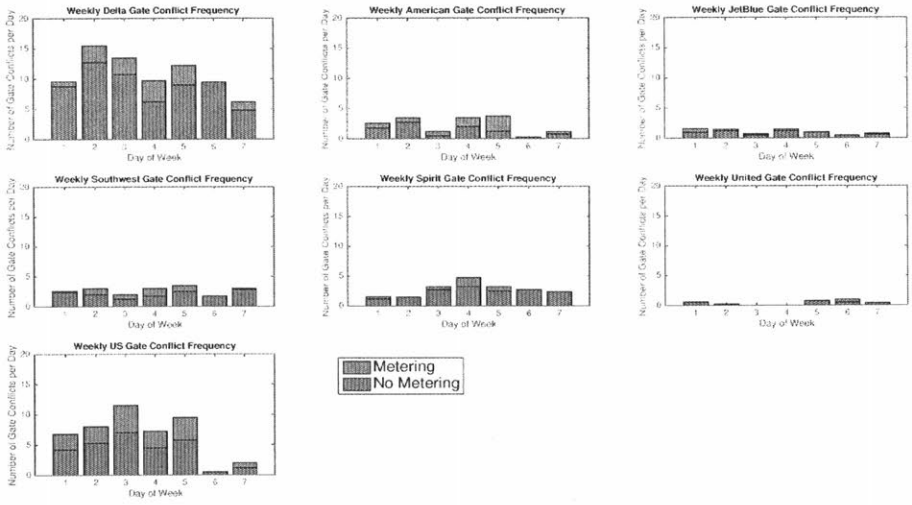

Figure 3-3: Average gate conflict frequency per day for each airline, with and without metering. 1 corresponds to Monday and 7 corresponds to Sunday.

3.5 Conclusions

This chapter explores the policy parameter variation of time windows and time horizons associated with PRC policies. The time window is the length of time for which a departure pushback rate is valid, while the time horizon is the number of time windows into the future for which a departure pushback rate is calculated in advance. The tradeoffs associated with the length of both time windows and time horizons are policy accuracy, ease of implementation, and value added to operators. Extending time windows or horizons reduces operator workload resulting from the policy, but policy accuracy and value also decrease. Shortening time windows or horizons increases accuracy and value, but operator workload soars. For both N-control and dynamic programming policies, variable time window and time horizon analyses simulate the policies at LGA. The results indicate that lengthening the time window decreases policy benefits more drastically than lengthening the time horizon. These results reflect that longer time horizon simulations monitor departure surface traffic more dynamically than longer time window simulations.

Chapter 4

Operational Uncertainty

Chapter 2 introduces and derives the PRC policies of N-control and dynamic programming. Chapter 3 simulates airport operations at LGA using both of the PRC policies while also exploring the variation of policy parameters, the time window and time horizon. In addition to this, operational uncertainty also affects the performance of PRC policies. As Chapter 2 describes, the PRC policies require input data to calculate the departure pushback rate. The regression trees need both the arrival rate and RAPT for the next time window to predict the departure throughput. The arrival rate and weather for a period in the future are both uncertain. Also, the departure schedule is uncertain as many factors can cause a departure to be ready for pushback before or after the scheduled departure time. In order to be considered for acutal implementation at an airport, PRC policies must maintain effectiveness when accounting for uncertainty. This chapter considers uncertainty in the departure schedule and arrival rate for N-control and dynamic programming. Also, building off of the policy parameter variation analysis, this chapter combines both policy parameter variation and operational uncertainty to more realistically simulate airport operations.

4.1 Departure Schedule

Any person who has flown commercially would likely testify to the uncertainty in the scheduled departure time of a flight. The departure pushback time of a flight may change for myriad reasons. Delays occur due to weather, maintenance, congestion (surface and air), boarding, and ground crew availability, to name a few causes. While frustrating, these issues arise while operating airports and airlines on a national and global scale. Also, airlines strive to "turn around" an aircraft quickly, which means deplaning an arrival aircraft and preparing it for departure as fast as possible. These efforts sometimes result in a departure aircraft ready to push back before the scheduled departure time. Both N-control and dynamic programming rely on an adequate supply of departures ready for pushback in a given time window. If the available departures are less than the pushback rate, the departure surface traffic may fall below acceptable levels. If there are many available departures during a time of low congestion, those departures may cause congestion in the future. As such, PRC policies need to handle a variable departure schedule.

To explore the performance of the PRC policies with a variable departure schedule, the simulations must undergo some adjustments. Perturbations can be added to the scheduled departure times to approximate the small delays incurred by a flight. The following analysis does not account for larger delays because PRC policies would likely not operate during times of large delay. For example, during severe weather, other traffic management programs may be in effect. While large delays can occur for a small number of flights, those situations closely approximate a normal departure schedule.

The simulation approximates the perturbations to the flight schedule by assuming that the perturbations are drawn from a normal distribution with a mean of the scheduled departure time and a standard deviation of 3.5 minutes. The assumption of a standard deviation of 3.5 minutes ensures that two standard deviations from the scheduled departure time roughly encompasses a 15-minute time window. The perturbation time probability distribution is shown in Figure 4-1.

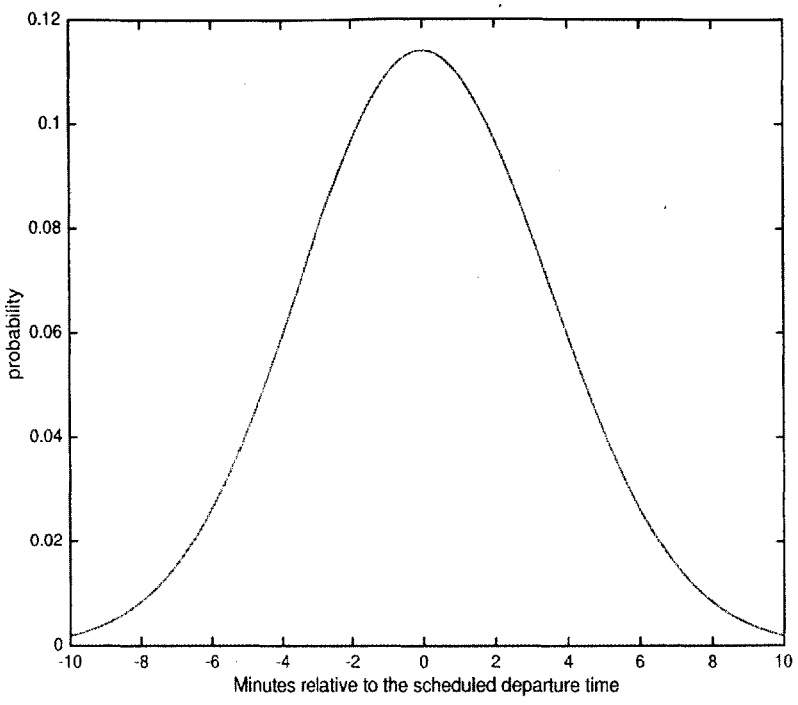

Figure 4-1: Probability distribution of the departure time perturbations relative to the scheduled departure times.

For each flight, a perturbation time is randomly drawn from the probability distribution in Figure 4-1 and added to the original departure time. The equation for the new departure time of a flight t^* is

$$t^* = t + t_p, \tag{4.1}$$

where t is the scheduled departure time of the flight and t_p is the perturbation time. Equation 4.1 updates the scheduled departure time for each flight in a day. Then, the simulation of airport operations with the PRC policies proceeds as usual.

Perturbing the departure schedule only results in one different schedule. The results of the simulation with the original schedule can be compared to the results of the simulation with the perturbed schedule. However, because the perturbed schedule is subject to random sampling, the results of the simulation with the perturbed schedule

are also subject to the random sampling. As such, the variable departure schedule uses a Monte Carlo method to get a better sense of the results from perturbing the schedule. The Monte Carlo method simply runs the 2-month LGA simulations with a PRC policy 50 times, each time with a different perturbed schedule. The results of each simulation with a perturbed schedule can then be viewed and considered together relative to the results of the simulation with the original schedule.

4.1.1 N-control Results

Figures 4-2 and 4-3 contain the results of the N-control Monte Carlo method simulations with the schedule perturbations for a 15-minute time window and time horizon of 0. Figure 4-2 shows the total taxi-out benefits for each of the 50 simulations from the Monte Carlo method. Figure 4-3 shows the percent taxi-out reduction, separated by airline, for each of the 50 simulations from the Monte Carlo method. Figure 4-2 provides an illustration of the total policy benefits, while 4-3 reveals the distribution of savings a given airline can expect due to a variable departure schedule.

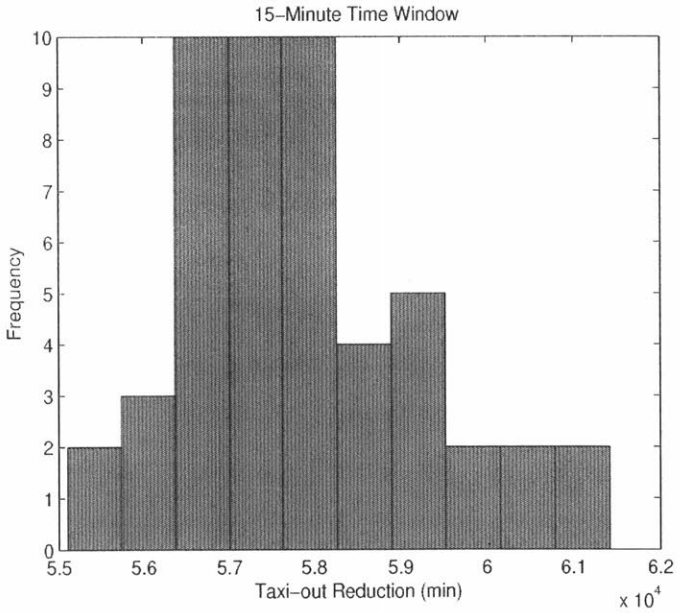

Figure 4-2: Frequency of total taxi-out reduction benefits for the N-control 50 Monte Carlo method simulations.

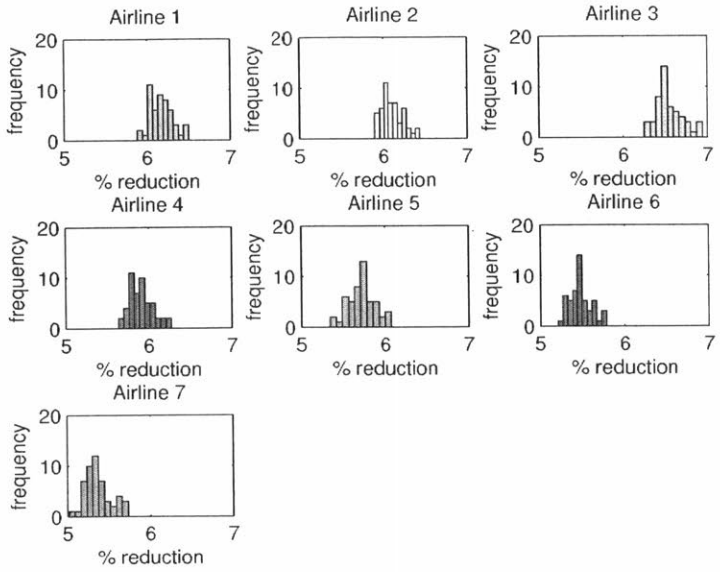

Figure 4-3: Percent taxi-out reduction by airline for the N-control 50 Monte Carlo method simulations.

Recall that the total taxi-out reduction for the 15-minute time window, 0 time horizon N-control simulations for LGA for Summer 2013 is 63,773 minutes. The mean total taxi-out reduction from the 50 Monte Carlo method simulations is 57,826 minutes. This indicates that the departure schedule uncertainty reduces the N-control policy benefits to 90.7% of the policy benefits in the case with no uncertainty. Also, Figure 4-3 shows that all airlines can expect a percent taxi-out reduction between 5% and 7% compared to their operations without metering, even with a variable departure schedule. Because the total and airline benefits remain comparable to the benefits in the case with no uncertainty, the N-control policy performs well when accounting for a variable departure schedule.

4.1.2 Dynamic Programming Results

The dynamic programming policy simulations have the same departure schedule as the N-control policy simulations, so the variable departure schedule analysis for dy-

namic programming uses the same departure time perturbation method described by Equation 4.1. Figures 4-4 and 4-5 contain the results of the dynamic programming Monte Carlo method simulations with the schedule perturbations for a 15-minute time window and time horizon of 0. These figures correspond to Figures 4-2 and 4-3 for the N-control variable departure schedule analysis to allow for direct comparison between PRC policies.

Recall that the total taxi-out reduction for the 15-minute time window, 0 time horizon dynamic programming simulations for LGA for Summer 2013 is 216,120 minutes. The mean total taxi-out reduction from the 50 Monte Carlo method simulations is 203,880 minutes. This indicates that the departure schedule uncertainty reduces the dynamic programming policy benefits to 94.3% of the policy benefits in the baseline case. Figure 4-5 shows that all airlines can expect a percent taxi-out reduction comparable to their operations without metering, even with a variable departure schedule. Like the N-control policy, the dynamic programming policy performs well when accounting for a variable departure schedule.

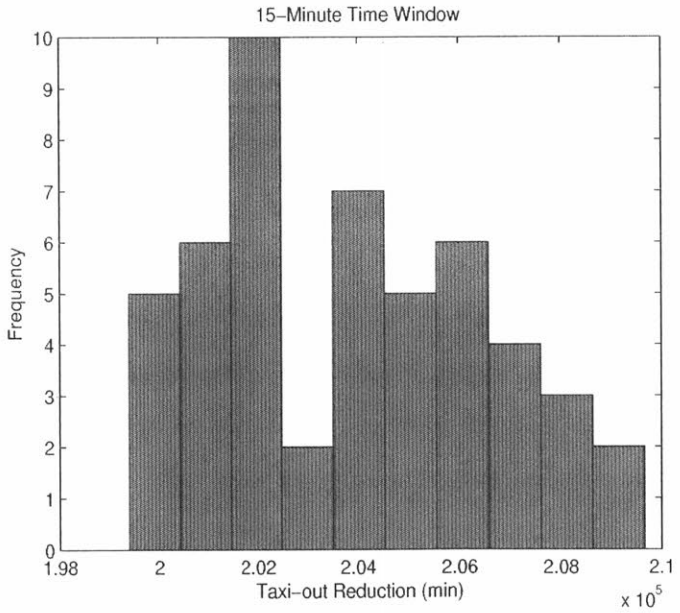

Figure 4-4: Frequency of total taxi-out reduction benefits for the dynamic programming 50 Monte Carlo method simulations.

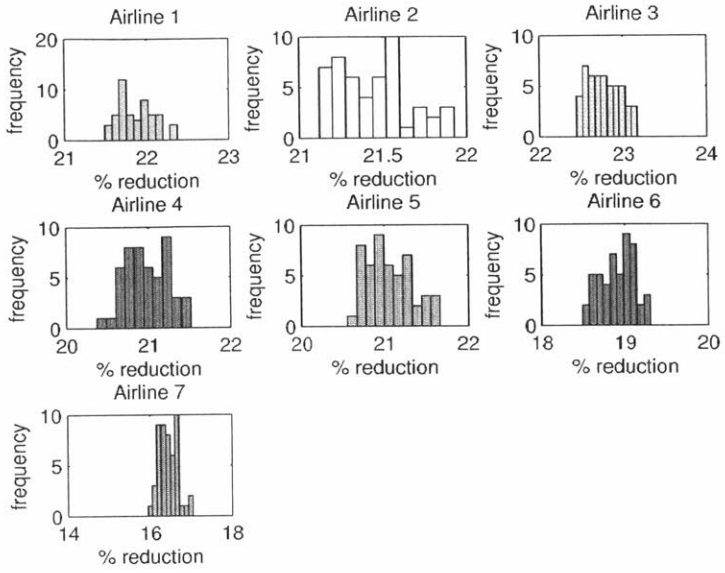

Figure 4-5: Percent taxi-out reduction by airline for the dynamic programming 50 Monte Carlo method simulations.

When accounting for a variable departure schedule, the total taxi-out reduction for both N-control and dynamic programming remains within 10% of the total taxi-out reduction of the simulations with no uncertainty. This bodes well for the possible implmentation of PRC policies. Examining the driving forces behind the comparable performances of the simulation with no uncertainty and a variable departure simulation provides further comprehension of the results. The departure time perturbations simply shuffle the departure schedule, causing some 15-minute windows to have more or fewer departures. This spreads the departure times out more throughout the day when, in reality, many flights may have departure times clustered together. During peak hours, many flights have departure times very close together, causing congestion. Spreading out departure times, even with small purturbations, can slightly decrease this cause of congestion.

While allowing for a variable departure schedule makes the simulations more realistic, weather and arrival predictions remain constant. These variables affect the predicted throughput for a time window, so the PRC policies must perform well with

uncertainty in these variables. To account for arrival rate uncertainty, the following section details an uncertainty analysis with a variable arrival rate.

4.2 Arrival Rate

The number of arrivals landing at an airport in a given time window is uncertain. The scheduled arrival time of an aircraft changes due to many of the reasons that affect the scheduled departure time, including weather and congestion. Instead of scheduled arrival times, a tool called the Flight Status Monitor (FSM) forecasts the number of arrivals that will be ready to land for time windows into the future. The tool that gives the FSM predictions for LGA is the Airport Arrival Demand Chart (AADC) via the FAA website [8]. Figure 4-6 shows an example of the AADC predictions for LGA.

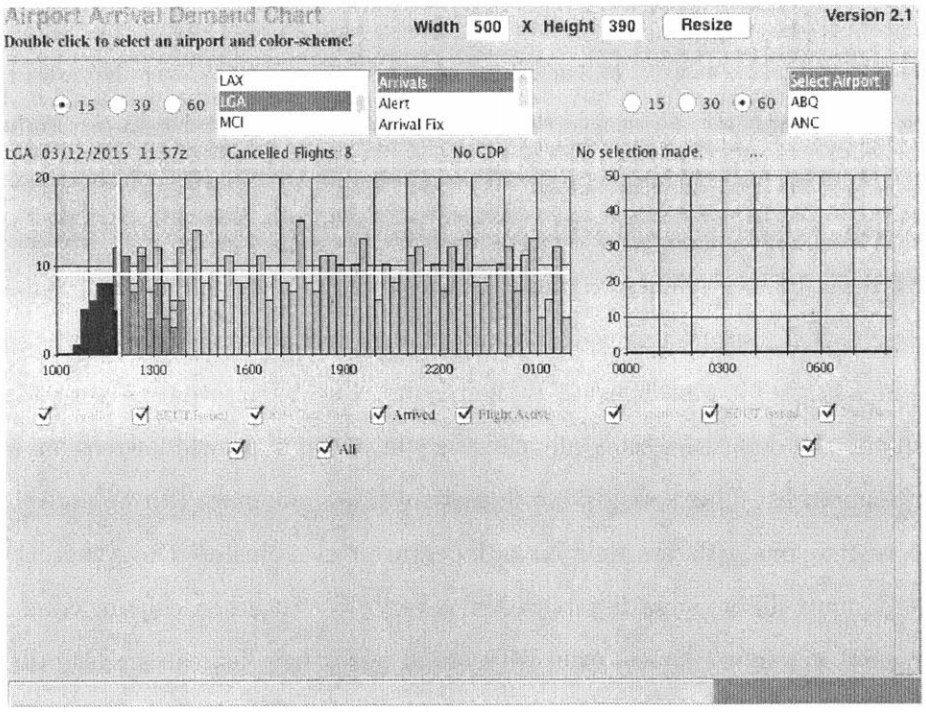

Figure 4-6: Airport Arrival Demand Chart for LGA on 3/12/2015. The bars indicate how many aircraft will be approaching LGA for landing during each 15-minute time window.

Notice the horizontal line and the vertical line marked with an arrow in Figure 4-6. The horizontal line indicates the arrival capacity, which is 9 aircraft per 15 minutes in this example. The vertical line indicates the current time. The bars to the left of this line show how many aircraft actually landed during a 15-minute time period. The bars to the right of this line show the forecasts for 15-minute time windows into the future.

With this information, the predicted arrivals for the next 15, 30, and 60 minutes can be extracted. If the predicted number of arrivals is above the capacity, the predicted arrivals is set to the capacity. The prediction accuracy analysis that follows establishes the perturbations for the variable arrival rate analysis. The AADC data has been gathered over a four month period from January - April 2015.

4.2.1 15-minute Predictions

Figure 4-7 shows the distribution of the difference between AADC arrival predictions and the actual arrivals for 15-minute windows.

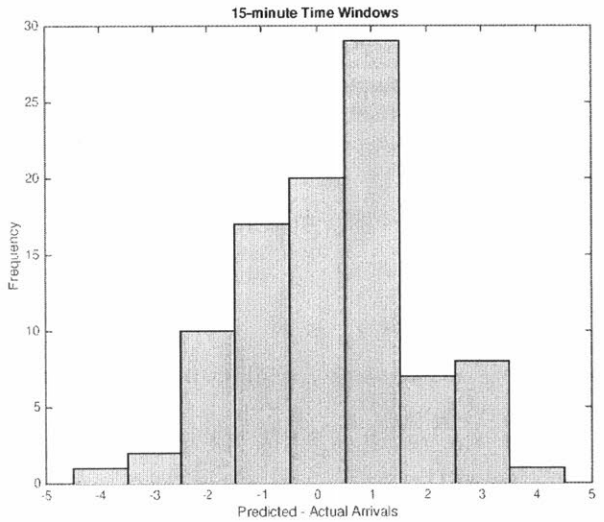

Figure 4-7: Predicted minus actual arrivals at LGA for the next 15-minute window.

The distribution in Figure 4-7 contains 95 data points, with a sample mean of 0.25 and a sample standard deviation of 1.56. The mean and Figure 4-7 show that

the 15-minute predictions tend to over-predict slightly. The distribution seems approximately normal, except for the slight bias towards over-prediction. Bootstrapping the sample gives confidence intervals for the mean and standard deviation. A 95% confidence interval for the mean is [-0.0673, 0.5497], and a 95% confidence interval for the standard deviation is [1.3459, 1.7621].

Similar to the departure schedule perturbations, an arrival rate perturbation is drawn from a normal distribution with mean 0 and standard deviation of 1, an approximation to Figure 4-7. Because arrival rates express the number of aircraft to land in a certain time window, the arrival rate perturbation rounds to the nearest whole number. The new arrival rate a^* then becomes

$$a^* = a + a_p, \tag{4.2}$$

where a is the original arrival rate and a_p is the arrival rate perturbation. Equation 4.2 updates all of the original arrival rates for each time window. From there, the simulation of airport operations with PRC policies proceeds as usual. Again, because perturbing the arrival rate only results in one different arrival schedule, the Monte Carlo method runs the simulation 50 times, each time with a different arrival schedule. The results then provide a distribution of expected policy performance by accounting for arrival uncertainty.

4.2.2 N-control Results

Figures 4-8 and 4-9 contain the results of the N-control Monte Carlo method simulations with the arrival rate perturbations for a 15-minute time window and time horizon of 0. Figure 4-8 shows the total taxi-out benefits for each of the 50 simulations from the Monte Carlo method. Figure 4-9 shows the percent taxi-out reduction, separated by airline, for each of the 50 simulations from the Monte Carlo method. Figure 4-8 provides an illustration of the total policy benefits, while 4-9 reveals the distribution of savings a given airline can expect due to a variable arrival rate.

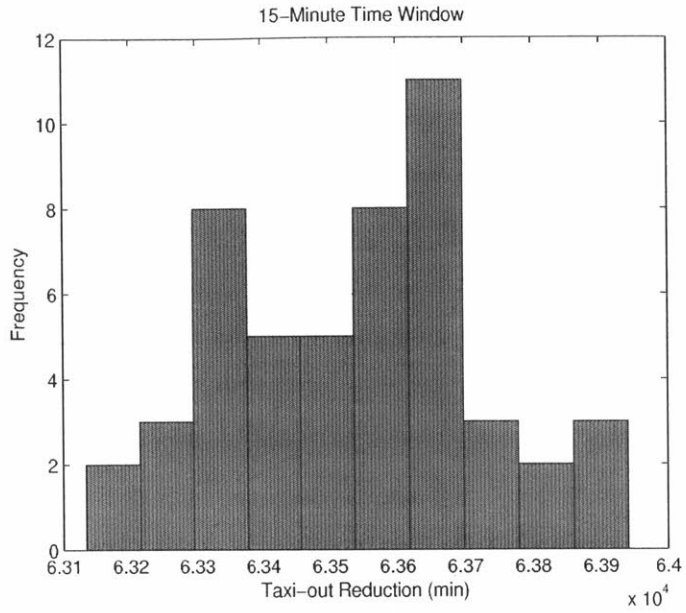

Figure 4-8: Frequency of total taxi-out reduction benefits for the N-control 50 Monte Carlo method simulations with arrival rate uncertainty.

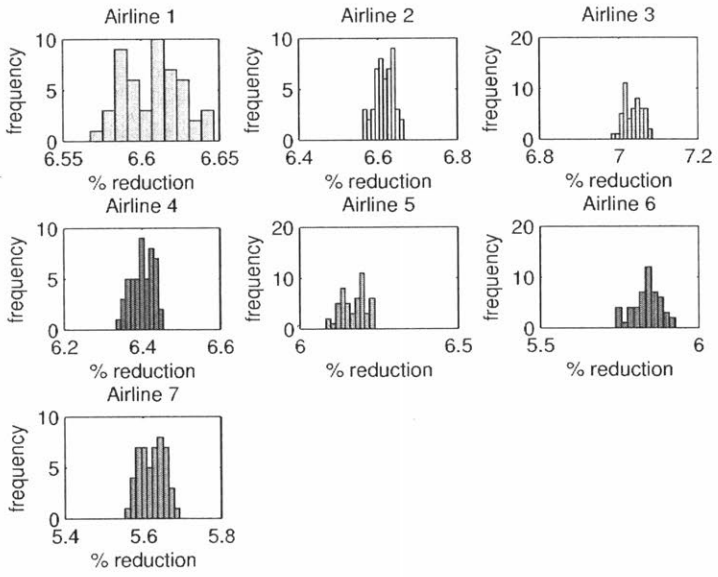

Figure 4-9: Percent taxi-out reduction by airline for the N-control 50 Monte Carlo method simulations with arrival rate uncertainty.

Recall that the total taxi-out reduction for the 15-minute time window, 0 time horizon N-control simulations for LGA for Summer 2013 is 63,773 minutes. The mean total taxi-out reduction from the 50 Monte Carlo method simulations is 63,536 minutes. This indicates that the arrival rate uncertainty reduces the N-control policy benefits to 99.6% of the policy benefits in the case with no uncertainty. Also, Figure 4-9 shows that all airlines can expect a percent taxi-out reduction between 5% and 7% compared to their operations without metering, even with a variable arrival rate. Because the total and airline benefits remain virtually identical to the benefits in the case with no uncertainty, the N-control policy performs well when accounting for a variable arrival rate. Relative to the variable departure schedule, the variable arrival rate barely impacts the results of the simulations.

4.2.3 Dynamic Programming Results

The dynamic programming policy simulations have the same arrival schedule as the N-control policy simulations, so the variable arrival rate analysis for dynamic programming uses the same arrival rate perturbation method described by Equation 4.2. Figures 4-10 and 4-11 contain the results of the dynamic programming Monte Carlo method simulations with the arrival rate perturbations for a 15-minute time window and time horizon of 0. These figures correspond to Figures 4-10 and 4-11 for the N-control variable arrival rate analysis to allow for direct comparison between PRC policies.

Recall that the total taxi-out reduction for the 15-minute time window, 0 time horizon dynamic programming simulations for LGA for Summer 2013 is 216,120 minutes. The mean total taxi-out reduction from the 50 Monte Carlo method simulations is 215,938 minutes. This indicates that the arrival rate uncertainty reduces the dynamic programming policy benefits to 99.9% of the policy benefits in the case with no uncertainty.

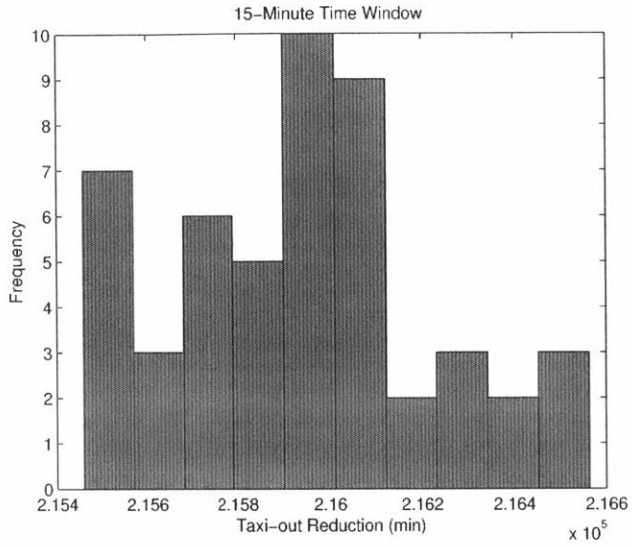

Figure 4-10: Frequency of total taxi-out reduction benefits for the dynamic programming 50 Monte Carlo method simulations with arrival rate uncertainty.

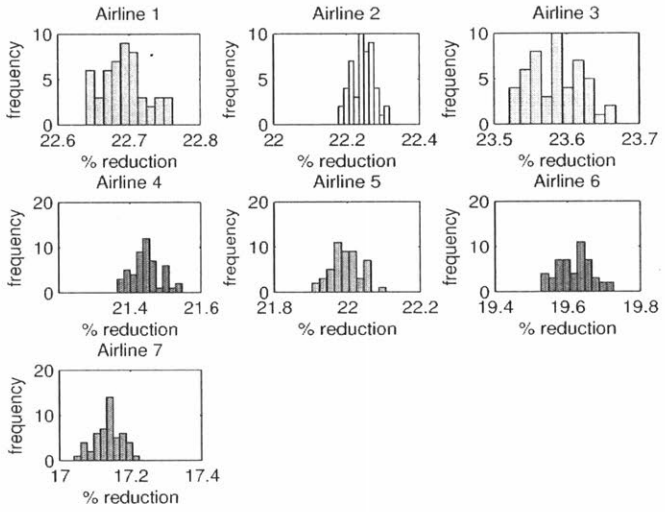

Figure 4-11: Percent taxi-out reduction by airline for the dynamic programming 50 Monte Carlo method simulations with arrival rate uncertainty.

Figure 4-11 shows that all airlines can expect a percent taxi-out reduction comparable to their operations without metering, even with a variable arrival rate. Like the

N-control policy, the dynamic programming policy performs well when accounting for a variable arrival rate. With the N-control policy maintaining 99.6% of policy benefits and the dynamic programming policy maintaining 99.9% of policy benefits, the arrival rate uncertainty does not significantly affect the PRC policy performance.

4.3 Overall Uncertainty

Section 4.1 isolates and examines the uncertainty in the departure schedule. Section 4.2 does the same for the uncertainty in the arrival rate. To make the simulations more realistic, one simulation can contain these two uncertainty sources. By accounting for both of these common types of uncertainty, the simulation provides more realistic results, revealing the effects of the overall uncertainty.

The methods for perturbing the departure schedule and arrival schedule mimic the methods used in Section 4.1 and 4.2. Equations 4.1 and 4.2 change the departure schedule and arrival rate, respectively. Again, with a perturbed departure schedule and arrival rate, the simulations continue as usual. The Monte Carlo method runs the overall uncertainty simulations 50 times.

4.3.1 N-control Results

Recall that the total taxi-out reduction for the 15-minute time window, 0 time horizon N-control simulations for LGA for Summer 2013 is 63,773 minutes. The mean total taxi-out reduction from the 50 Monte Carlo method simulations is 58,246 minutes. This indicates that the overall uncertainty reduces the N-control policy benefits to 91.3% of the policy benefits in the case with no metering. Figure 4-12 shows the N-control policy benefits disribution for the 50 simulations, while Figure 4-13 shows the percent taxi-out reduction distribution for each airline.

Comparing the overall uncertainty results to the departure and arrival uncertainty results shows that the departure uncertainty dominates. Relative to the policy benefits with no uncertainty, the departure uncertainty simulations have 90.7% of the benefits, while the overall uncertainty simulations have 91.3% of the benefits. This

follows directly from the individual uncertainty analyses where the arrival rate uncertainty had very little effect on the policy benefits.

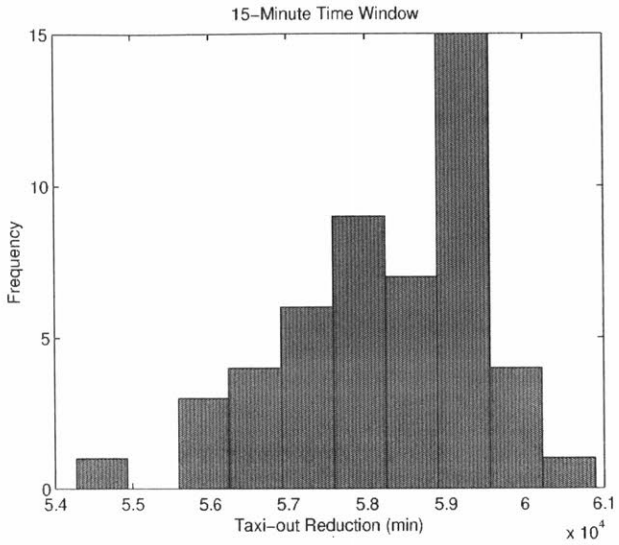

Figure 4-12: Frequency of total taxi-out reduction benefits for the N-control 50 Monte Carlo method simulations with overall uncertainty.

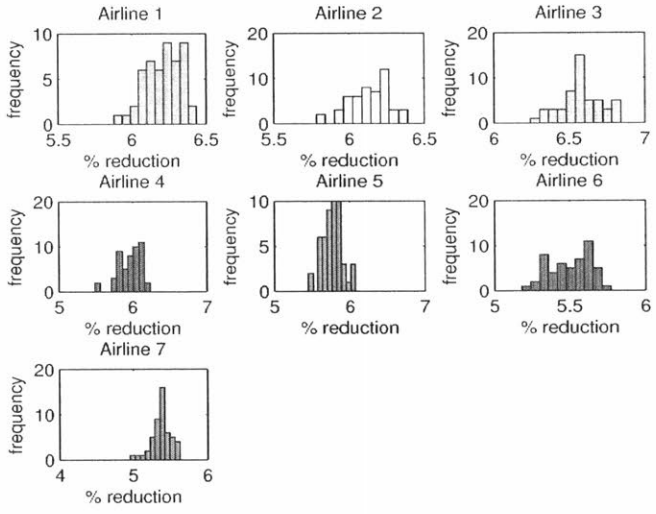

Figure 4-13: Percent taxi-out reduction by airline for the N-control 50 Monte Carlo method simulations with overall uncertainty.

4.3.2 Dynamic Programming Results

Recall that the total taxi-out reduction for the 15-minute time window, 0 time horizon dynamic programming simulations for LGA for Summer 2013 is 216,120 minutes. The mean total taxi-out reduction from the 50 Monte Carlo method simulations is 203,423 minutes. This indicates that the overall uncertainty reduces the dynamic programming policy benefits to 94.1% of the policy benefits in the case with no uncertainty. Figure 4-14 shows the dynamic programming policy benefits distribution for the 50 simulations, while Figure 4-15 shows the percent taxi-out reduction distribution for each airline. Like the N-control results with overall uncertainty, the departure schedule uncertainty drives the reduction in policy benefits and the arrival rate uncertainty has little effect on the results. Relative to the policy benefits with no uncertainty, the departure uncertainty simulations have 94.3% of the benefits, while the overall uncertainty simulations have 94.1% of the benefits. Again, this intuitive result follows directly from both individual uncertainty simulations.

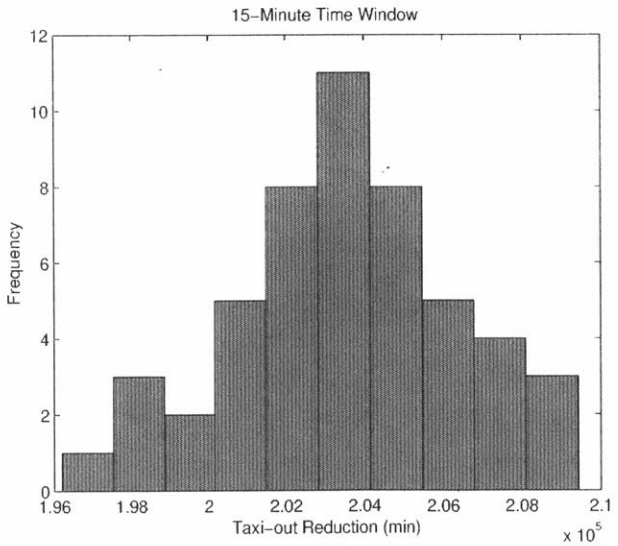

Figure 4-14: Frequency of total taxi-out reduction benefits for the dynamic programming 50 Monte Carlo method simulations with overall uncertainty.

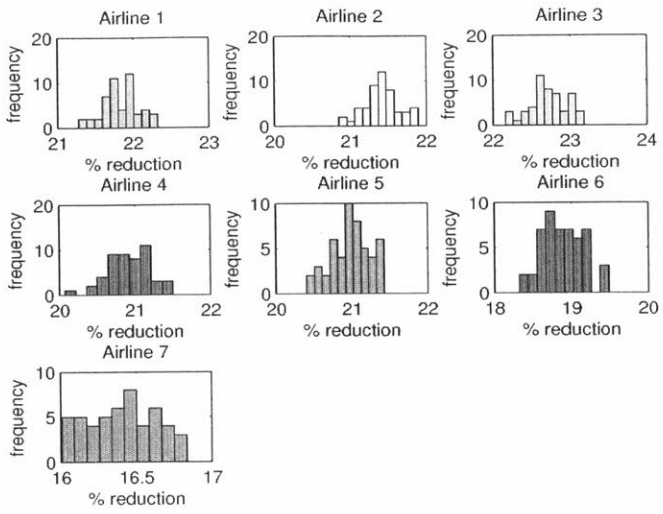

Figure 4-15: Percent taxi-out reduction by airline for the dynamic programming 50 Monte Carlo method simulations with overall uncertainty.

4.4 Variation of Parameters with Uncertainty

The operational uncertainty analysis above only considers 15-minute time windows with a time horizon of 0. This enables a closer look at the results, but the operational uncertainty can be added to simulations with variable policy parameters. For each PRC policy, the departure, arrival, and overall uncertainty analyses extend to time windows and time horizons of different lengths. The following sections present the total taxi-out reduction across all airlines for each different simulation of both PRC policies.

4.4.1 Arrival Rate Predictions

For the arrival rate uncertainty analysis, the arrival rate perturbation came from fitting a normal distribution to the predicted minus actual arrival rates at LGA. To extend the arrival rate analysis to longer time windows, the arrival rate predictions for these time windows must have a corresponding uncertainty. Because the time horizons

only consider 15-minute time windows, the arrival rate perturbations for each time window in a time horizon equal the arrival rate perturbations for 15-minute time windows.

30-minute Predictions

Figure 4-16 shows the distribution of the difference between AADC arrival predictions and the actual arrivals for 30-minute windows.

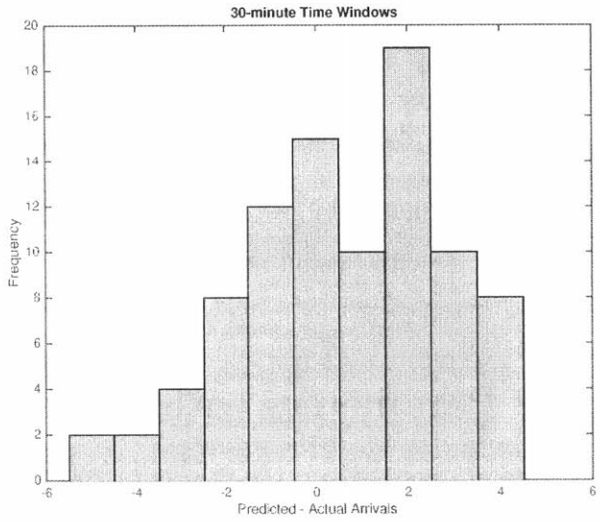

Figure 4-16: Predicted minus actual arrivals at LGA for the next 30-minute window.

The distribution in Figure 4-16 contains 90 data points, with a sample mean of 0.58 and a sample standard deviation of 2.21. The mean and Figure 4-16 shows that the 30-minute predictions tend to over-predict slightly, more so than the 15-minute predictions. The distribution seems approximately normal, except for the slight bias towards over-prediction, as well as a high upper tail. Bootstrapping the sample gives confidence intervals for the mean and standard deviation. A 95% confidence interval for the mean is [0.1393, 1.0463], and a 95% confidence interval for the standard deviation is [1.9129, 2.4807]. As an approximation to Figure 4-16, the arrival rate perturbation is drawn from a normal distribution with mean 0 and standard deviation of 2.

60-minute Predictions

Figure 4-17 shows the distribution of the difference between AADC arrival predictions and the actual arrivals for 60-minute windows.

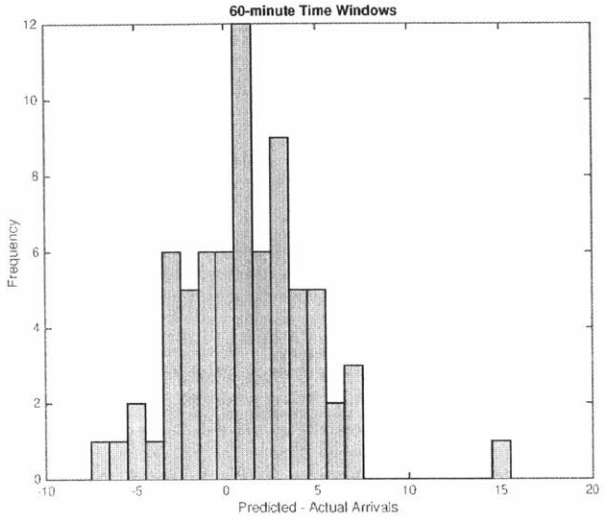

Figure 4-17: Predicted minus actual arrivals at LGA for the next 60-minute window.

The distribution in Figure 4-17 contains 71 data points, with a sample mean of 1.17 and a sample standard deviation of 3.59. The mean and Figure 4-17 shows that the 60-minute predictions tend to over-predict slightly, more so than the 15-minute and 30-minute predictions. The distribution seems approximately normal. Bootstrapping the sample gives confidence intervals for the mean and standard deviation. A 95% confidence interval for the mean is [0.3394, 1.9974], and a 95% confidence interval for the standard deviation is [2.7116, 4.3431]. As an approximation to Figure 4-17, the arrival rate perturbation is drawn from a normal distribution with mean 0 and standard deviation of 4.

4.4.2 N-control Results

Figures 4-18 and 4-19 summarize the taxi-out reduction with operational uncertainty for both the time window and time horizon variability, respectively.

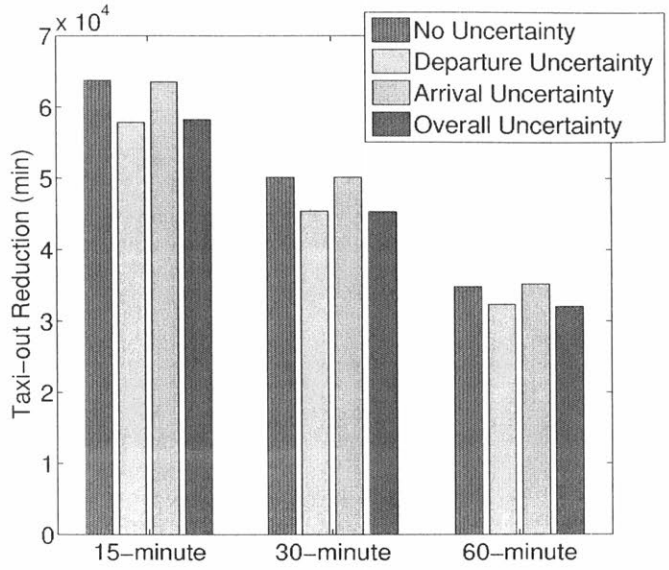

Figure 4-18: Total taxi-out reduction for the July-August 2013 LGA N-control time window simulations.

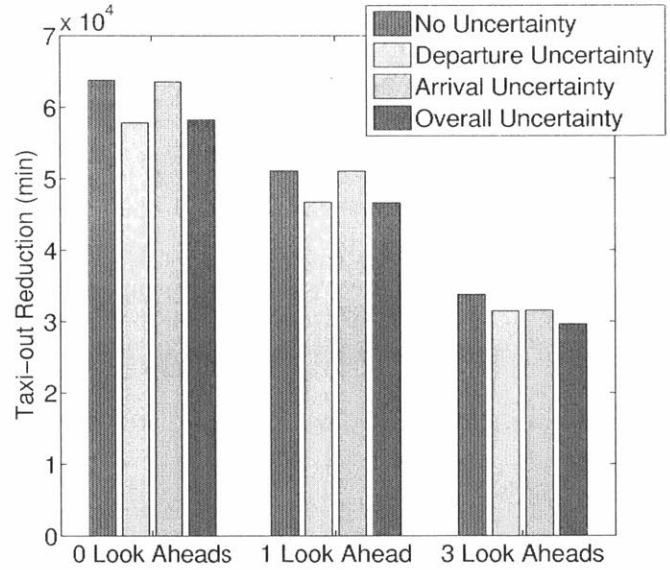

Figure 4-19: Total taxi-out reduction for the July-August 2013 LGA N-control time horizon simulations.

The case with no uncertainty for each time window and time horizon length represents the one simulation done in Chapter 3. The uncertainty cases for each time window and time horizon represent the average taxi-out reduction of 50 Monte Carlo Method simulations. Figure 4-18 shows the decrease in taxi-out reduction with increasing time-window lengths. The departure uncertainty dominates the decrease in taxi-out reduction compared to the arrival rate uncertainty. The arrival rate has very little effect on the taxi-out reduction results. The same trends can be seen in Figure 4-19. However, the decrease in taxi-out reduction is less severe compared to the decrease in Figure 4-18. This shows that extending the time window lengths has a larger negative effect on taxi-out reduction compared to extending the length of the time horizon. This trend, also seen in Chapter 3, also extends to the simulations that include uncertainty.

4.4.3 Dynamic Programming Results

Figures 4-20 and 4-21 summarize the taxi-out reduction with operational uncertainty for both the time window and time horizon variability, respectively.

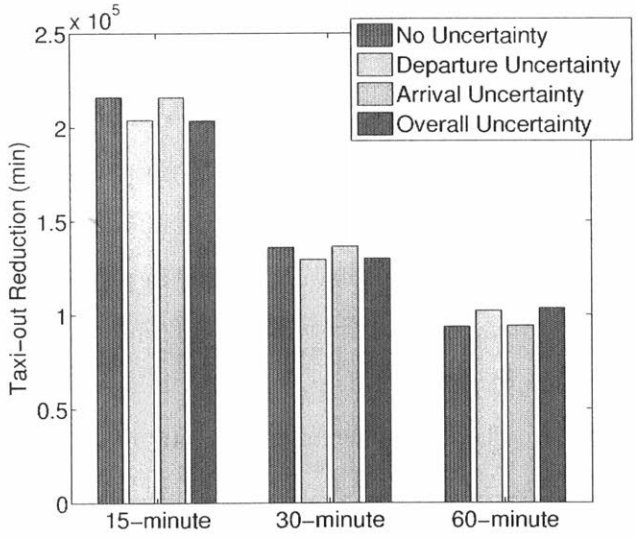

Figure 4-20: Total taxi-out reduction for the July-August 2013 LGA dynamic programming time window simulations.

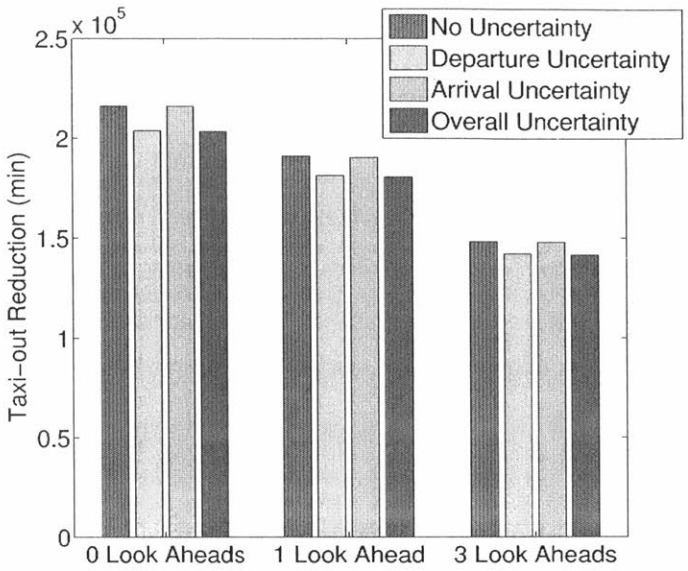

Figure 4-21: Total taxi-out reduction for the July-August 2013 LGA dynamic programming time horizon simulations.

The case with no uncertainty for each time window and time horizon length represents the one simulation done in Chapter 3. The uncertainty cases for each time window and time horizon represent the average taxi-out reduction of 50 Monte Carlo Method simulations. Figure 4-20 shows the sharp decrease in taxi-out reduction with increasing time window lengths. Like the 15-minute time window analysis, the departure schedule uncertainty has a larger effect on taxi-out reduction than does the arrival rate uncertainty. The overall uncertainty analysis results virtually match the departure schedule uncertainty analysis results for all time window lengths. While the 15-minute and 30-minute time window simulations have a decrease in taxi-out reduction due to departure schedule uncertainty, the 60-minute time window simulation has a slight increase in taxi-out reduction. The slight increase may be due to the spreading out of the departure schedule through perturbations. The perturbations could result in more flights being metered compared to the original schedule. Figure 4-21 shows the gradual decrease in taxi-out reduction with incresing time horizon length. The longer time horizon results also indicate the larger effect of the departure

schedule uncertainty compared to the arrival rate uncertainty.

4.5 Conclusions

This chapter introduces operational uncertainty to the PRC policy simulations. Both the departure schedule and arrival rates at airports are uncertain, so PRC policies must perform well under such conditions. The simulations account for these uncertainties by perturbing the departure schedule and arrival rate. The Monte Carlo method runs the simulations many times with perturbations to provide a distribution of the effects of operational uncertainty.

The results show that the departure schedule uncertainty has a larger effect on policy benefits than does the arrival rate uncertainty. For both N-control and dynamic programming, the departure schedule uncertainty has less of a negative influence on the results as the time window length increases. In fact, for the 60-minute time window dynamic programming simulation, the departure schedule uncertainty slightly increases the taxi-out reduction. For the time horizon simulations, the influence of each uncertainty remains proportional to the policy benefits with no uncertainty. The arrival rate uncertainty has virtually no effect on taxi-out reduction for any simulation.

Chapter 5

Conclusion

This thesis presents two PRC policies, N-control and dynamic programming, and explores the robustness of the policies through the variation of parameters and operational uncertainty. The N-control policy monitors the airport surface traffic and only meters departures when the departure traffic exceeds a defined capacity for each segment. The dynamic programming policy models the state of the airport surface as the number of departures in the runway queue and the number of departures taxiing to the runway. With a distribution of runway service times and a cost of queuing function, the optimal pushback rate for a time window is found through value iteration. The N-control policy is more straightforward, making possible implementation more likely. The dynamic programming policy is more sophisticated and the pushback rate accounts for the underlying uncertainty of the evolution of airport surface traffic.

Policy parameters, namely the lengths of the time windows and time horizons, affect the pushback rate calculation. The time window governs the length of time for which one pushback rate remains valid. The time horizon governs the number of time windows into the future for which a pushback rate is found in advance. Historically, for airport surface management policies, the time window has been set to 15 minutes and the time horizon has been set to 0. However, airlines and airports may need to alter these parameters to tailor a PRC policy for implementation. Therefore, the effects of the variation of parameters on policy performance must be known. Lastly, the PRC policies require the input of operational data, such as the departure schedule

and arrival rate of aircraft. The uncertainty in these data sources also has a direct effect on policy performance.

5.1 Summary of results

5.1.1 Variation of Parameters

Simulations of the both PRC policies with time windows of 15, 30, and 60 minutes ran airport operations at LGA for July-August 2013. As the length of the time window increases for the N-control policy, the policy benefits of taxi-out reduction decrease. Relative to the benefits of the 15-minute time window simulations for the N-control policy, the 30-minute time window simulation has 78.6% of the total taxi-out time reduction and the 60-minute time window simulation has 54.7% of the total taxi-out time reduction. This trend agrees with intuition as one pushback rate, calculated based on airport conditions at the beginning of a time window, remains valid for a longer period of time. The same trend can be seen in the dynamic programming policy results. Relative to the benefits of the 15-minute time window simulations for the dynamic programming policy, the 30-minute time window simulation has 62.9% of the total taxi-out time reduction and the 60-minute time window simulation has 43.3% of the total taxi-out time reduction. However, note that the absolute benefits of the dynamic programming policy are considerably greater than the absolute benefits of the N-control policy.

Also, simulations of both PRC policies with time horizons of 0, 1, and 3 ran operations at LGA for July-August 2013. Again, as the length of the time horizon increases for the N-control policy, the policy benefits of taxi-out reduction decrease. However, compared to the time-window simulations, the rate of decrease in policy benefits is lower for increasing time horizons. Relative to the benefits of the 0 time horizon simulations for the N-control policy, the 1 time horizon simulation has 80.1% of the total taxi-out time reduction and the 3 time horizon simulation has 53.0% of the total taxi-out time reduction. The dynamic programming policy exhibits a smaller

taxi-out reduction decrease for longer time horizons. Relative to the benefits of the 0 time horizon simulations for the dynamic programming policy, the 1 time horizon simulation has 88.4% of the total taxi-out time reduction and the 3 time horizon simulation has 68.7% of the total taxi-out time reduction. However, note that the absolute benefits of the dynamic programming policy are considerably greater than the absolute benefits of the N-control policy.

5.1.2 Operational Uncertainty

Because PRC policies meter the rate of departure pushbacks, the departure schedule is a crucial input to simulations. As such, uncertainty in the departure schedule changes the departure demand for pushbacks in a given time window. In this thesis, perturbing the departure schedule with random samples from a normal distribution approximates the departure schedule uncertainty. Also, the Monte Carlo method runs the simulations 50 times, each with a different perturbed departure schedule. For the 15-minute time window N-control simulations with departure uncertainty, the mean total taxi-out reduction is 90.7% of the total taxi-out reduction from the 15-minute time window simulation with no uncertainty. For the 15-minute time window dynamic programming simulations with departure uncertainty, the mean total taxi-out reduction is 94.3% of the total taxi-out reduction from the 15-minute time window simulation with no uncertainty. The decrease in taxi-out reduction, relative to the baseline simulations, is greater for the N-control policy, but departure schedule uncertainty reduces taxi-out reduction benefits between 5% and 10% for both PRC policies.

The N-control policy uses the arrival rate in a time window to find the predicted throughput through regression trees, which then enters into the calculation for the departure pushback rate. The dynamic programming policy also uses the arrival rate in a time window to find the predicted throughput, but this information enters into the departure pushback rate calculation through the distribution of service times. The uncertainty in arrival rate predictions is shown by the difference between the actual arrival rate and the predicted arrival rate by the AADC. Similar to the departure

schedule perturbations, the simulations perturb the arrival rates by a random sample from a normal distribution, an approximation of the predicted minus actual arrival rate distribution. Also, the Monte Carlo method runs the simulations 50 times, each with different perturbed arrival rates for each time window. However, the arrival rate uncertainty has little effect on the taxi-out reduction. For the 15-minute time window N-control simulations with arrival uncertainty, the mean total taxi-out reduction is 99.6% of the total taxi-out reduction from the 15-minute time window simulation with no uncertainty. For the 15-minute time window dynamic programming simulations with arrival uncertainty, the mean total taxi-out reduction is 99.9% of the total taxi-out reduction from the 15-minute time window simulation with no uncertainty. These reductions in benefits are not significant.

5.1.3 Overall Uncertainty

Naturally, the variation of parameters analysis and operational uncertainty analysis can combine to provide a complete picture of the factors that affect the performance of a PRC policy. Considering this, Tables 5.1 and 5.2 summarize the total taxi-out reduction relative to a comparable simulation with no uncertainty. That is, for each time window or time horizon analysis, the operational uncertainty analysis is performed, as was done for the 15-minute time window, 0 time horizon case. Tables 5.1 and 5.2 provide more detail to the information presented in Figures 4-18, 4-19, 4-20, and 4-21. Clearly, the trends discussed in the overall uncertainty analysis arise for longer time windows and time horizons, with the exception of the 60-minute time window for the dynamic programming policy. This exception needs to be explored further. One possible explanation is that the departure schedule perturbations spread departure times throughout an entire 60-minute time window. This could create sustained demand for longer periods, leading to more metering.

Table 5.1: Overall Uncertainty Analysis for the N-control Policy.

Time Window Analysis	15-minute	30-minute	60-minute
No Uncertainty	100.0%	100.0%	100.0%
Departure Uncertainty	90.7%	90.6%	92.8%
Arrival Uncertainty	99.6%	100.0%	101.0%
Overall Uncertainty	91.3%	90.4%	91.9%
Time Horizon Analysis	0	1	3
No Uncertainty	100.0%	100.0%	100.0%
Departure Uncertainty	90.7%	91.4%	93.1%
Arrival Uncertainty	99.6%	99.9%	93.3%
Overall Uncertainty	91.3%	91.2%	87.7%

Table 5.2: Overall Uncertainty Analysis for the Dynamic Programming Policy.

Time Window Analysis	15-minute	30-minute	60-minute
No Uncertainty	100.0%	100.0%	100.0%
Departure Uncertainty	94.3%	95.3%	109.0%
Arrival Uncertainty	99.9%	100.0%	100.5%
Overall Uncertainty	94.1%	95.7%	110.5%
Time Horizon Analysis	0	1	3
No Uncertainty	100.0%	100.0%	100.0%
Departure Uncertainty	94.3%	94.9%	95.8%
Arrival Uncertainty	99.9%	99.6%	99.8%
Overall Uncertainty	94.1%	94.5%	95.5%

5.2 Contributions of this thesis

Reflecting on the contributions of this thesis in Chapter 1, the main items proposed have been achieved. The variation of parameters analysis presents the policy performance with different time windows and time horizons. Also, the tradeoffs that arise when choosing policy parameters involve policy performance, accuracy, and personnel workload. Combining the results of this analysis with operational constraints and requirements, a PRC policy can be tailored to achieve the required benefits while satisfying the needs of airports and airlines.

The operational uncertainty analysis addresses the problem of the reliability of input data. The departure schedule and arrival rate represent just two operational

uncertainties that affect the performance of PRC policies. The analysis shows a negative effect on performance for departure uncertainty, while arrival uncertainty has a negligible effect on performance. Again, this information gives airports and airlines a realistic expectation of policy performance when exposed to the realities of the airport industry.

5.2.1 Future work

While this thesis serves as the introduction to the problem of uncertainty surrounding airport surface management, much can still be accomplished in this area. This analysis can be extended to include both other airports and other PRC policies. LGA is a notoriously congested airport, serving as an interesting case study for this analysis. However, this analysis must be repeated for smaller and less congested airports to explore the policy performance in the presence of less congestion. Also, other PRC policies may address the problem adequately while being easier to implement in reality.

Considering possible implementation, field trials remain a necessary component of this research. While the N-control policy has been tested in the field, the dynamic programming policy needs to be implemented to learn more about the challenges of implementing a more complicated PRC policy. Field test performance can then be compared to these simulation results. Any similarities or differences will provide further insight into the complexities of implementation.

This thesis gives an overview of the policy parameters and uncertainty sources that affect the performance of PRC policies. This thesis has not considered all of the outside forces that affect policy performance. In terms of policy parameters, the choice of N^* clearly affects the performance of the policy because that definition establishes periods of congestion. A sensitivity analysis on N^* would show how the policy benefits change. The dynamic programming algorithm in this thesis groups segments by their Erlang distribution shape. This is an assumption, but a more thorough analysis could have a unique Erlang distribution for each leaf of each regression tree. While this author believes that approach may be too granular, examining the results would

confirm or deny that assumption.

On the operational uncertainty side, the weather is an obvious uncertainty not examined in this thesis. A study of RAPT values over time compared to the actual weather could give a good approximation to weather uncertainty. Alternatively, the simulations could be done assuming constant clear weather (all RAPT values equal to 0). Then, further simulations could increase the constant RAPT value by 0.5 or 1. The results of that exercise would show the affect of weather on policy performance. If the effects are large, the uncertainty analysis of weather would be very valuable.

Another important aspect of airport surface management is the scheduling and availability of ground crews. Using a virtual queue works well in simulations, but when that queue increases, an adequate ground crew schedule may not be possible. Adding a ground crew constraint to the simulations would provide a more realistic sense of the severity of this problem. Instead of one value of gateholding time, the time spent at the gate would be split into two categories: gateholding time due to the policy and pushback delay due to the unavailability of the ground crew. Assumptions must be made about the number of crews and the time of travel between different gates, but such an algorithm would be a great addition to this research.

To conclude, the driving force behind these ideas for future work is the need to convince airlines and airports to adopt airport surface management policies. Quite a bit of inertia must be overcome before implementation becomes a reality. Care was taken in this thesis to present all of the positives and negatives of PRC policies. Even accounting for operational uncertainty, the PRC policies reduce taxi-out time significantly. If these results continue to hold by accounting for more realistic simulations that include ground crew or weather uncertainty, PRC policy implementation should be in the future of the aviation industry.

Appendix A

LGA Saturation Curves and Regression Trees

A.0.2 31|4; IMC

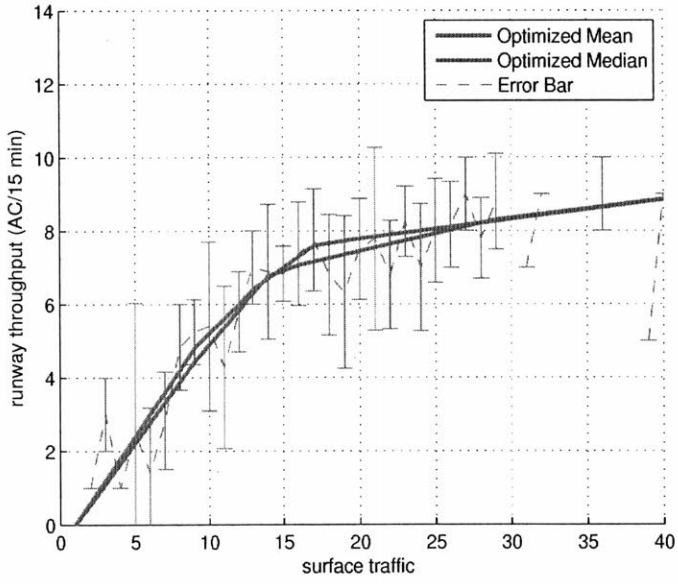

Figure A-1: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 31|4; IMC segment.

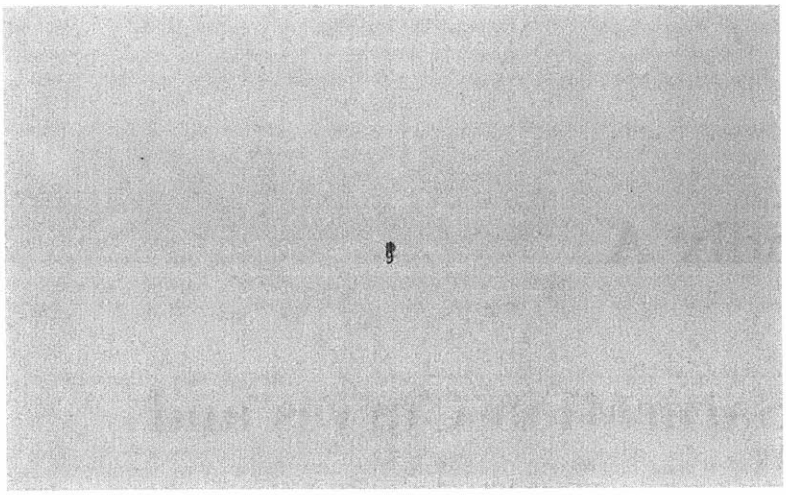

Figure A-2: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (31|4; IMC) segment with a 15-minute window.

A.0.3 22|31; VMC

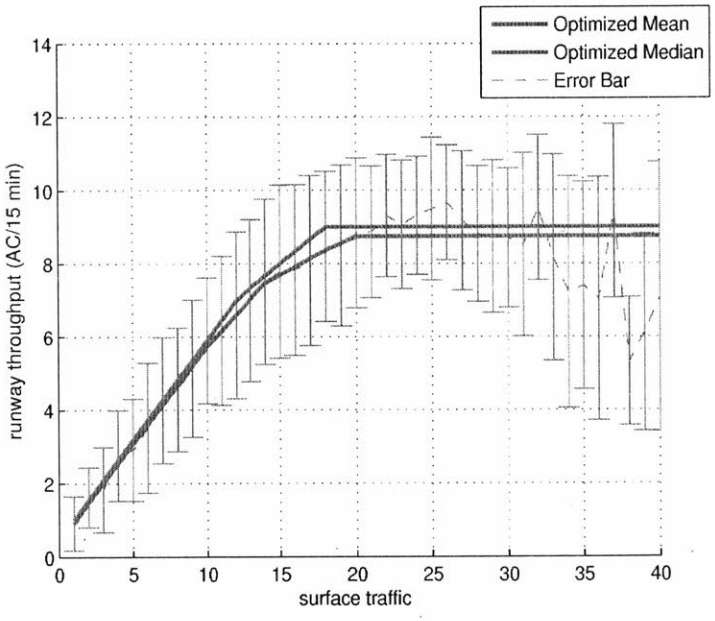

Figure A-3: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 22|31; VMC segment.

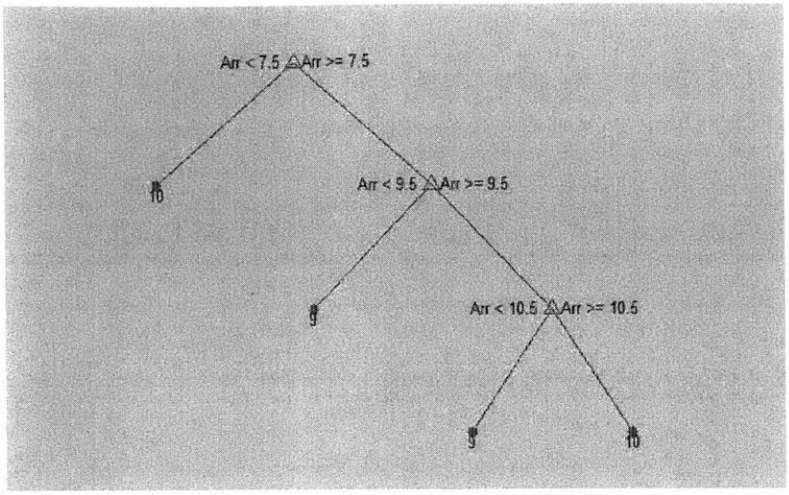

Figure A-4: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (22|31; VMC) segment with a 15-minute window.

A.0.4 22|31; IMC

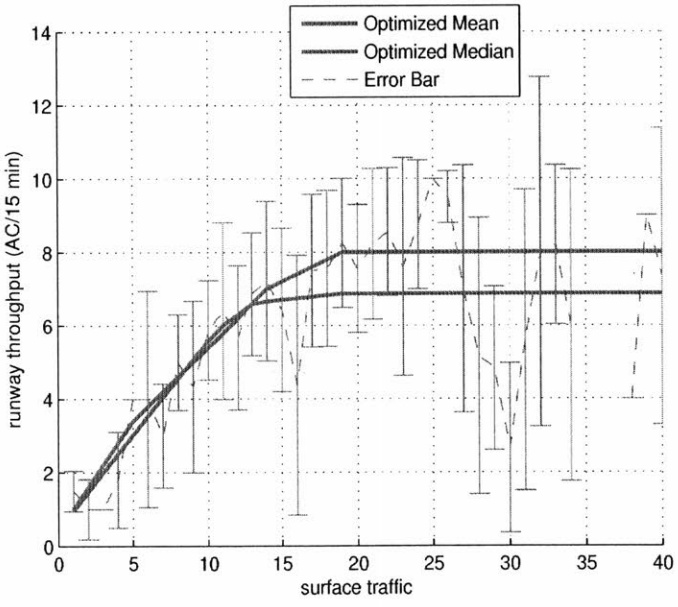

Figure A-5: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 22|31; IMC segment.

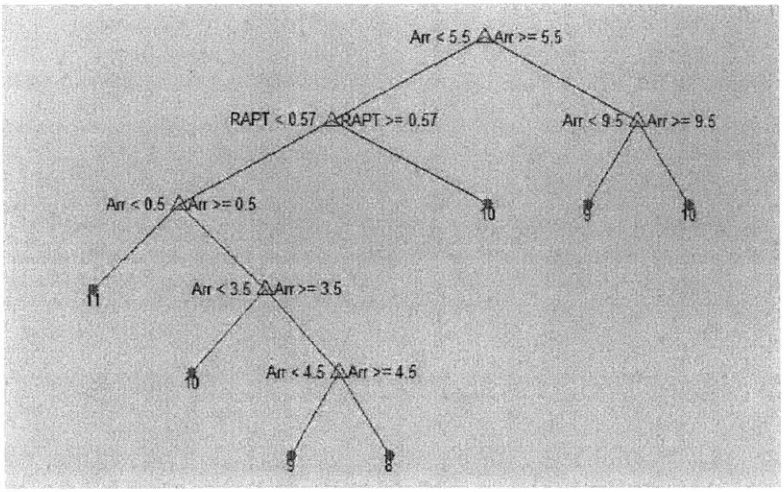

Figure A-6: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (22|31; IMC) segment with a 15-minute window.

A.0.5 31|31; VMC

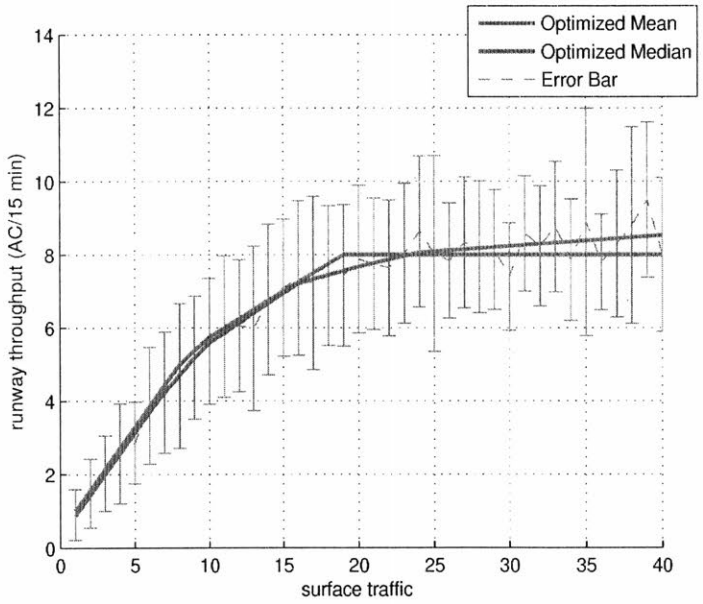

Figure A-7: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 31|31; VMC segment.

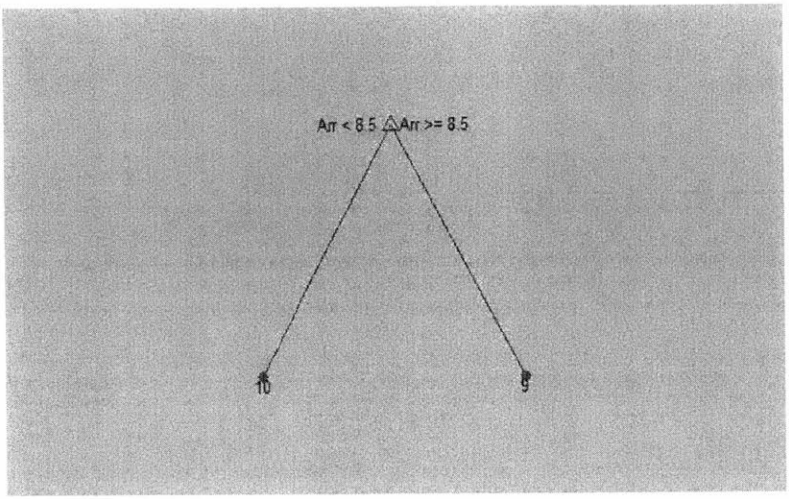

Figure A-8: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (31|31; VMC) segment with a 15-minute window.

A.0.6 31|31; IMC

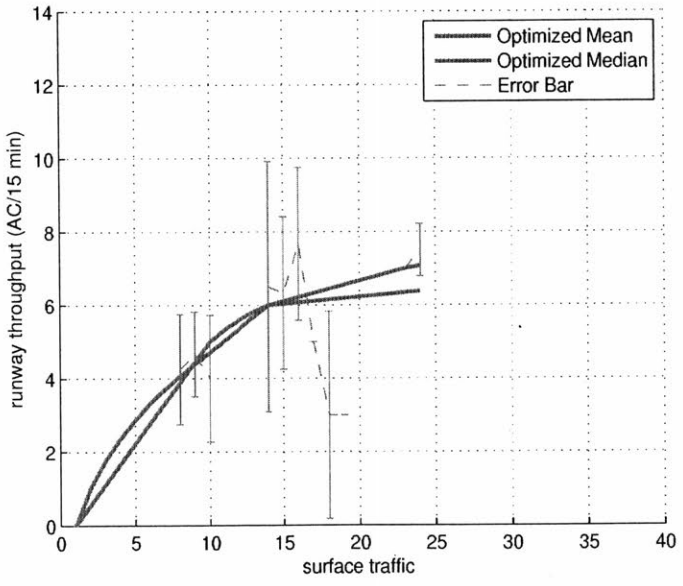

Figure A-9: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 31|31; IMC segment.

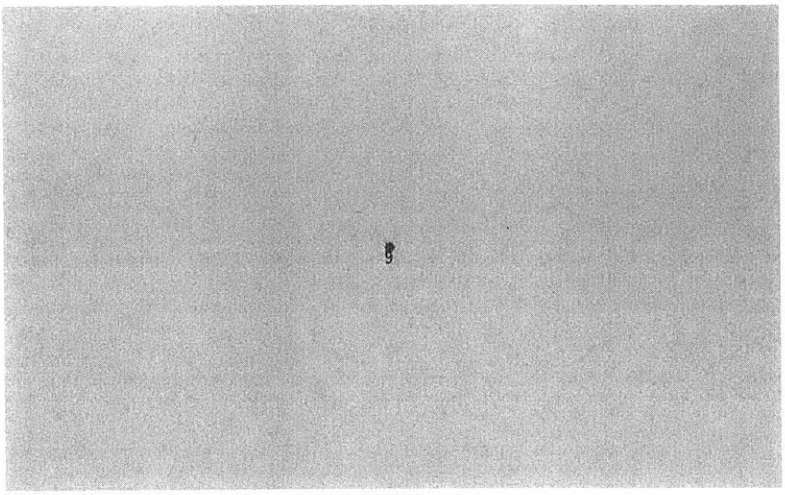

Figure A-10: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (31|31; IMC) segment with a 15-minute window.

A.0.7 4|31; VMC

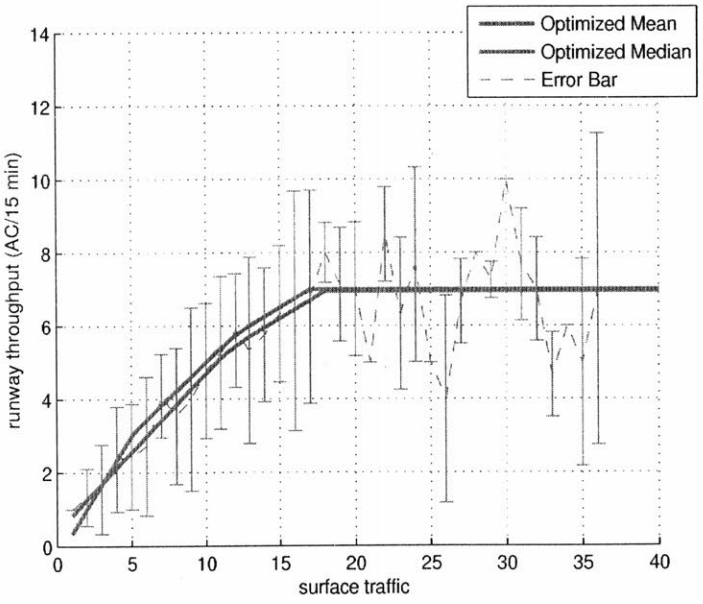

Figure A-11: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 4|31; VMC segment.

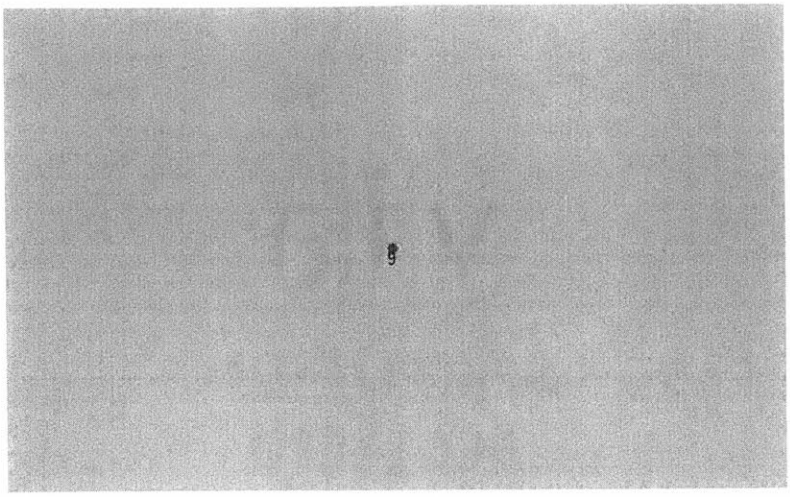

Figure A-12: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (4|31; VMC) segment with a 15-minute window.

A.0.8 4|31; IMC

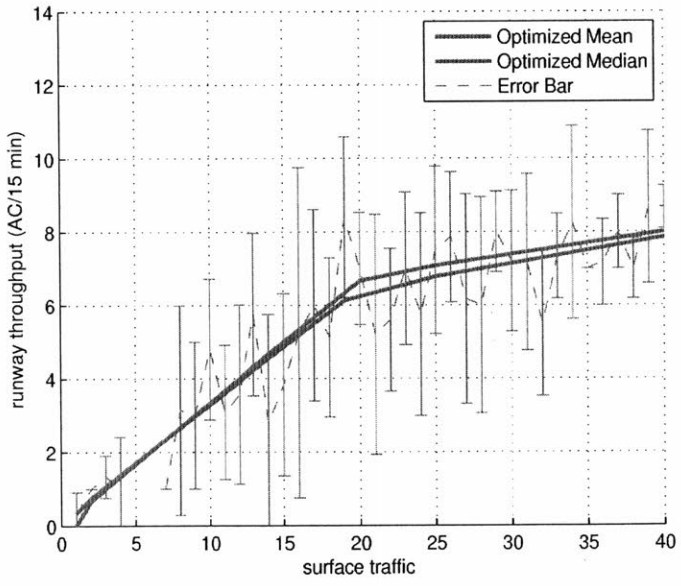

Figure A-13: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 4|31; IMC segment.

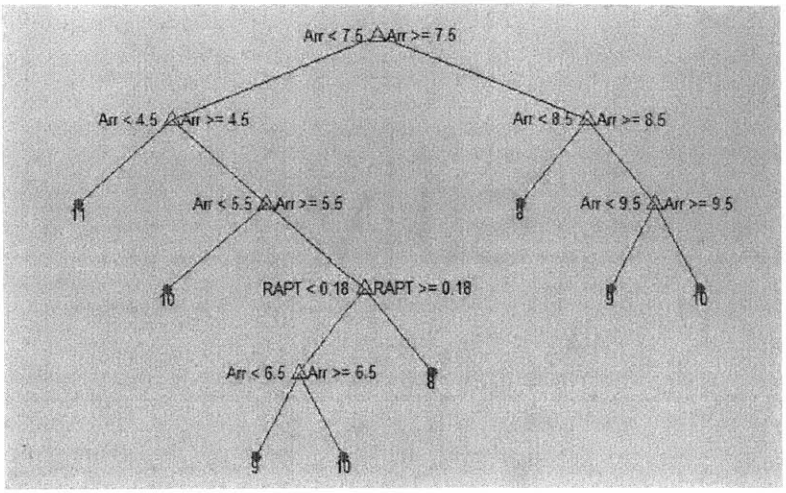

Figure A-14: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (4|31; IMC) segment with a 15-minute window.

A.0.9 22|13; VMC

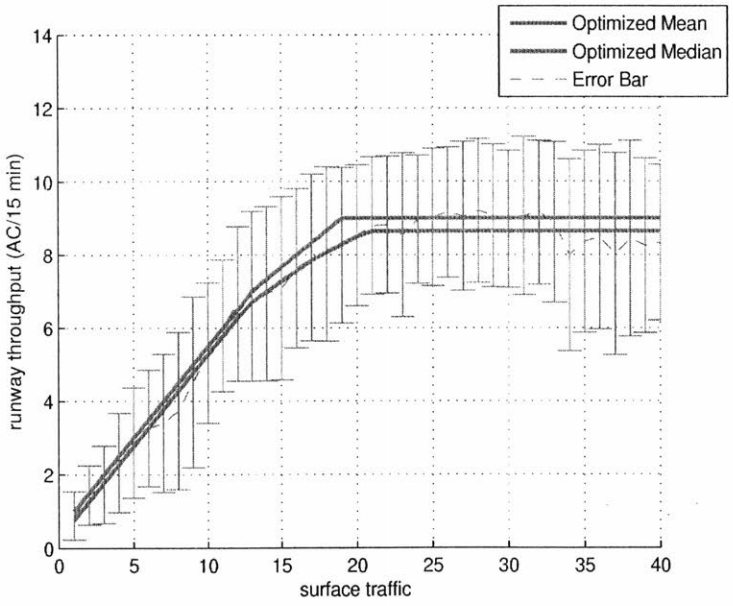

Figure A-15: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 22|13; VMC segment.

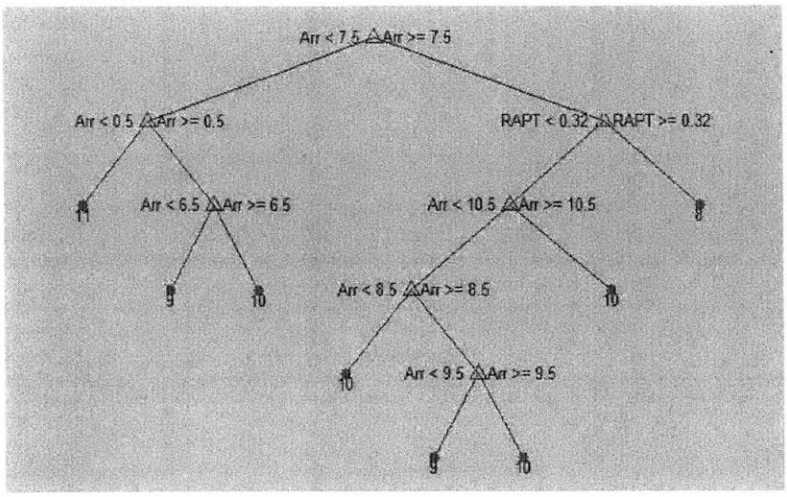

Figure A-16: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (22|13; VMC) segment with a 15-minute window.

A.0.10 22|13; IMC

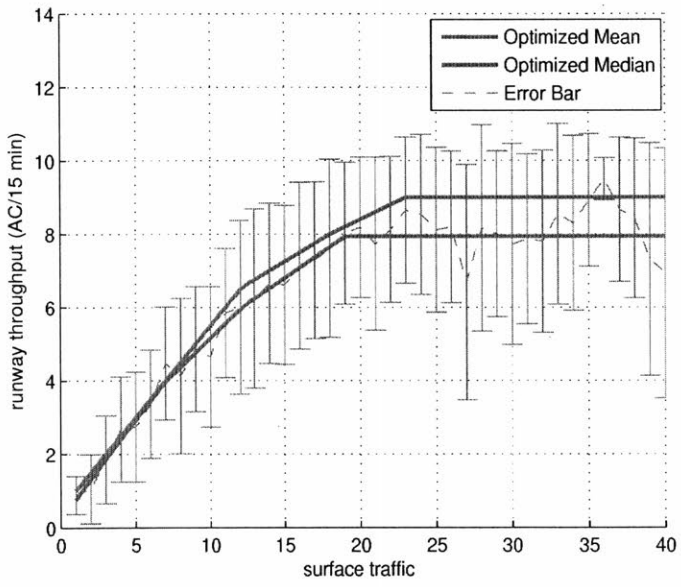

Figure A-17: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 22|13; IMC segment.

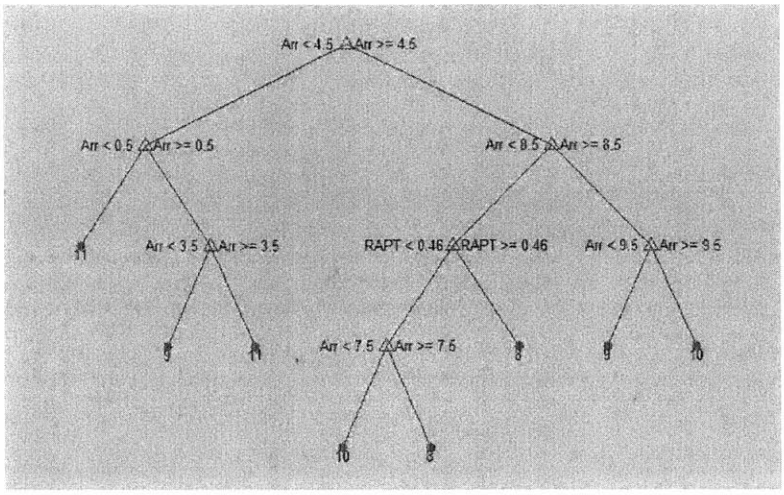

Figure A-18: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (22|13; IMC) segment with a 15-minute window.

A.0.11 4|13; VMC

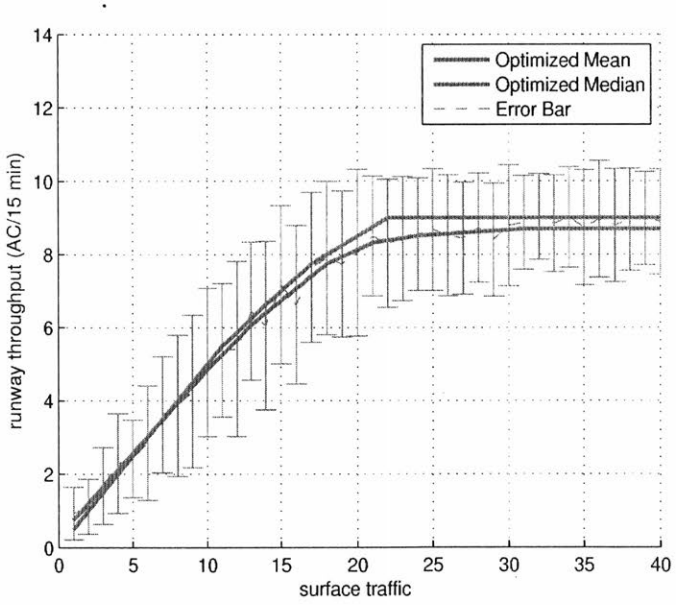

Figure A-19: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 4|13; VMC segment.

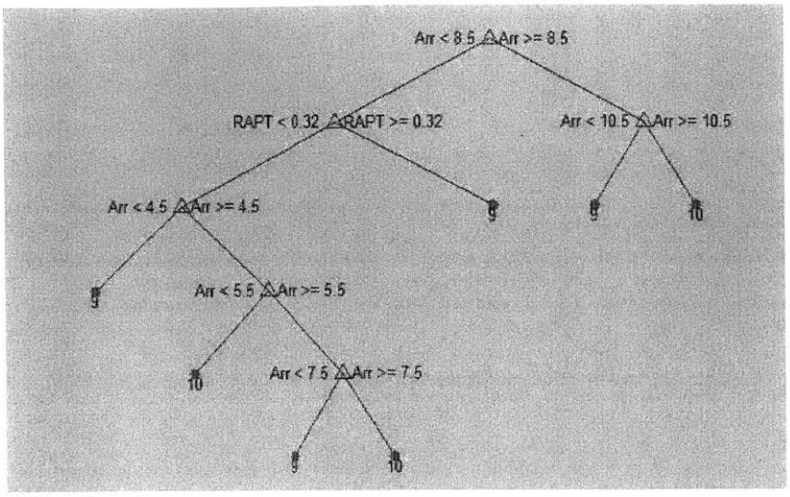

Figure A-20: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (4|13; VMC) segment with a 15-minute window.

A.0.12 4|13; IMC

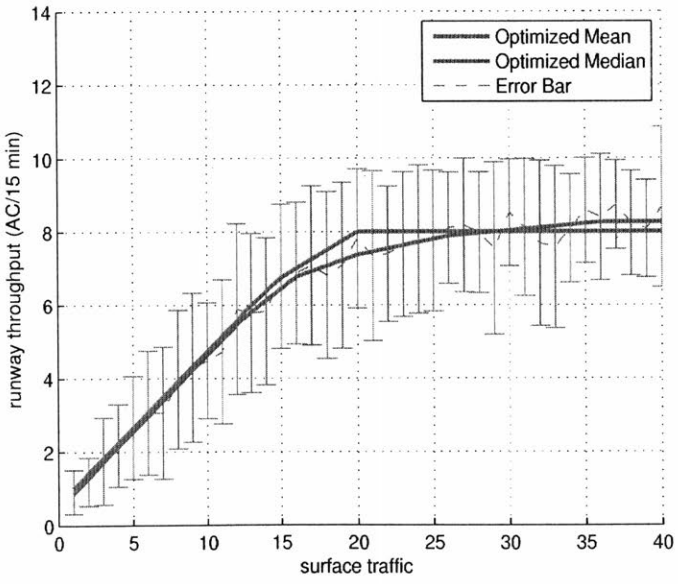

Figure A-21: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 4|13; IMC segment.

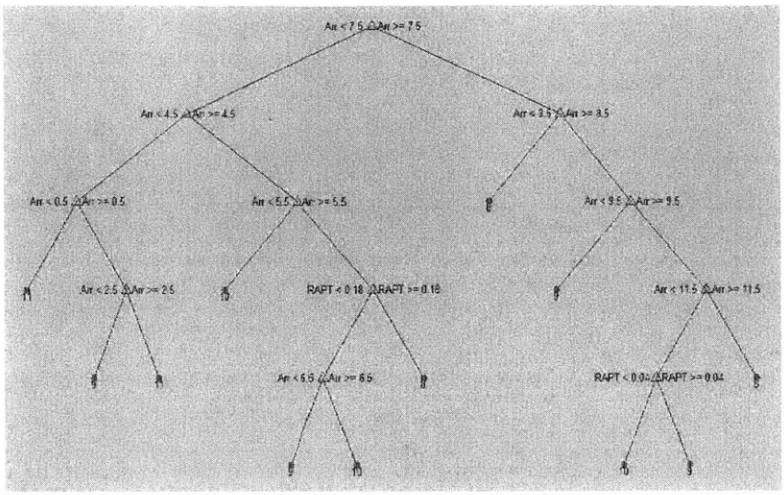

Figure A-22: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (4|13; IMC) segment with a 15-minute window.

A.0.13 4|4; VMC

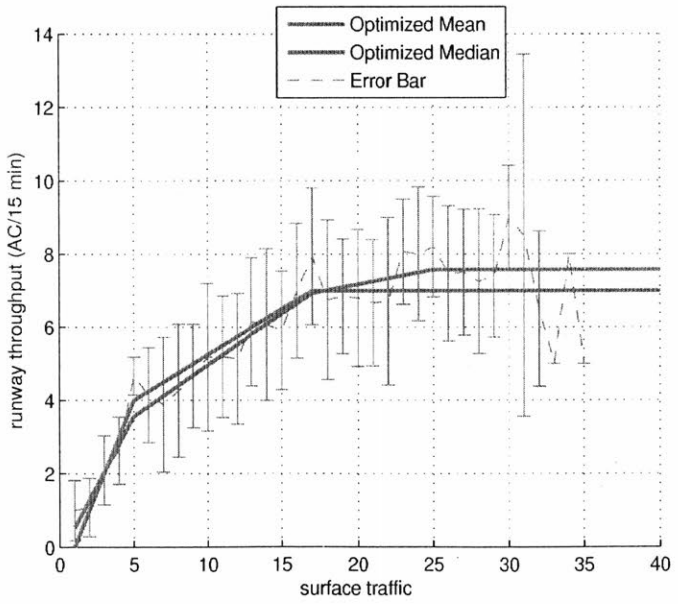

Figure A-23: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 4|4; VMC segment.

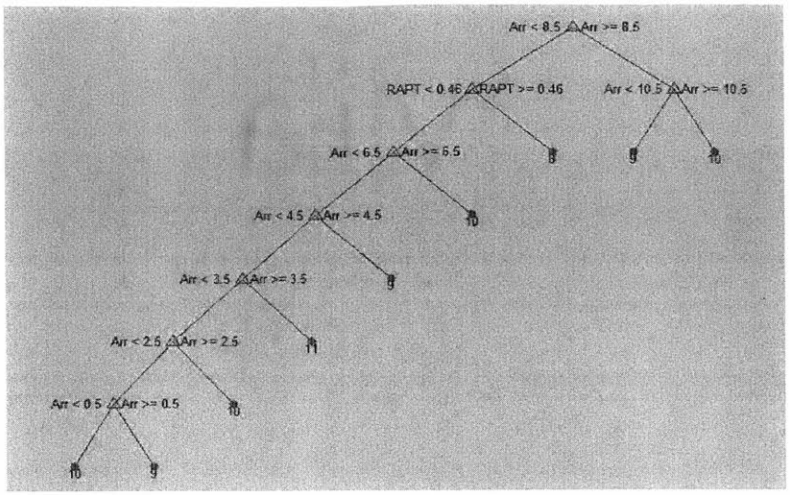

Figure A-24: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (4|4; VMC) segment with a 15-minute window.

A.0.14 4|4; IMC

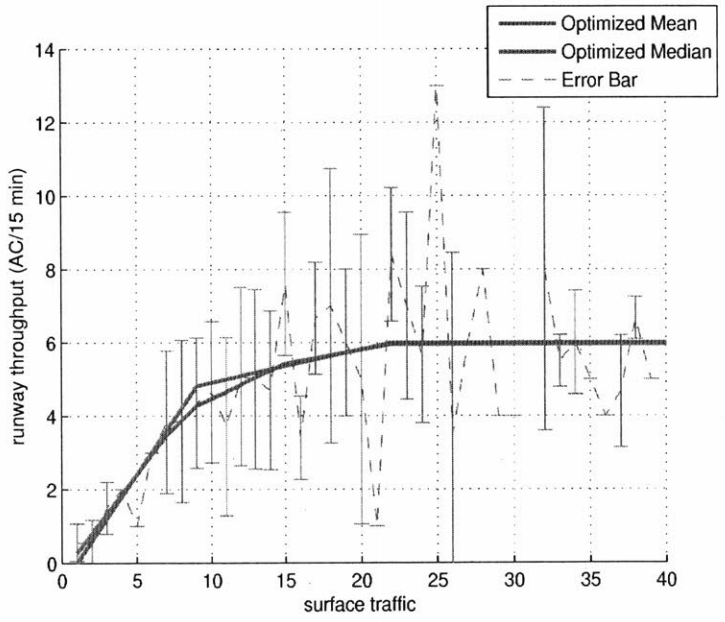

Figure A-25: Departure throughput versus departure aircraft taxiing at LaGuardia Airport for the 4|4; IMC segment.

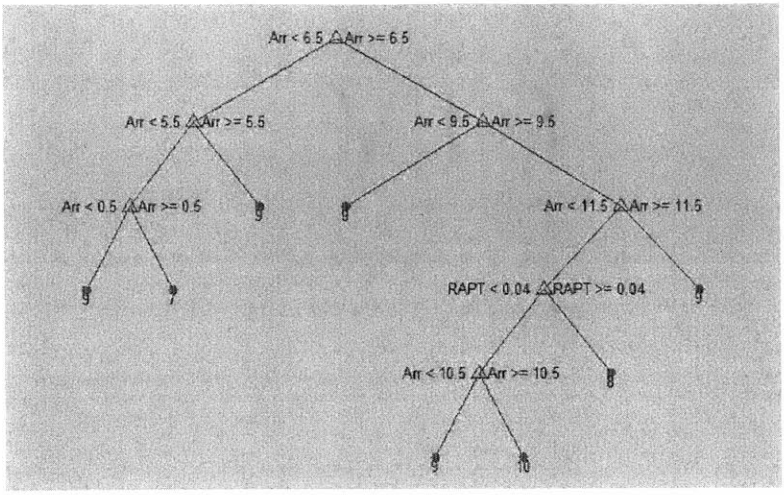

Figure A-26: Regression tree with predicted departure throughput at the leaves for LaGuardia Airport for the (4|4; IMC) segment with a 15-minute window.

Bibliography

[1] P. Burgain, E. Feron, J. Clarke, and A. Darrasse. Collaborative virtual queue: Fair management of congested departure operations benefit analysis. *arXiv Preprint*, (arXiv:0807.0661), 2008.

[2] P. Burgain, O. Pinon, E. Feron, J. Clark, and D. Mavris. On the value of information within a collaborative decision making framework for airport departure operations. *Digital Avionics Systems Conference*, 6250:3.D.3 1–13, 2009.

[3] F. Carr, A. Evans, E. Feron, and J. Clarke. Software tools to support research on airport departure planning. In *Digital Avionics Systems Conference*, 2002.

[4] B. Chandran and H. Balakrishnan. A dynamic programming algorithm to robust runway scheduling. *American Control Conference*, 6250:1161–1165, 2007.

[5] David Chin and Maurizio Castelletti. Comparison of air traffic management-related operational performance: U.S./Europe, Jun 2014.

[6] Joint Economic Committee. Flight delays cost passengers, airlines, and the U.S. economy billions, May 2008.

[7] P. Dell'Olmo and G. Lulli. A dynamic programming approach for the airport capacity allocation problem. *IMA Journal of Management Mathematics*, 14:235–249, 2003.

[8] FAA. Airport arrival demand chart. http://www.fly.faa.gov/.

[9] E. R. Feron, R. J. Hansman, A. R. Odoni, R. B. Cots, B. Delcaire, W. D. Hall, H. R. Idris, A. Muharremoglu, and N. Pujet. The departure planner: A conceptual discussion. Technical report, Massachusetts Institute of Technology, 1997.

[10] H. Khadilkar. Analysis and modeling of airport surface operations. Master's thesis, Massachusetts Institute of Technology, 2011.

[11] H. F. Martinez and H. Balakrishnan. Analysis of potential implementations of pushback rate control at laguardia airport. In *Transportation Research Board 94th Annual Meeting*, number 15-5274, 2015.

[12] U.S. Department of Transportation. Aviation System Performance Metrics, 2014.

[13] Bureau of Transportation Statistics. Form 41 schedule P12A, September 2014.

[14] N. Pujet, B. Delcaire, and E. Feron. Input-output modeling and control of the departure process of congested airports. Technical report, Massachusetts Institute of Technology, 2003.

[15] S. Rathinam, Z. Wood, B. Sridhar, and Y. Jung. Generalized dynamic programming approach for a departure scheduling problem. *AIAA Guidance, Navigation, and Control Conference*, 6250:1–12, 2009.

[16] S. Ravizza, J. Chen, J. Atkin, E. Burke, and P. Stewart. The trade-off between taxi time and fuel consumption in airport ground movement. *Public Transport*, 5(1-2):25–40, 9 2013.

[17] M. Sandberg, I. Simaiakis, H. Balakrishnan, T. G. Reynolds, and R. J. Hansman. A decision support tool for the pushback rate control of airport departures. *IEEE Transactions on Human-Machine Systems*, 44(3):416–421, 6 2014.

[18] I. Simaiakis. Modeling and control of airport departure processes for emissions reduction. Master's thesis, Massachusetts Institute of Technology, 2009.

[19] I. Simaiakis. *Analysis, Modeling, and Control of the Airport Departure Process.* PhD thesis, Massachusetts Institute of Technology, 2013.

[20] I. Simaiakis. Analysis of potential implementations of pushback control at Laguardia airport. Master's thesis, Massachusetts Institute of Technology, 2015.

[21] I. Simaiakis and H. Balakrishnan. Queuing models of airport departure processes for emissions reduction. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2009.

[22] I. Simaiakis and H. Balakrishnan. Impact of congestion on taxi times, fuel burn, and emissions at major airports. *Transportation Research Record: Journal of the Transportation Research Board*, 2184:22–30, 2010.

[23] I. Simaiakis, M. Sandberg, and H. Balakrishnan. Dynamic control of airport departures: Algorithm development and field evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):285–295, 2 2014.