

Scalable Approximate Dynamic Programming Models with Applications in Air
Transportation

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy at George Mason University

By

Poornima Balakrishna
Master of Science
The University of Oklahoma, 2005
Bachelor of Engineering
Madras University, India, 2002

Co-Director: Dr. Rajesh Ganesan, Assistant Professor
Department of Systems Engineering and Operations Research
Co-Director: Dr. Lance Sherry, Associate Professor
Department of Systems Engineering and Operations Research

Summer Semester 2009
George Mason University
Fairfax, VA

Copyright © 2009 by Poornima Balakrishna
All Rights Reserved

Acknowledgments

I am grateful to Dr. Rajesh Ganesan for spending numerous hours discussing this research with me over the past four years. His guidance, availability and accessibility for extended hours, and the opportunities he has provided me have helped me complete this effort in a timely, rewarding, and fulfilling manner. I am also thankful to Dr. Lance Sherry for his insightful ideas, constant encouragement, and enthusiasm. His patience and trust have helped me keep my spirits up through all the challenges. I feel fortunate to have both Dr. Ganesan and Dr. Sherry as my advisors.

It has been a pleasure having Dr. Chun-Hung Chen, Dr. John Shortle, and Dr. Jill Nelson on my committee. Dr. Chen's perspective on simulation-based methods provided me with an appreciation for the many facets of this work. I express my earnest thanks to Dr. Shortle for his ready assistance and spot-on observations that have served to help me discern many aspects of my research. Dr. Nelson's keen interest in my work has helped me gain a better grasp of the fundamental ideas of my research.

I am grateful to Dr. Hillel Kumin, Williams Professor at the University of Oklahoma for supporting me in my decision to pursue an academic career, and for recommending me to the OR program here at George Mason University. I thank Dr. Ariela Sofer for providing me an opportunity to be part of an excellent and invigorating academic environment. Dr. George Donohue's kind encouragement has always motivated me to work harder. I thank Dr. Benjamin Levy at Sensis Corporation, Syracuse, NY for his insights into the operational aspects of the airport departure process, and for the opportunity to visit the air traffic control tower at St. Paul, MN to gain first hand knowledge on ATC procedures. Working with actual flight data has added meaning to this work. I acknowledge the Federal Aviation Administration for their permission to access their Aviation System Performance Metrics database.

I thank my affectionate husband Ravi, for his patient support when I needed to attend to my research for extended hours. By setting an example, my parents have instilled in me a will to take challenges in my stride. Their constant support inspires me to remain passionate about my goals. My mother's tireless enthusiasm for problem-solving and our many online sessions on probability theory helped me take-on the qualifying exam with confidence. My brother Ramachandran, was always ready to lend an ear and help me with any troubleshooting even at odd hours. Support and affection from my grandparents has been a source of great happiness. I thank my family and friends for many cheerful and lighter moments.

Josefine Wiecks and Angel Manzo of SEOR, and Lisa Nolder of IT & E have worked efficiently to help me keep up with the inevitable assortment of time-sensitive paperwork. To every person who has influenced my life in some positive way, I express my grateful thanks.

Table of Contents

	Page
List of Figures	vii
Abstract	xi
1 INTRODUCTION	1
1.1 Research Objective	1
1.2 Background and Motivation: Approximate Dynamic Programming	2
1.2.1 Research Issues in Dynamic Programming	4
1.2.2 State-of-the-art Curse Mitigation Strategies	5
1.2.3 Research Gap	6
1.3 Background and Motivation: Taxi-Out Time Estimation	6
1.3.1 Delays in the Air Transportation System	7
1.3.2 Delay Mitigation Strategies	8
1.3.3 Importance of Estimating Taxi-Out Time	9
1.3.4 Challenges in Predictiong Taxi-Out Time	10
1.3.5 Research Gap	11
1.4 Significance of this Research	12
1.4.1 Significance from the Methodology’s Perspective	12
1.4.2 Significance from the Applications’ Perspective	13
1.5 Structure of the Dissertation	14
2 REVIEW OF PRIOR RESEARCH	15
2.1 Introduction	15
2.2 Literature on Dynamic Programming	15
2.2.1 The Model-Based Controllers	15
2.2.2 Process Model-Free (Dynamic Data Driven) Control	16
2.3 Evolution of the Field of Dynamic Programming	17
2.3.1 Deterministic and Stochastic Dynamic Programming	19
2.3.2 Value Iteration Algorithm for an Infinite Horizon Problem	22
2.3.3 The Need for Approximate Dynamic Programming	22
2.4 Value Function Approximation in Approximate Dynamic Programming	27

2.4.1	Approximation with Aggregation	29
2.4.2	Approximation with Function Fitting	30
2.4.3	Approximation with Interpolation	33
2.5	Case-Study: Taxi-Out Time Estimation	34
2.5.1	The Airport Departure Process	34
2.5.2	Review of Previous Work Related to Taxi-Out Time Prediction . . .	44
3	RESEARCH METHODS	49
3.1	Addressing Scalability in Approximate Dynamic Programming	49
3.1.1	Introduction to Classical Wavelet Theory	50
3.1.2	Diffusion Wavelet Theory	53
3.1.3	Value Function Approximation using Diffusion Wavelets	56
3.1.4	Updating the Value Function: Incorporating the Diffusion Wavelet Scheme in the Iterative ADP algorithm	57
3.2	Case Study: Prediction of Taxi-Out Time using Approximate Dynamic Pro- gramming	59
3.2.1	Data Source	60
3.2.2	Adapting Approximate Dynamic Programming to Taxi-Out Time Es- timation	61
3.2.3	Q-Learning Implementation of the Taxi-Out Time Prediction Problem	65
3.2.4	Advantages of using RL for Taxi-Out Time Estimation	73
3.2.5	Integrating Diffusion Wavelet Based Value Function Approximation and Taxi-out Time Estimation	73
4	ANALYSIS, EXPERIMENTS, AND RESULTS	78
4.1	Airport Analysis	78
4.1.1	Detroit International Airport (DTW)	79
4.2	Taxi-Out Time Estimation Results	87
4.2.1	Experiments and Analysis Based on the Q-Learning Algorithm . . .	89
4.2.2	Results Based on Value Function Approximation	94
4.2.3	Taxi-Out Time Estimation Results Based on a Regression Model . .	98
4.3	Algorithm Extension for John F. Kennedy International Airport (JFK) . .	101
4.4	Study of Algorithm Complexity	107
5	CONCLUSIONS	146
5.1	Methodology	146
5.2	Taxi-Out Time Estimation	147
A	The Robbins-Monro Algorithm	150

B Derivation of Unimpeded Taxi Times	152
Bibliography	154

List of Figures

Figure	Page
1.1 Scope of the dissertation research.	2
1.2 Schematic representation of the airport departure process.	10
2.1 Shortest Path Network.	18
2.2 Difference between ADP and DP.	24
2.3 Phases and Aircraft-ATC communication Links	35
2.4 Departure Process based on ASQP	36
2.5 Summary of observed Flow Constraints and identified causes.	40
3.1 Relation between the scaling and wavelet function spaces.	52
3.2 Flight data for the departure process:ASPM.	61
3.3 Flight data for the arrival process:ASPM.	61
3.4 Factors influencing taxi-out time.	62
3.5 Taxi-out time prediction: state variables.	63
3.6 Reinforcement Learning Functional Block Diagram.	67
3.7 Decay of stepsizes for the taxi-out time estimation algorithm.	70
3.8 Decay of stepsizes for the generalized harmonic sequence.	71
3.9 Convergence properties of stepsizes for the taxi-out time estimation algorithm.	71
3.10 Convergence properties of stepsizes for the generalized harmonic sequence.	72
3.11 Learnt Phase: Q-factor matrix.	72
3.12 Schematic representation of the diffusion wavelet based taxi-out time estimation algorithm.	75
4.1 Airport Diagram (DTW).	80
4.2 Daily standard deviation of taxi-out time (DTW, Mar-Aug).	82
4.3 Daily means of taxi-out time (DTW, Mar-Aug).	83
4.4 Daily standard deviation of taxi-out time (DTW, Sep-Feb).	84
4.5 Daily means of taxi-out time (DTW, Sep-Feb).	85
4.6 Total daily departure demand.	86
4.7 Average hourly demand for pushback.	87

4.8	Average hourly demand for takeoff.	87
4.9	Methods compared for taxi-out time estimation.	88
4.10	Average taxi-out time in 15 minute intervals of the day	91
4.11	Q-learning algorithm (DTW: Individual flights)	91
4.12	Q-learning algorithm (DTW:Averages)	92
4.13	Airport behavior (DTW:December 5, 2007)	95
4.14	Diffusion wavelet based algorithm (DTW: Individual flights)	95
4.15	Q-learning and Diffusion wavelet based algorithm (DTW: Individual flights)	96
4.16	Regression approximation based algorithm (DTW: Individual flights)	96
4.17	Q-learning and value function approximation based methods (DTW: Individual flights)	97
4.18	Taxi-out time estimation based on regression (DTW: Individual flights)	99
4.19	Comparison of prediction accuracy based on different methods (DTW: Individual flights)	100
4.20	Actual taxi-out times and demand, JFK, Dec 4th, 2007	101
4.21	Airport Diagram (JFK).	104
4.22	Prediction accuracy (JFK) within 4 min.	105
4.23	Prediction accuracy based on Q-learning algorithm (JFK: Individual flights)	107
4.24	Q-learning and diffusion wavelet based algorithm (JFK: Individual flights)	108
4.25	Q-learning and value function approximation based algorithms (JFK: Individual flights)	108
4.26	Prediction accuracy based on the four solution methods (JFK: Individual flights)	109
4.27	A comparison of computation times for the learning phase.	111
4.28	A comparison of computation times for the learnt phase.	112
4.29	Daily % prediction accuracy within 4 minutes (DTW, Mar06-Apr06)	113
4.30	Daily % prediction accuracy within 4 minutes (DTW, May06-Jun06)	114
4.31	Daily % prediction accuracy within 4 minutes (DTW, Jul06-Aug06)	115
4.32	Daily % prediction accuracy within 4 minutes (DTW, Mar07-Apr07)	116
4.33	Daily % prediction accuracy within 4 minutes (DTW, May07-Jun07)	117
4.34	Daily % prediction accuracy within 4 minutes (DTW, Jul07-Aug07)	118
4.35	Daily % prediction accuracy within 4 minutes (DTW, Mar08-Apr08)	119

4.36	Daily % prediction accuracy within 4 minutes (DTW, May08-Jun08)	120
4.37	Daily % prediction accuracy within 4 minutes (DTW, Jul08-Aug08)	121
4.38	Daily % prediction accuracy within 4 minutes (DTW, Sep05-Oct05)	122
4.39	Daily % prediction accuracy within 4 minutes (DTW, Nov05-Dec05)	123
4.40	Daily % prediction accuracy within 4 minutes (DTW, Jan06-Feb06)	124
4.41	Daily % prediction accuracy within 4 minutes (DTW, Sep06-Oct06)	125
4.42	Daily % prediction accuracy within 4 minutes (DTW, Nov06-Dec06)	126
4.43	Daily % prediction accuracy within 4 minutes (DTW, Jan07-Feb07)	127
4.44	Daily % prediction accuracy within 4 minutes (DTW, Sep07-Oct07)	128
4.45	Daily % prediction accuracy within 4 minutes (DTW, Nov07-Dec07)	129
4.46	Daily % prediction accuracy within 4 minutes (DTW, Jan08-Feb08)	130
4.47	Airport trends across time of day	131
4.48	Taxi-out time trends (average in 15 min intervals)	131
4.49	Airport trends across time of day	132
4.50	Taxi-out time trends (average in 15 min intervals)	132
4.51	Airport trends across time of day	132
4.52	Taxi-out time trends (average in 15 min intervals)	133
4.53	Airport trends across time of day	133
4.54	Taxi-out time trends (average in 15 min intervals)	133
4.55	Airport trends across time of day	134
4.56	Taxi-out time trends (average in 15 min intervals)	134
4.57	Airport trends across time of day	135
4.58	Taxi-out time trends (average in 15 min intervals)	135
4.59	Airport trends across time of day	135
4.60	Taxi-out time trends (average in 15 min intervals)	136
4.61	Airport trends across time of day	136
4.62	Taxi-out time trends (average in 15 min intervals)	136
4.63	Airport trends across time of day	137
4.64	Taxi-out time trends (average in 15 min intervals)	137
4.65	Airport trends across time of day	137
4.66	Taxi-out time trends (average in 15 min intervals)	138
4.67	Airport trends across time of day	138
4.68	Taxi-out time trends (average in 15 min intervals)	138

4.69	Airport trends across time of day	139
4.70	Taxi-out time trends (average in 15 min intervals)	139
4.71	Daily % actual and prediction taxi-out time statistics (DTW, Mar06-Aug06)	140
4.72	Daily % actual and prediction taxi-out time statistics (DTW, Mar07-Aug07)	141
4.73	Daily % actual and prediction taxi-out time statistics (DTW, Mar08-Aug08)	142
4.74	Daily % actual and prediction taxi-out time statistics (DTW, Sep05-Feb06)	143
4.75	Daily % actual and prediction taxi-out time statistics (DTW, Sep06-Feb07)	144
4.76	Daily % actual and prediction taxi-out time statistics (DTW, Sep07-Feb08)	145

Abstract

SCALABLE APPROXIMATE DYNAMIC PROGRAMMING MODELS WITH APPLICATIONS IN AIR TRANSPORTATION

Poornima Balakrishna, PhD

George Mason University, 2009

Dissertation Co-Director: Dr. Rajesh Ganesan

Dissertation Co-Director: Dr. Lance Sherry

The research objective of the dissertation is to develop methods to address the curse of dimensionality in the field of approximate dynamic programming, to enhance the scalability of these methods to large-scale problems.

Several problems, including those faced in day to day life involve sequential decision making in the presence of uncertainty. These problems can often be modeled as Markov decision processes using the Bellman's optimality equation. Attempts to solve even reasonably complex problems through stochastic dynamic programming are faced with the curse of modeling and the curse of dimensionality. The curse of modeling has been addressed in the literature through the introduction of reinforcement learning strategies, a strand of approximate dynamic programming (ADP). In spite of considerable research efforts, curse of dimensionality which affects the scalability of ADP for large scale applications still remains a challenge. In this research, a value function approximation method based on the theory of diffusion wavelets is investigated to address the scalability of ADP methods.

The first contribution of this dissertation is an advancement of the state-of-the-art in the field of stochastic dynamic programming methods that are solved using ADP approaches.

An important intellectual merit is the innovatively designed diffusion wavelet based value function approximation method which is integrated with ADP to address the curse of dimensionality. The innovation lies in this integration that exploits the structure of the problem to achieve computational feasibility.

The ADP method with diffusion wavelet based value function approximation is tested on the problem of taxi-out time estimation of aircrafts (time duration between gate-pushback and wheels-off) to establish a proof of concept for the research objective. The second contribution of this dissertation is the modeling of the taxi-out time estimation of flights as a stochastic dynamic programming problem with the capability to provide sequential predictions in real-time as the system evolves. The model aims to accurately predict the taxi-out time of a flight at least fifteen minutes before its scheduled gate pushback time. As a case study for Detroit International Airport, results indicate that there is a 6 % to 12 % increase in the percentage of flights predicted accurately (with a root mean square error of two minutes) using ADP when compared with a regression model for taxi-out time predictions.

The outcomes of this dissertation research provide a generic methodology for sequential decision making under uncertainty in large scale applications by uniting concepts from signal processing, statistics, stochastic processes, and artificial intelligence, which may provide solutions for future automated decision making in large scale complex applications in other engineering domains.

Chapter 1: INTRODUCTION

1.1 Research Objective

The research objective of the dissertation is to develop methods to address the “curse of dimensionality” in the field of approximate dynamic programming, to enhance the scalability of these methods to large-scale problems.

The goals of this dissertation are,

1. To investigate a new value function approximation method based on diffusion wavelet theory to mitigate the “curse of dimensionality” in approximate dynamic programming (ADP) approaches.
2. To cast the problem of taxi-out time estimation of an aircraft as a stochastic dynamic programming model, and solve it using an ADP approach. The model aims to predict taxi-out time of a flight at least 15 minutes in advance of its scheduled gate pushback time with a root mean square error of 2 minutes.
3. To test the diffusion wavelet based value function approximation method for ADP approaches on the taxi-out time estimation problem, and examine the performance with state-of-the-art ADP methods to establish a proof of concept for the research objective.

The scope of this dissertation research is illustrated in Figure 1.1 and is explained in the sections that follow.

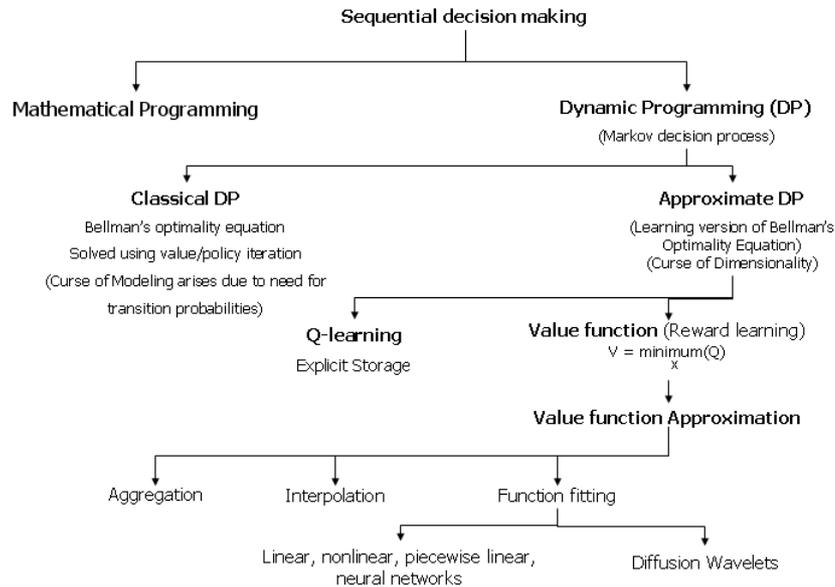


Figure 1.1: Scope of the dissertation research.

1.2 Background and Motivation: Approximate Dynamic Programming

Dynamic programming is a collection of mathematical tools for solving sequential decision making problems, often in the presence of uncertainty. Conventional approaches to solve multi-period decision problems such as dynamic resource allocation, machine replacement or repair policy problems and dynamic inventory control using mathematical programming require that the problem be formulated and solved by simultaneously considering constraints from all stages [Bellman, 2003], [Denardo, 2003]. These methods can be used to model single-period decision making problems or multi-period (sequential) decision making problems that are of finite horizon although they are often more complex for multi-period decision making. Additionally, they are not suitable for handling infinite horizon decision making problems. In contrast, the dynamic programming approach models the multi-stage decision problem through a recursive formulation in which the computation involves only two successive stages at a time. Another limitation of mathematical programming for stochastic problems is that the probability distributions of the random parameters are assumed to be

given and cannot depend on the decisions taken. In many instances, such as an inventory control problem, the decisions made in a given period influence how the system evolves in the next time period. Dynamic programming approaches may be applied to these otherwise intractable problems. However, the viability of dynamic programming is still limited by computational issues such as the “curse of dimensionality” which is addressed in this research.

The dynamic programming approach proceeds by defining state variables, which captures the evolution of the process over time. Problems that are modeled with continuous states and decisions (and typically, in continuous time) are addressed as “control theory”, while problems with discrete states and decisions, modeled in discrete time, are studied as Markov decision processes (MDP). This dissertation focuses on Markov decision processes. Typically, states evolve according to transition probabilities that are governed by decisions made (or actions taken) at each state. More formally, if we are in state s_i and take action x , we will move to state s_j , with probability $p(s_i, x, s_j)$. The set of allowable actions depends on the state s_i , and is a result of the “constraints” of the problem.

The idea behind dynamic programming is to relate the value of being in a particular state at a specific point in time with the value of the states visited in future points in time [Powell, 2007]. Dynamic programming allows for a tradeoff between immediate and future costs, thus avoiding myopic solutions to the problem. The decisions are ranked based on the sum of two terms - the expected contribution (cost or reward) of the present period given that you are in state s_i and take action x , $C(s_i, x, s_j)$, and the optimal expected contribution of all subsequent periods, $V^*(\cdot)$. Mathematically, this is given by Equation 1.1.

$$V^*(s_i) = \min_{x \in X} E[C(s_i, x, s_j) + V^*(s_j)|s_i, x] \quad \forall s_i \in S \quad (1.1)$$

where $V^*(s_i)$ is the optimal cumulative discounted cost of being in state s_i , S is the set of all possible states, X is the action space, and $E[\cdot|s_i, x]$ denotes expected value with respect

to s_j , given s_i and x [Bertsekas and Tsitsiklis, 1996].

The collection of decisions made at each of the states constitutes a “policy”. The objective is to optimize the total contribution over a number of stages with respect to all allowable policies (or decision vectors/variables).

1.2.1 Research Issues in Dynamic Programming

“Curse of Dimensionality”

Equation 1.1 is known by several names, including Bellman’s equation, Hamilton-Jacobi-Bellman (HJB) equation, functional equation, and optimality equation. The optimality equation is solved through iterative algorithms such as value and policy iteration. These solution strategies require looping over all possible states at each iteration (called synchronous updation of the value of being in each state), which becomes computationally intensive when the number of states (and actions to be compared) is significantly large. This is known in the literature as the “curse of dimensionality” [Powell, 2007].

“Curse of Modeling”

As seen, Equation 1.1 assumes that the value of being in particular states at future points in time is known. The equation is thus solved through backward recursion. In addition, to compute the expectation in Equation 1.1, knowledge of the transition probabilities between states under each given policy becomes necessary. Often, even for only reasonably complex problems these probabilities are extremely difficult, if not impossible, to compute. As an example, for the airport departure process, it is difficult to find an explicit probabilistic model that governs the transitions between different states of congestion on the ground. This is referred to as the “curse of modeling”.

1.2.2 State-of-the-art Curse Mitigation Strategies

Mitigation of “Curse of Modeling”

The curse of modeling has been alleviated by the introduction of reinforcement learning (RL) (sometimes also called neuro-dynamic programming [Bertsekas and Tsitsiklis, 1996]), which is a strand of approximate dynamic programming (ADP). ADP steps forward in time and uses asynchronous updation of system states, in which the value function of only one state is updated in each iteration through a learning version of Bellman’s equation. Rather than compute the transition probabilities for all possible state-action combinations; at each iteration, one sample path of the uncertain information is simulated (say, using Monte Carlo simulation). This approach is referred to in the literature as simulation-based optimization [Gosavi, 2003]. This is explained further in Section 2.3.3.

Mitigation of “Curse of Dimensionality”

Despite asynchronous updation in ADP, the “curse of dimensionality” arises due to the large number of system states and the need to store the value function for each state. For example, even for a small problem with 5 state variables, each of which can take on 10 possible values, there are 10^5 state combinations.

The “curse of dimensionality” is typically addressed by introducing an approximation to the value function. The choice of architecture for the value function approximation has become a problem of considerable interest amongst several research communities including mathematicians, computer scientists, control engineers, and Operations Research analysts (The online repository at [UMass, 2009] provides a vast listing of research efforts in this field). The statistical problem of function fitting is rendered more challenging in the context of value function approximation (refer Section 2.4.2 for a more detailed description of the challenges). Research is ongoing to find efficient ways to approximate the value function while still ensuring a near-optimal solution.

1.2.3 Research Gap

Current methods for value function approximation use traditional regression with fixed parameters which are updated at every iteration. These methods require the type and the number of basis functions to be pre-specified, which becomes difficult because the shape of the value function is not known in advance. This affects the quality of value function approximation which in turn affects the decisions made and the convergence of the value functions. The problem of determining the best basis functions for a given application becomes more complex with an increase in the dimension of the system state space thus justifying the need for constant research efforts. A new method for efficient and accurate value function approximation based on diffusion wavelet theory is developed and tested in this research.

1.3 Background and Motivation: Taxi-Out Time Estimation

In general, large-scale dynamic systems (with large system state space dimensions), such as the air transportation system, require dynamic prediction to support sequential decision making for optimizing operations. Such systems are highly stochastic, complex, adaptive, and networked [Donohue and Shaver III, 2008]. They cannot be analytically modeled within reasonable accuracy due to the high complexity and non-linear interactions between their subsystems and there exist no closed form solutions [Callaham et al., 2001], [Donohue and Shaver III, 2008]. Their evolving properties can be studied through their interactions with the environment. A dynamic prediction system to support decision making can be achieved via the design and analysis of learning-based (dynamic data-driven) intelligent prediction systems that are solved using ADP methods. These stochastic approximation methods can solve certain non-linear dynamic systems, particularly those with fewer variables and fewer discretizations of the variable range, within reasonable accuracy. However, there are limitations to ADP's viability in large-scale systems due to the "curse of dimensionality", which is an open research issue addressed in this dissertation. This justifies the need for

new research that advances the science of ADP approaches, particularly by enhancing the scalability and practicality of ADP solutions for accurate prediction, which forms the core of this research.

1.3.1 Delays in the Air Transportation System

The United States National Airspace System (NAS) is one of the most complex networked systems ever built [Donohue and Shaver III, 2008]. The major components include the administration, control centers, airports, airlines, aircrafts, and passengers. The complexity of the NAS poses many challenges for its efficient management and control. One of the challenges includes regulating and managing reliability. It is necessary for all stakeholders (the FAA, airlines, passengers, and the ATC) to stay informed, understand the causes, and find solutions to predict and mitigate any delays.

Delays are caused due to increase in demand beyond capacity and near capacity operations. Excessive delays may result in cancellations and increased passenger complaints. The U.S. Government Accountability Office's report [Fleming, 2008] and the Bureau of Transportation Statistics report [Goldburg and Chesser, 2008] suggest a deterioration in the performance of the NAS in terms of flight delays over the last two years, which are worse than pre-9/11 years. The total cost of flight delays in 2007 was estimated to be \$41B [JPDO, 2008].

Delays propagate throughout the system and they are stochastic and elastic in nature [Xu et al., 2008]. The uncertainty in arrival and departure demand and the uncertainty of the impact of wind and visibility on capacity result in the stochasticity of delays. For example, determining the scope and duration of the ground delay programs (GDP¹) requires managing these uncertainties. The elastic behavior is due to the fact that delay may be adjusted (positively or negatively) by flying speed, taking alternate routes, turnaround time on the ground, aircraft swapping by airlines [Beatty et al., 1999] and position in the

¹A GDP is said to be in effect when an aircraft is held at the gate of the origin airport due to capacity limits at the destination airport.

departure clearance queue especially during busy hours of the airport.

Total delay of a flight segment from its origin to destination comprises of gate-out delay, taxi-out delay, airborne delay, and taxi-in delay. Among these delay elements, historical data indicates that taxi-out time contributes to over 20% of the total delay [Schumer and Maloney, 2008]. Since departure delays propagate across the NAS, it is imperative to improve the efficiency of airport operations by minimizing taxi-out delay.

1.3.2 Delay Mitigation Strategies

Delays at the airport can be addressed by scheduling operations within capacity limits and improvements in airspace capacity management. Operational efficiency may be improved via better traffic flow (demand) management both on ground (by minimizing taxi-out time delay) and in the airspace around the airport while concurrently achieving total coordination between ground and air movement at the airport. This problem is particularly complex and severely compounded in a metroplex where multiple large airports are in close proximity having shared airspace, which requires simultaneous departures from these airports, arrivals over the ocean to minimize noise, and specific runway configurations such that departures from one airport do not head into the arrivals at another, e.g. New York and Washington airports.

The mid-term implementation plan for the NextGen ATS includes delay mitigation strategies for different phases of flight [NextGen, 2009]. The plan for the taxi, pushback and departure phase include:

1. New tools for improving situational awareness of the position of aircrafts and other vehicles on the ground.
2. Implementation of multiple precise departure paths from each runway end which will allow for more efficient separation of departing aircrafts, aid in circumnavigating thunderstorms and other severe weather phenomena near the airport vicinity, and allow each airport to operate more independently.

3. Wind monitoring systems that will determine when wake vortices from departing aircraft have sufficiently dissipated, which will allow for simultaneous operations on closely spaced parallel runways. An increase in capacity can be realized by airports from their existing runways. Additional runways may also be built without expanding physical boundaries and with minimal cost and environmental impacts.

1.3.3 Importance of Estimating Taxi-Out Time

The need for accurate planning tools to improve airport operational efficiency has become more pressing than ever from the perspective of many stakeholders. The airlines are faced with the prospect of higher fuel costs which is the second largest operating cost (Air Transportation Association, 2009). Over-subscription of the national airspace capacity has resulted in progressively increased use of delay management programs, hitherto used to compensate for poor weather conditions [Donohue and Shaver III, 2008]. Airlines and airport operators are faced with the eventuality of more stringent emissions compliance regulations (National Ambient Air Quality Standards), which may be met through reduction in aircraft delays on the airport surface. Additionally, the large uncertainty in current taxi-out time predictions may result in a departure push-back that is followed by a return to the gate for re-fueling (Dalton personal communication, July 2008) thus compounding the initial delay. Yet, without an accurate taxi-out time prediction for departures there is no way to effectively manage fuel consumption, emissions, or cost, nor will the inputs to the flow control modeling be accurate enough to allow rational imposition of delay programs. In the future, it can be expected that accurate predictions of ‘wheels-off’ time may be used in determining whether an aircraft can meet its allocated slot time, thereby fitting into an en-route traffic flow.

Taxi-out time prediction is the first step to managing taxi-out delays. Taxi-out time of a flight is defined as the duration between gate pushback and takeoff (wheels -off). A schematic representation of the departure process is shown in Figure 1.2. To minimize taxi-out delay, it is necessary to accurately predict taxi-out time under dynamic airport

conditions. This information will allow the airlines to dynamically adjust departures, minimizing congestion and allowing for smoother airport operations by avoiding situations when demand (departure rates) nears or exceeds airport capacity. Taxi-out time predictions will serve as key inputs for the implementation of the NextGen ATS focused on flow contingency management and capacity management at airports [NextGen, 2009], and will benefit the implementation of a holistic Total Airport Management system [Meier and Eriksen, 2006]. As an example of a future concept of automating airport control towers and Terminal Radar Approach Control (TRACON) operations, it will be necessary to predict airport dynamics such as taxi-out times and feedback this information for aiding the artificial intelligence-based departure time decision making process at the airport operational level.

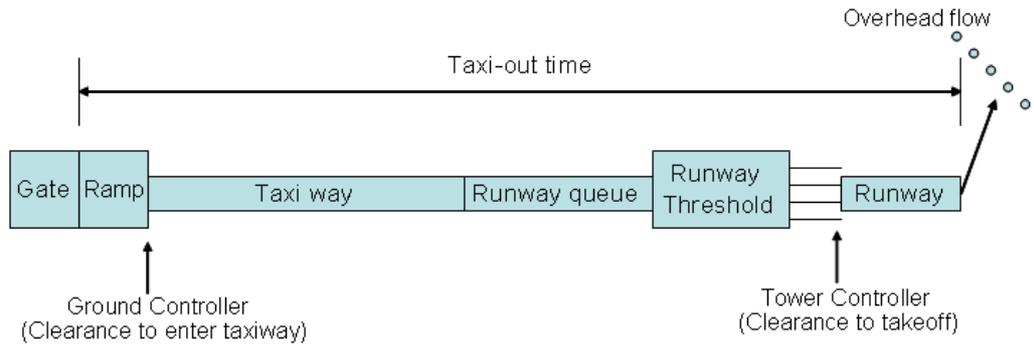


Figure 1.2: Schematic representation of the airport departure process.

1.3.4 Challenges in Predictiong Taxi-Out Time

The taxi-out phase of the departure process exhibits a high level of stochastic behavior since it is affected by factors such as arriving flights (which are given higher preference over departures), congestion on the airport surface, ground clearance process to leave the gate, position in the queue for tower clearance to takeoff, multiple departures scheduled at the same time (airlines do not coordinate departure time with one another), unscheduled departures (a delayed departure aircraft may join the queue with other scheduled departures),

capacity restrictions determined by weather, visibility, ceiling and wind shear, de-icing operations for aircrafts in winter months, time of day, taxiway configuration, and aircraft size (influences time between take-offs due to wake-vortices, which in turn affects time spent on taxiways) .

1.3.5 Research Gap

The surface traffic management decision tool described in the NextGen concept of operations [NextGen, 2009] will provide a “look-ahead” feature for conflict predictions between aircrafts on the surface and in the airspace surrounding the airport. The Integrated Departure/Arrival Capability (IDAC) will also increase automation and decision support for coordination of shared and congested NAS resources. These tools are expected to account for uncertainty on the surface and enroute airspace.

This research on taxi-out time estimation is thus an important step toward achieving the total coordination between ground and air movement at airports that is described in the roadmap for the Next Generation Air Transportation System (NGATS) by the U.S. Government’s Joint Program Development Office [JPDO, 2008]. This roadmap also includes automation of some of the routine lower-level functions currently performed by Air Traffic Controllers (ground and tower). Clearly, the Federal Aviation Administration (FAA) is looking towards a modernized Air Traffic Control (ATC) system with increased automation. This requires sequential prediction capabilities in real time as the system evolves. In addition, the prediction method must ideally be adaptive to changing airport dynamics. The technology to accurately predict the sequencing and spacing of departures in taxi-out is not yet available today. Continuous taxi-out time prediction is essential for airlines to make downstream adjustments to schedules, and for airport controllers to adjust gate departures to manage congestion on the ground. This calls for technological breakthroughs in new methods that can predict taxi-out time accurately, assist in real-time decision making to mitigate taxi-out delays, and support the future automation concepts.

This dissertation research is focused on investigating practically viable ADP-based

stochastic models for advancing the science of dynamic prediction in large-scale (large-dimension) systems such as the air transportation system. These models are tested on an air transportation application for their capability in capturing the airport dynamics (congestion) to predict the taxi-out time sequentially in real-time for a given schedule of operations.

1.4 Significance of this Research

1.4.1 Significance from the Methodology's Perspective

Scalability in approximate dynamic programming: This research advances the state of knowledge in the field of approximate dynamic programming (ADP). In specific, this research investigates a model to enhance scalability of ADP algorithms through addressing the “curse of dimensionality”. The method uses principles of diffusion wavelet theory [Coifman and Maggioni, 2004] for value function approximation.

1. The dissertation research that integrates ADP with diffusion wavelet-based value function approximation describes a generalized solution approach for prediction and optimization problems that may be modeled as stochastic dynamic programming problems. It is hypothesized that this method will boost the viability of ADP models to other engineering domains (e.g. manufacturing enterprises, power systems).
 - (a) Due to the dynamic nature of the problem the shape of the value function is seldom known in advance. While existing methods to approximate the value function requires one to specify a choice of basis functions, the diffusion wavelet based approach identifies the number and geometry of the basis functions that best fits the data.
 - (b) The diffusion wavelet procedure can effectively compute basis functions given multidimensional data.
 - (c) Approximate dynamic programming is a learning based strategy. A state that

is not visited in the training phase may be visited in the test phase, for which an optimal decision has not been determined. An efficient interpolation scheme is an inherent property of the diffusion wavelet approach, which also eliminates the need to address the interpolation problem separately.

2. The “curse of dimensionality” has limited the use of ADP models for large scale applications. It is hypothesized that the properties of the diffusion wavelet adaptation to value function approximation as described above will improve the quality of the ADP solution for complex large-scale problems which will provide a higher accuracy than by using simplified analytical models that do not capture the entire dynamics of the large-scale systems.

1.4.2 Significance from the Applications’ Perspective

Taxi-out time estimation: An adaptive learning-based approach (i.e., ADP) is applied, to model the problem of taxi-out time estimation [Balakrishna et al., 2008b], [Balakrishna et al., 2008a], [Balakrishna et al., 2008c]. To date, there is very little research done on taxi-out time predictions made 15-45 minutes in advance of scheduled gate pushback [Shumsky, 1997], [Signor and Levy, 2006], [Futer, 2006], which is a critical aspect for departure management and planning. This dissertation research is the first time that a method integrating state-of-the-art stochastic dynamic programming methods and ADP principles is used to predict taxi-out times. This approach has several transformative benefits as stated below:

1. The method provides an effective framework for prediction in large-scale problems which are stochastic, complex, adaptive, and networked in nature, such as the air transportation application addressed in this research.
2. The estimation model is solved using ADP methods in which adaptive learning takes place by interacting with the environment and through a reward updation scheme. This is in contrast to neural networks based on supervised learning which relies on good training data that consists of desired outputs.

3. Dynamic systems under uncertainty such as the operation of an airport evolve over time and have to be predicted periodically. This naturally fits the problem of sequential prediction of taxi-out time which can be aptly modeled by stochastic dynamic programming.
4. A further advantage of the learning-based stochastic dynamic programming method is that it is data-driven and built on the principles of Markov decision processes. Hence, an explicit mathematical model such as a differential equation or regression model of the dynamic system is not required for prediction.
5. The integration of the diffusion wavelet based value function approximation method with the field of air transportation is especially pertinent as the dimension of problems in this area is typically large and involves a significant amount of uncertainty.

1.5 Structure of the Dissertation

This dissertation is organized as follows. In Chapter 2, a discussion of the state-of-the-art in dynamic programming, and approximate dynamic programming is introduced, followed by a description of the airport departure process which is analyzed in the context of taxi-out time prediction. Currently existing methods for taxi-out time prediction are also discussed. Chapter 3 presents the research methods used to achieve the goals of this dissertation. Analysis and results are detailed in Chapter 4. Chapter 5 summarizes the conclusions and future work.

Chapter 2: REVIEW OF PRIOR RESEARCH

2.1 Introduction

The sequential decision making problem may be viewed as a stochastic control problem. In this chapter a brief comparison between model-based and model-free controllers is discussed and a historical motivation for pursuing the model-free approach is presented. The evolution of the field of dynamic programming (DP) as well as the motivation for approximate dynamic programming (ADP) is described, along with literature on currently available methods to approximate the value function.

In order to introduce the problem of taxi-out time estimation, a detailed description of the airport departure process based on field observations conducted at Boston Logan International Airport by [Idris and Hansman, 2000] is provided. This is followed by a discussion of previous research attempts related to taxi-out time estimation.

2.2 Literature on Dynamic Programming

Typically, control theory is classified into optimal, robust, and adaptive control. The literature reviewed here pertains to the model-based and model-free classification of control theory.

2.2.1 The Model-Based Controllers

These controllers use two main types of models: differential-algebraic equations and difference equations. The differential-algebraic equation approach has been used for both linear and linear-quadratic optimal control [Wonham, 1979], and in control of non-linear systems [Martin et al., 2001]. Robust control for non-linear systems has been addressed by [Baras

and Patel, 1995] which in turn reduces to finding a solution to the Hamilton-Jacobi-Bellman equation. In recent years, linear and non-linear hybrid dynamical system has been the focus of research [Schaft and Schumacher, 2000] and [Matveev and Savkin, 2000]. The most popular form of control using difference equations is the Run-by-Run (RbR) controller in which the control laws are obtained from designed experiments and/or regression models. Some of the RbR methods are given in [Castillo and Hurwitz, 1997]. Model-based simulation techniques have been used for the control of discrete-event dynamic systems but lack closed form solutions [Ho et al., 1992]. Some primary limitations of model-based controllers are that 1) they depend on good process models, 2) control actions are based on the parameters of the model which are often fixed, 3) they cannot handle large perturbations of the system because the system is not intelligent, 4) they need multiple filtering steps to compensate for drifts and autocorrelation, and 5) they lack scalability. One of the ways to handle these limitations is through a process model-free (data-driven) learning-based control, which is a simulation-based optimization technique.

2.2.2 Process Model-Free (Dynamic Data Driven) Control

Model-free control systems use some form of artificial intelligence such as neural networks, fuzzy-logic rules, and machine learning. They have strong mathematical foundations underlying their construction. Some of these systems, particularly neural networks and fuzzy-logic rules, though are claimed to be model-free, do contain certain hidden or implicit models and make strong modeling assumptions when it comes to proving the stability of the controller [Ahmed and Anjum, 1997].

Optimal Control and approximate dynamic programming (ADP) stochastic approximation methods have been proven to be effective for control of non-linear dynamic systems. In this method the controller is constructed using a function approximator (FA). However, it is not possible for a model-free framework to obtain the derivatives necessary to implement standard gradient-based search techniques (such as back-propagation) for estimating the unknown parameters of the FA. Usually, such algorithms for control applications rely

on well-known finite-difference stochastic approximations (FDSA) to the gradient [Bayard, 1991]. The FDSA approach can be very costly in terms of the number of system measurements required, especially in high-dimensional problems for estimating the parameters of the FA vector. This led to the development of Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms for FA that are based only on measurements of the system that operates in closed-loop [Spall and Cristion, 1998]. Among the several variants and applications of SPSA, the implementation of SPSA in simulation-based optimization using Reinforcement Learning (RL) offers several advantages in solving many stochastic dynamic sequential decision-making problems of which the stochastic control problem is a subset [Gosavi, 2003]. RL (a strand of ADP) is a method for solving Markov decision processes (MDP), which is rooted in the Bellman [Bellman, 1954] equation and uses the principle of stochastic approximation (e.g. Robbins-Monro method [Robbins and Monro, 1951]). [Howard, 1960] first showed how the optimal policy for a MDP may be obtained by iteratively solving the linear system of Bellman equations. Textbook treatment of this topic and convergent average reward RL algorithms can be found in [Bertsekas and Tsitsiklis, 1996], [Gosavi, 1998], and [Ganesan et al., 2007]. The connection between various control theories and ADP is available in [Werbos, 1998].

2.3 Evolution of the Field of Dynamic Programming

In order to discuss how the field of dynamic programming has evolved, a brief introduction to the concepts of Markov chains and Markov decision processes is required. Consider a system with m finite set of states denoted by $S = \{s_1, s_2, \dots, s_m\}$ (note that each state can be multidimensional by way of number of attributes). Due to the uncertainties in how the system evolves, the process moves from state s_i to state s_j with probability p_{s_i, s_j} , which is known as the transition probability. The process can be viewed as a directed graph where each node represents a particular state.

The process is called a Markov chain, if the immediate future state depends on what

the current state is, and not on the past. Markov chains are observed at unit time steps apart. Mathematically, this Markov property means, the,

$$Pr(S_{t+1} = s_j | S_t = s_i, \dots, S_1 = s_1) = Pr(S_{t+1} = s_j | S_t = s_i) \quad (2.1)$$

Now, in a Markov *decision* process (MDP), a decision say x , has to be made in each state. The decision made (or the action taken) influences the transition between states, and the transition probabilities represent the probability of moving from state s_i to state s_j under action x . The transition probabilities are denoted by $p(s_i, x, s_j)$. The objective of solving an MDP is to find the best action to be taken in every state, that is, to find the optimal *policy* R , which minimizes the cost or maximizes the reward (often the *expected* cost or reward). Here, a *policy* is a mapping from every state to an action.

The sequential decision making problem is best explained by considering the *shortest path problem*. For simplicity, we consider a deterministic case. Consider the network (directed graph) with nine nodes shown in Figure 2.1 (refer [Denardo, 2003]). The weights on the edge connecting node i and node j represents the time t_{ij} , that is taken to travel from i to j . At each node, a decision has to be made as to which node to travel to in the next time step such that the total travel time from node 1 to node 9 is minimized.

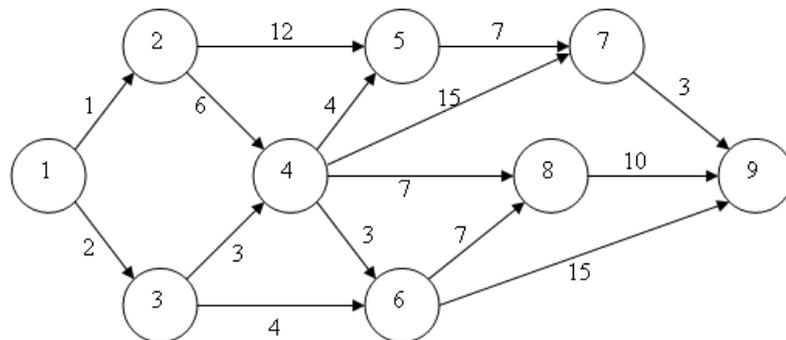


Figure 2.1: Shortest Path Network.

To solve this problem via dynamic programming, a value function f_i is defined, to denote

the minimum travel time from node i to node 9. A model based on backward recursion is established, as shown in Equation 2.2.

$$f_i = \min_j \{t_{ij} + f_j\}, \quad i \neq 9, i < j \quad (2.2)$$

For $i = 9$, $f_9 = 0$. One must ensure that the condition $i < j$ is satisfied when labeling the nodes, and there is exactly one way in which this can be done. The solutions to the Equation set 2.2 and $f_9 = 0$ provides the optimal value of f_i for each i . To obtain the optimal path from node 1 to node 9, a forward trace is implemented, starting at node 1. At each node i that is visited, the edge moving forward which provided the minimum value of f_i is picked.

2.3.1 Deterministic and Stochastic Dynamic Programming

The discussion on dynamic programming (DP) and the evolution of approximate dynamic programming (ADP) presented in [Powell, 2007] is well motivated and insightful. It is useful to understand some of these concepts.

Consider the problem of optimizing an expected cost over a time horizon of T time periods. The form of the related objective function is given by Equation 2.3.

$$\min_{\Pi} E \left\{ \sum_{t=0}^T \gamma^t C_t^{\Pi}(S_t, x_t^{\Pi}(S_t)) \right\} \quad (2.3)$$

where, γ is the discount parameter, or the time value of money, S_t is the system state at time t , Π is the set of all actions, and C_t is the contribution or cost function.

Further, it is assumed that a *transition function* given by Equation 2.4 is available. The transition function is a physical model that relates the state S_{t+1} visited at future time $t + 1$, with the state S_t and action x_t associated with current time period t .

$$S_{t+1} = S^M(S_t, x_t) \quad (2.4)$$

Letting $C_t(S_t, x_t)$ be the contribution or reward generated from being in state S_t , and taking action x_t , the value of being in a particular state at one point in time can be expressed recursively in terms of the value of the states that you are carried into at the next point in time. In the deterministic case this is given by Equation 2.5, and is often called the Bellman's equation or the optimality equation for the deterministic case. Note that to solve Equation 2.5, the value of being in a state at a future point in time, that is, time $t + 1$, is required. Hence, a backward recursion procedure is necessary.

$$V_t(S_t) = \min_{x_t \in X_t} (C_t(S_t, x_t) + \gamma V_{t+1}(S_{t+1})) \quad \forall S_t \quad (2.5)$$

$$= C_t(S_t, x_t^*(S_t) + \gamma V_{t+1}(S_{t+1}(S_t, x_t^*(S_t)))) \quad (2.6)$$

where, S_{t+1} is determined using the transition function defined by Equation 2.4, and X_t is the set of all actions allowed in state S_t . Note here that given S_t , and x_t , S_{t+1} is deterministic. Equation 2.6 follows since, the value of being in state S_t is the value of using optimal decision $x_t^*(S_t)$.

However, it is often the case that new information becomes available *after* decision x_t is made. This is easily demonstrated by considering the case of a simple inventory problem. Suppose that S_t is the inventory at end of time t , and x_t units are ordered. Now, a demand \hat{W}_{t+1} arises during the time period $(t, t + 1)$. Observe that this demand is random and unknown at time t . Therefore, an uncertainty arises in both the contribution earned, and in the determination of the next state S_{t+1} that is visited.

The stochastic recursive equation may then be expressed as in Equation 2.7.

$$V_t(S_t) = \min_{x_t} \left\{ \hat{C}_{t+1}(S_t, x_t, \hat{W}_{t+1}) + \gamma V_{t+1}(S_{t+1}) | S_t \right\} \quad (2.7)$$

where, $V_{t+1}(S_{t+1}) | S_t$ is the *expected* value of the state that you end up at.

Letting,

$$C_t(S_t, x_t) = E \left\{ \hat{C}_{t+1}(S_t, x_t, W_{t+1}) | S_t \right\} \quad (2.8)$$

Then, Equation 2.7 may be expressed using the *Expectation form of Bellman's Equation*, which is given by Equation 2.9.

$$V_t(S_t) = \min_{x_t \in X_t} (C_t(S_t, x_t) + \gamma E \{ V_{t+1}(S_{t+1}(S_t, x_t)) | S_t \}) \quad (2.9)$$

and, Equation 2.10 below represents the equivalent *Standard form of Bellman's Equation*.

$$V_t(S_t) = \min_{x_t \in X_t} (C_t(S_t, x_t) + \gamma \sum_{s_j \in S} P(S_{t+1} = s_j | S_t, x_t) \cdot V_{t+1}(s_j)) \quad (2.10)$$

While the deterministic case of Bellman's optimality equation can be solved through backward recursion, the same solution method cannot be adopted in the stochastic case. This is because the computation of the expectation in either Equation 2.9 or Equation 2.10 uses the fact that $\sum_{s_j \in S} P(S_{t+1} = s_j | S_t = s_i) = 1$. This is a forward transition probability.

This constraint on the sum of probabilities does not hold while stepping backward in time. Hence, to solve the stochastic dynamic programming formulation, we must consider the backward equation, but solve it forward in time. In classical DP, this is accomplished by adopting iterative solution methods (one such method is discussed in Section 2.3.2).

Many times, we are interested in determining a policy over an infinite horizon. In this case, the parameters of the contribution function, transition function, and the process governing the exogenous information, do not vary over time. Therefore, we let $t \rightarrow \infty$, and $V(S) = \lim_{t \rightarrow \infty} V_t(S_t)$. Assuming that the limit exists, the steady state optimality equations are obtained as

$$V(s_i) = \min_{x \in X} \{ C(s_i, x) + \gamma \sum_{s_j \in S} P(s_j | s_i, x) V(s_j) \} \quad (2.11)$$

It has been shown [Powell, 2007] that, solving Equation 2.11 is equivalent to solving for the original objective function given by Equation 2.3 (with T set to ∞).

2.3.2 Value Iteration Algorithm for an Infinite Horizon Problem

The optimality equation given by Equation 2.11, can be implemented using a value iteration scheme. Rather than using a subscript t that is decremented from T back to 0 (representing stages in the finite horizon problem), we use an iteration counter, n , that starts at 0, and increases, until we satisfy a convergence criterion. The basic structure of the value iteration algorithm is summarized in Algorithm 1.

Algorithm 1. *Step 0: Initialization, Set $v^0(s_i) = 0 \quad \forall s_i \in S$ Fix a tolerance parameter $\epsilon > 0$ Set $n = 1$*

Step 1: For each $s_i \in S$ compute:

$$v^n(s_i) = \min_{x \in X} (C(s_i, x) + \gamma \sum_{s_j \in S} P(s_j | s_i, x) v^{n-1}(s_j)) \quad (2.12)$$

and let x^n be the decision vector that solves Equation 2.12.

Step 2: If $\|v^n - v^{n-1}\| < \frac{\epsilon(1-\gamma)}{2\gamma}$, let Π^ϵ , be the resulting policy that solves Equation 2.12, and let $v^\epsilon = v^n$, and stop; else set $n = n + 1$ and goto Step 1.

This value iteration scheme can be proven to converge to the optimal solution, and an exact solution to the problem is obtained [Powell, 2007].

2.3.3 The Need for Approximate Dynamic Programming

There are two significant disadvantages related to the solution approach stated above,

1. Firstly, these models are often intractable. For instance, a formal model of the information process (such as for the demand in the inventory problem), or the transition function, may not be known. In this case, we cannot compute the expectation required

in say, Equation 2.12. This is formally referred to in the literature as the “curse of modeling”.

2. Secondly, the value iteration algorithm described in Section 2.3.2, requires us to loop over all possible states at each iteration (this is known as *synchronous* updation). This algorithm is difficult to implement when the state space is very large.

Approximate dynamic programming (ADP) methods have been developed in an attempt to address these issues. Earlier, we saw that classical DP steps backward in time. In contrast, the foundation of ADP is based on an algorithm strategy that steps *forward* through time. However, two new challenges arise -

1. A way to randomly generate a sample of what *might* happen (in terms of the random information) is needed.
2. In stepping forward through time, the value of being in a state at a future point in time is unknown. A way to make decisions at each iteration is hence needed.

It is based on the subtlety of point (2.) above that two alternative algorithms to solve ADP models are developed (refer Algorithm 3 and Algorithm 4 and the discussion that follows soon after).

Figure 2.2 summarizes the important aspects in which ADP differs from classical DP [Gosavi, 2003].

Value Function Based ADP

To address the first challenge, a random sample realization of the uncertain information is picked, possibly through a Monte Carlo simulation technique. Since we do not know the value function ahead of time, ADP proceeds by approximating a value function $\bar{V}_t(S_t)$ iteratively. Since this is an approximation, it can take on any functional form.¹

¹It is important to note here that, the approximation of the value function that the ADP algorithm incorporates is different from the value function approximation scheme that we propose to address in this dissertation. In some sense, the value function approximation procedure that we are concerned with, takes place “outside” the ADP algorithm.

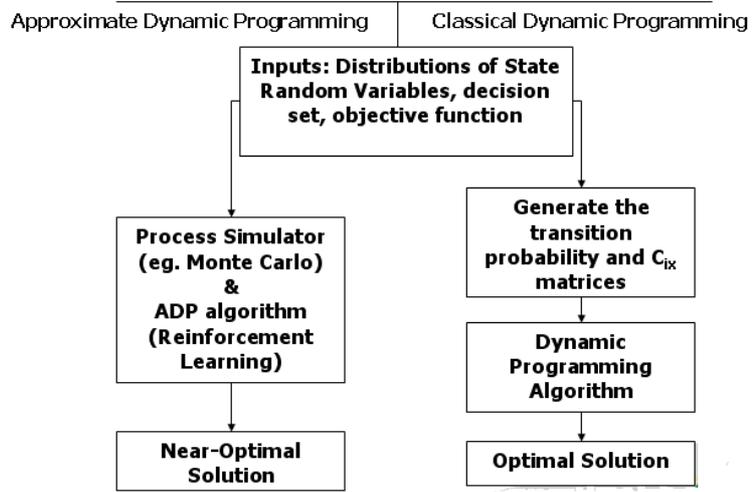


Figure 2.2: Difference between ADP and DP.

Now, assuming that this approximation to the value function is available to us, we have a way to make decisions, as explained in Algorithm 2.

Algorithm 2. *Step 0: Initialization. Step 0a: Initialize $\bar{V}_t^0(S_t)$ for all states S_t Step 0b: Choose an initial state S_0^1 Step 0c: Set $n = 1$*
Step 1: Choose a sample path ω^n Step 2: For $t = 0, 1, 2, \dots, T$ do:
Step 2a: Solve,

$$\hat{v}_t^n = \min_{x_t \in X_t^n} (C_t(S_t^n, x_t) + \gamma \sum_{S_{t+1} \in S} P(S_{t+1} | S_t^n, x_t) \bar{V}_{t+1}^{n-1}(S_{t+1})) \quad (2.13)$$

and let x_t^n be the value of x_t that solves the minimization problem represented by Equation 2.13.

Step 2b: Update \bar{V}_t^{n-1} using

$$\bar{V}_t^n(S_t) = \begin{cases} \hat{v}_t^n & \text{if } S_t = S_t^n \\ \bar{V}_t^{n-1} & \text{otherwise} \end{cases} \quad (2.14)$$

Step 2c: Compute

$$S_{t+1}^n = S^M(S_t^n, x_t^n, W_{t+1}(\omega^n)) \quad (2.15)$$

Step 3: Let $n = n + 1$, and, if $n < N$ goto Step 1.

Note that this algorithm alleviates the need to loop over all states, which was a concern using classical DP (refer beginning of Section 2.3.3). The method of updation described in Algorithm 2 may hence be characterized as asynchronous. However, the “curse of modeling” is not yet resolved. This is addressed through the introduction of reinforcement learning (RL), which is a strand of ADP.

Reinforcement Learning

An intuitive analysis of the concepts of reinforcement learning (RL) is available in [Gosavi, 2003], and the material presented in this section is suitably adapted from this text.

The ADP algorithm described earlier still requires the computation of an expectation (Equation 2.13). RL introduces a stochastic approximation scheme to approximate this expectation. The approximation of the expectation is based on the Robbins-Monro method [Robbins and Monro, 1951]. The simple illustration of the Robbins-Monro algorithm is provided in Appendix A.

In order to make use of the Robbins-Monro approximation to the expectation, the concept of Q-factors is introduced, so that,

$$Q(s_i, x) = \sum_{s_j \in S} P(s_j | s_i, x) \left[C(s_i, x) + \gamma \min_{b \in X} Q(s_j, b) \right] \quad (2.16)$$

$$= E \left[C(s_i, x) + \gamma \min_{b \in X} Q(s_j, b) \right] \quad (2.17)$$

$$= E [\text{SAMPLE}] \quad (2.18)$$

where, Equation 2.18 is written since, the term inside the expectation of Equation 2.17 is a random variable.

Applying the Robbins-Monro approximation [Robbins and Monro, 1951], a learning version of Bellman’s optimality equation (Q-Learning) is obtained, and is given by Equation 2.19. The equation is represented in terms of the infinite horizon problem.

$$Q^n(S^n, x^n) = (1 - \alpha^n)Q^{n-1}(S^n, x^n) + \alpha^n \left[C(S^n, x^n) + \gamma \min_{b \in X} Q^{n-1}(S^{n+1}, b) \right] \quad (2.19)$$

where n is the iteration number, and α is the decay parameter (or learning rate). The decay rate of α is critical to the accuracy of the approximation (it is known as the stepsize in learning literature). Note that Equation 2.19 does not require the probabilities to be known. The “curse of modeling” associated with classical dynamic programming is thus eliminated by this approach.

The relationship between the value function and the Q-factor is written as,

$$V^{n-1}(S^n) = \min_{x \in X} Q^{n-1}(S^n, x^n) \quad (2.20)$$

Thus, Q-Learning proceeds by considering (S, x) to be an augmented state. The algorithm for implementing the Q-Learning approach over N iterations is adapted from [Powell, 2007] and summarized in Algorithm 3 .

Algorithm 3. *Step 0: Initialization.*

Step 0a: Initialize the approximation for the value function $Q^0(s, x), \forall s \in S, x \in X$.

Step 0b: Set $n = 0$

Step 0c: Initialize S^0

Step 1: Choose sample path ω^n

Step 2a: Obtain the decision to be taken, using current Q-factors:

$$x^n = \arg \min_{x \in X} Q^{n-1}(S^n, x) \quad (2.21)$$

Step 2b: Compute

$$\hat{q}^n = C(S^n, x^n) + \gamma V^{n-1}(S^M(S^n, x^n, \omega^n)) \quad (2.22)$$

where $V^{n-1}(S^{n+1})$ is given by Equation 2.20.

Step 2c: Update the values of Q^{n-1} and V^{n-1} using

$$Q^n(S^n, x^n) = (1 - \alpha^n)Q^{n-1}(S^n, x^n) + \alpha^n \hat{q}^n \quad (2.23)$$

$$V^n(S^n) = \min_{x \in X} Q^n(S^n, x^n) \quad (2.24)$$

Step 2d: Find the next state $S^{n+1} = S^M(S^n, x^n, \omega^n)$

Step 3: $n = n + 1$. If $n < N$ go to Step 1.

2.4 Value Function Approximation in Approximate Dynamic Programming

The Q-Learning algorithm described in Section 2.3.3 alleviates the “curse of modeling” in that the transition probabilities are no longer required to be computed explicitly. However, this algorithm provides a lookup-table representation of the value function, and is not very efficient for storage and computation. This is especially a concern when dealing with problems of large scale. Previous attempts have been made to solve this “curse of dimensionality” through approximating the value function.

First, Equation 2.19 and Equation 2.20 are combined to obtain a value function based

learning scheme. Here, the elements of the value function vector $V(S)$, are learnt at each iteration, rather than the Q-factors. This is represented by Equation 2.25.

$$V^{n+1}(S^n) = (1 - \alpha^{n+1})V^n(S^n) + \alpha^{n+1} \min_{x \in X} \{C(S^n, x) + \gamma V^n(S^{n+1})\} \quad (2.25)$$

So, rather than store the Q-factors for each state-action pair (S, x) , the value function for each state, $V(S)$, is stored, along with only the corresponding best action at each iteration, which is determined by Equation 2.26.

$$x^* = \arg \min_{x \in X} \{C(S^n, x) + \gamma V^n(S^{n+1})\} \quad (2.26)$$

Algorithm 4 discusses the iterative updation procedure (over N iterations) for the values of $V(S)$.

Algorithm 4. *Step 0: Initialization.*

Step 0a: Initialize $V(S^0)$, S^0 , and set $n = 0$.

Step 1: Obtain a sample path ω^n

Step 2a: Update S^n if $n > 0$.

Step 2b: Solve

$$\hat{v}^n = \min_{x \in X} (C(S^n, x) + \gamma V^{n-1}(S^M(S^n, x, \omega^n))) \quad (2.27)$$

and let x^n be the value of x that solves Equation 2.27.

Step 2c: Update the value function

$$V^n(S^n) = (1 - \alpha^n)V^{n-1}(S^n) + \alpha^n \hat{v}^n \quad (2.28)$$

Step 2d: Find the next state, $S^{n+1} = S^M(S^n, x^n, \omega^n)$

Step 3: $n = n + 1$. If $n < N$, go to Step 1.

A comparison of Algorithm 3 and Algorithm 4 indicates two key differences in the implementation. First, Algorithm 3 does not require looping over all actions in X at each iteration. However, this also means that the algorithm requires more iterations to converge as opposed to Algorithm 4. In addition, it is noted that the Q-learning approach described in Algorithm 3 first computes the decision x^n , and then proceeds by updating the Q-factor value based on the simulated sample path ω^n . Whereas, the learning scheme detailed in Algorithm 4 first samples the external information (the uncertainty given by ω^n), and then solves for the best action (x^n) that is to be taken at a particular iteration (refer Equation 2.27).

For large scale problems, the Q-learning approach in Algorithm 3 is often computationally inefficient since the method requires iterative updation of a two dimensional matrix of size that equals the number of state-action combinations. In contrast, the value function matrix based on Algorithm 4 is a vector of single dimensional values of length equal to the number of possible states. However, for complex systems with a large number of state variables, this strategy is still not viable. This “curse of dimensionality” is often addressed through function approximation, where the value function vector is approximated as $V(S) \approx \bar{V}(S)$. The approximation scheme is then incorporated in Algorithm 4.

Some value function approximation methods that have been studied in the literature are discussed in the following subsections. Broadly, the function approximation methods in the context of ADP may be categorized into three groups [Gosavi, 2003].

1. State Space Aggregation,
2. Function Fitting, and,
3. Function Interpolation

2.4.1 Approximation with Aggregation

The method of aggregation of the state space in conjunction with ADP has been used by researchers for several years. The idea is to aggregate the state space, solve the resulting

problem exactly, and then disaggregate the state space to obtain an approximate solution to the original problem. Interestingly, a feature of ADP is that state space aggregation is only used to approximate the value function, and does not result in a simplification of the state space during the evolution of the process, which is given by the transition function

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \quad (2.29)$$

The idea behind state space aggregation is to arrive upon an aggregation function $G(\cdot)$, which maps the detailed state vector into something simpler. The function G may disregard some attributes of the state vector, and may lead to a coarse discretization of some states. The simpler version of the state vector is then used to approximate the value function, as, $\bar{V}(G(s))$.

Aggregation has been known to perform well in certain empirical studies. Other useful details on the method of aggregation may be obtained from [Powell, 2007] and [Gosavi, 2003].

The method of aggregation presents some disadvantages,

1. Two entirely different states, $S'_1 \neq S'_2$, may have the same value function, $\bar{V}(S'_1) = \bar{V}(S'_2)$. It may happen then, that we make the same decision in both states though the difference between states S'_1 and S'_2 may call for two different actions.
2. Aggregation does not allow one to take advantage of structure in the state variable.
3. It is usually difficult to preserve the Markov property when states are combined. However, even if the Markov property is not lost, optimality of the solution cannot be guaranteed. This makes the approach vastly heuristic.

2.4.2 Approximation with Function Fitting

Function fitting in the context of ADP generally involves regression or a neural networks based strategy. The general form of value function approximation based on regression

may be expressed as $\bar{V}(S/\theta) = \sum_{f \in F} \theta_f \rho_f(S)$, where ρ_f are the basis functions, and θ_f are the parameters (or coefficients). There are several popular choices for the basis functions, ρ_f . These include linear basis functions, orthogonal polynomials, Fourier approximations, splines, or piecewise linear regression.

Challenges in Function Fitting

1. A significant challenge for function approximation in ADP is that the value functions change during every iteration, and we need to update the function fit at each iteration. This means that the scalars associated with the function must also be constantly updated. Traditional methods that involve solving systems of linear equations or solving a nonlinear programming problem are too slow in the context of DP. Incremental regression or incremental neural networks proves ineffective since they require that all the data samples must be related to the *same* function [Gosavi, 2003]. This condition is usually not met in ADP applications since the value function keeps changing across iterations. A recursive scheme to update the parameter θ_f , may be adopted. Methods for recursive updation, involving the least squares method and time series estimation have been proposed in the literature [Powell, 2007].

One possible approach is to keep the stepsize α relatively small, to ensure that the value function changes slowly.

2. “Spill-over” effect – at each iteration, a single value of the value function, pertaining to the current state, is updated. However, a function approximation may affect value functions at other states as well. Slow decay of the learning rate α is hence essential to maintain accuracy. This “spill-over” effect is best mitigated by dividing the state space into grids, and using independent function approximation schemes in each grid. This limits the “spill-over” to the set of states comprised in each grid.
3. The shape of the value function is never known beforehand. When the value functions

are nonlinear, backpropagation using nonlinear neural networks is a recommended approach. However, a drawback of backpropagation is the possibility of getting trapped in local optima. A piecewise linear approach for ρ_f , is often the best guess.

4. It is difficult to prove convergence of the ADP algorithm when it is coupled with a function approximation scheme. In practice, one must ensure that the approximated value function does not deviate significantly from what would have been obtained using the look-up table representation.

Experimental studies in the literature indicate that function approximation based on regression or neural networks often do not work well in the context of ADP [Gosavi, 2003]. This area is identified as one that needs considerable research efforts.

Approximation with Diffusion Wavelets and Laplacian Eigenfunctions

An attempt has been made in [Mahadevan and Maggioni, 2005] to build a foundation for analyzing Markov Decision Processes through what may be called Representation Policy Iteration (RPI), where the underlying basis functions and policies are simultaneously learned. Laplacian eigenfunctions and the diffusion wavelet transform (DWT) have been considered as methods to decompose the state space. Once the state space has been modeled as a finite undirected graph (G, E, W) , the degree of node x is represented in terms of the weight on edge $x \sim y$, given by the following relationship in Equation 2.30.

$$d(x) = \sum_{x \sim y} w(x, y) \tag{2.30}$$

Further, a matrix D is defined by $D_{xx} = d(x)$, and W , as $W_{xy} = w(x, y) = w(y, x)$.

Now, if f is any function; the Laplacian approximation of f is given by Equation 2.31.

$$Lf(x) = (D - W)f \tag{2.31}$$

The Laplacian approximation however, is limited in its ability to capture deviations from

global smoothness. The idea of approximating functions using Laplacian eigenfunctions is hence extended, by constructing the diffusion wavelet transform ². A sparse *diffusion operator* H , is computed based on spectral graph theory, and the approximation of the function f is then given by $(I - H)^{-1}f$.

Simple experiments based on considering 630 states demonstrate the superiority of the diffusion wavelet scheme over the Laplacian approach. In addition, the performance of the two methods was compared on a RPI problem with 50 states and 2 actions, while considering 5000 steps in the Markov chain. Further, the two methods were contrasted with two manually coded basis functions: the polynomial, and the radial-basis function (RBF) approximators. Results suggested that the automatically learned Laplacian and diffusion wavelet basis functions provided a more stable performance than the polynomial and RBF approaches. Between the Laplacian method and the DWT approach, a trade-off was observed with respect to speed of convergence and the accuracy of the solution. While the diffusion wavelet approach converged faster, the accuracy with Laplacian basis functions was higher. Also, as the number of diffusion basis functions are increased, convergence is slower, but accuracy is greater. Another conclusion is that unlike Laplacian eigenfunctions, as the number of basis functions is increased, the policy error obtained using diffusion wavelets does not decrease monotonically.

While the conclusions suggest that the diffusion wavelet approach needs to be researched further, its ability to approximate functions that are not globally smooth, remains appealing.

2.4.3 Approximation with Interpolation

Interpolation is known to be a robust method for function approximation in RL. The idea is to store a smaller set of representative value functions, and determine all others based on interpolation. Two methods, the *k-nearest-neighbors* and *kernel-based*, are described in [Gosavi, 2003]. In the *k-nearest-neighbors* approach, the interpolation is based on an Euclidean or rectilinear distance between the n-dimensional state vectors. When the value

²The structure of diffusion wavelet transforms is described in Section 3.1.2.

of being in a particular state is queried, the values of its k nearest neighbors may be averaged, or a regression may be performed, to obtain the desired value function. The *kernel-based* method assigns a weight to each of the representative value functions. To obtain the value of being at a state that is queried, the weighted average of the known value functions is computed.

These interpolation methods however, are computationally inefficient because they require (i) storage of the representative value functions, and (ii) evaluation of summations over a large number of quantities during the averaging process. One way to address this difficulty is to store the representative value functions as a “binary tree”, which is a data structure that helps in efficient storage and retrieval of data.

2.5 Case-Study: Taxi-Out Time Estimation

2.5.1 The Airport Departure Process

An overview of the flight departure process at an airport is available in [Idris and Hansman, 2000]. A gist of their discussion is given below since it proves insightful in understanding the factors that influence taxi-out times, and the complexity of the airport dynamics.

The authors view the airport as a node in the National Airspace System (NAS) network. It acts as a sink, receiving arrivals at the Airport Acceptance Rate (AAR), and as a source, supplying departure traffic at the Airport Departure Rate (ADR). The main components of the airport system at a higher level of detail can be identified as the terminal airspace (which includes entry and exit fixes for arriving and departing flights respectively), runways, taxiways, ramps and gates. Arriving and departing flights often share these resources resulting in possible interactions. The Air Traffic Control (ATC) Tower coordinates and controls the arrival and departure process and is an integral part of the airport system.

The departure process is considered as a collection of “Operational phases”. Each of these phases is associated with specific components of the airport system. The major operational phases are depicted in the lower part of Figure 2.3.

The departure flow is a controlled process, requiring communication between pilots and air traffic controllers. Before each operation can be performed a “clearance” communication is required from the ATC. Each operational phase is associated with an air traffic control position. At many airports the gate phase and ramp phase are controlled and managed by the airlines. The communication links between the controllers and aircraft at various stages of departure and the communication links amongst the controllers is shown in Figure 2.3. Just after take-off the local controller hands over control of the departing aircraft to the TRACON (Terminal Radar Control) who is responsible for the terminal airspace.

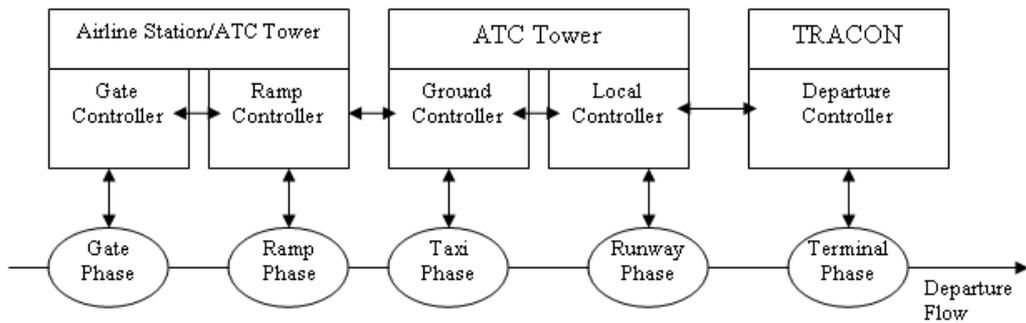


Figure 2.3: Phases and Aircraft-ATC communication Links

Method of Analysis

The authors present a highly detailed case study of the departure process at Boston Logan International Airport. The overall objective is to identify flow constraints and inefficiencies in the airport departure process and to gain insight into the underlying causes of, and interactions amongst, these flow constraints. These observations will potentially aid in the development of a decision support tool that could enable reduction in delays and mitigation of congestion. Data from various sources were consolidated to enable an in-depth analysis of the departure process dynamics. The data were classified into two functional categories, (a) Aircraft movement data and (b) Airport conditions data. ‘Aircraft movement data’ was used to identify flow constraints with respect to resources (such as runway, ATC work load, and communication channel capacity), and to analyze the underlying queuing dynamics.

It is noted here that the queuing dynamics is used to represent a physical model of the system rather than to serve any statistical purpose. Causality of the flow constraints were identified by further considering 'airport conditions data'. Data were collected during peak operations to observe queuing behavior when demand is high and congestion increases. A period of 200 hours spread over two years (1998-1999) was considered. 'Aircraft movement data' is obtained both from historical data available from the airlines and the FAA, and through field observations at Boston Logan International Airport. Historical data are obtained from ACARS (Airline Carrier Automated Reporting System), pilot delay reports, ASQP (Airline System for Quality Performance), and CODAS (Consolidated Operations and Delay Analysis System) which includes data from both ASQP and ETMS (Enhanced Traffic Management System) traffic counts. It is highlighted that for a study of this nature, ASQP data provides limited observability of the departure process since the dynamics between the pushback and takeoff events are aggregated into one phase (Refer Figure 2.4).

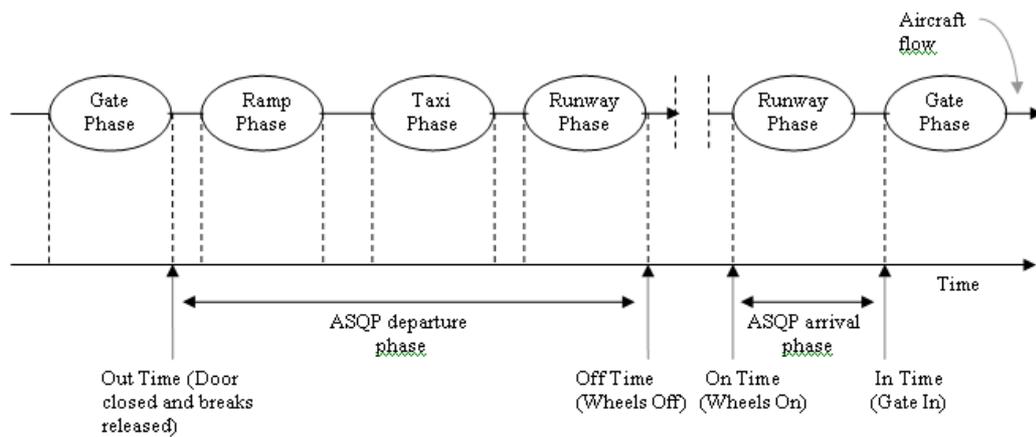


Figure 2.4: Departure Process based on ASQP

'Aircraft movement data' from field observations are obtained from communications between the air traffic controllers and the pilots, and from the 'flight progress strips' that details information starting from scheduled departure time of a flight and follows up with updates on the position of the flight as the aircraft is handed down from ground controller

to local controller and finally to the TRACON. Amongst other information, runway assignment and exit fix are recorded on the ‘flight progress strip’. Five communication events signal the transition between the four operational phases that were depicted in Figure 2.3. These communication signals are - (1) the pilot’s “ready for pushback” call and four control instructions - (2) ‘clear for pushback’ for jets (propellers do not undergo this procedure), (3) ‘clear for taxi’, (4) ‘monitor tower’ (this is a request for a frequency change when the control of the departing aircraft is handed over from ground controller to local controller, and (5) ‘clear for take off’. Of these, (1) to (4) are handled by the ground controller (note that the limited available area at BOS does not provide for the presence of ramp controllers), and (5) by the local controller. When communication traffic became heavy and communications were repeated, the last communication was used to indicate event time.

The singular aspect of this approach is that the dynamics of the departure process are translated into dynamics of the communication process; and these are then used to analyze the flow constraints and queuing dynamics of the four different operational phases! ‘Airport conditions data’ are obtained from Air Traffic Control Tower logs. The Traffic Management Coordinator (TMC) log records information on current weather, current runway assignments, imposed ATC restrictions and equipment failures; along with the cause (called ‘impacting conditions’). Restriction logs also maintained by the TMC record details of imposed departure restrictions such as type of restriction, its origin (in terms of downstream location), start time, end time, and ‘impacting conditions’. In addition, insightful information was obtained from ATC manuals. The physical model of the departure process and analysis of the dynamics were based on queuing and controller behavior which was observed through focused observations. These observations are considered as a manifestation of flow constraints and controller strategies. Main airport resources and different types of queues are identified. Each type of queue identifies a possible flow constraint (for instance, pile up of flight strips indicates flow restrictions due to ATC workload). Queue formation was observed through the ATC tower window and through communications between pilots and controllers, as well as through the accumulation of ‘flight progress strips’. The dynamics

of the departure process is affected by decisions and strategies made and employed by the controllers in the presence of flow constraints. Strategies used by controllers were identified through elicitation; by monitoring the behavior of controllers, and through interviews with Control Tower supervisors and air traffic managers - common patterns in controller behavior were identified (including strategies for sequencing aircraft for take-off, common control points where aircraft are held for critical sequencing decisions, and deviations from the first-come-first-served (FCFS) rule in accordance with ATC procedures). Detailed behavior and the reasons for specific controller behavior choices became visible following interviews with ATC supervisors and managers. In addition, more clarity was obtained regarding higher level decisions. Bias in data collection was reduced by observing several controllers and by monitoring different runway configurations and weather conditions.

Strengths of the Approach

1. In the absence of detailed track data, the proposed approach provides sufficient data to gain powerful insight into the airport departure process dynamics - in particular, potential flow constraints are identified and causes are analyzed.
2. Air traffic controllers currently employ established procedures in their decision making process. An in-depth study of this kind which is based on monitoring ATC personnel would highlight what operational changes may or may not be acceptable to the air traffic management.
3. The approach is a mix of varying data collection methods (such as voice communications, human inter-loop, and visual observations-queue formation on the surface, accumulation of flight strips). However the different data have been brought together in a coherent, meaningful, and convincing fashion.

Weaknesses of the Approach

1. The primary weakness of the approach is in the obvious tediousness of data collection and integration. It can thus not be easily extended to other airports.

2. Boston Logan International Airport operates under a single runway configuration and does not incorporate ramp controllers. Queuing dynamics at other airports may thus be more complicated perhaps with increased communication delays.
3. Data is collected for typical days of operation to study nominal airport behavior. Inclement weather (not infrequent at BOS) operations are not considered. ATC response on such days may be very different.

The few weaknesses of the approach are in a way all related to the tediousness of the data collection approach which discourages model extension. There is however no doubt that the presented approach is a highly creditable effort that provides crucial knowledge of departure operation dynamics.

Conclusions about Processes and Constraints that Determine the Performance of Airport Flows

The airport departure process was identified as a complex interactive queueing system with controlled blocking; the control actions being issued by air traffic controllers. Aircraft queues were considered as manifestations of flow constraints and were hence used to analyze potential constraints caused by contention for airport resources (terminal airspace, runways, taxiways, ramp, gates, air traffic controllers, communication channels). Causes behind the flow constraints affecting different operational phases are also established, and how they are managed by the ATC is discussed. Results and conclusions presented are specific to Boston Logan International Airport but the ideas are intuitively generalized to other airports.

At the strategic level it is stated that selection of runway configuration to a large extent drives the queuing dynamics. This process takes into account factors such as runway availability, weather, demand matching, and noise mitigation. The rest of the constraints are at the tactical level and are identified by observing aircraft movement on the ground. It was determined that irrespective of runway configuration, the type of queues and nature of interactions was similar. The queues are controlled, since the pilot has to wait for a “clearance” from ATC when he is ready to perform an operation. Hence the time an aircraft

spends in queue is caused by waiting for the controlled resource as well as by waiting for ATC, and for the communication channel. The Table 2.5 below summarizes the operational phases (or resources under contention) affected by flow constraints and their corresponding causal factors.

Operational Phase/Resource that experience Flow Constraints	Identified Causes for Flow Constraints
Runways	Wake vortex and runway separation requirements
	Noise
	Runway Crossing
	Landing Aircraft
	Sequencing Strategies
	ATC Workload
	Runway change
Downstream resources such as exit fixes and flight routes (affects almost all phases through propagation)	Aircraft Preparation
	Traffic flow management based on destination and local conditions, to regulate demand
Gates	Gate sharing by arrivals and departures
	Limited gate capacity
	High uncertainty and lack of observability in gate operations
	Management of gate operations
	Gate hold
	Gate interdependence
	Ground controller workload
Taxiways/Ramp	Limited ramp and taxiway capacity
	Routing and sequencing strategies
	Ground controller workload

Figure 2.5: Summary of observed Flow Constraints and identified causes.

Records of communication between pilot and controllers suggested that queues formed mainly at the runway when compared to any other phase. A sample of 8 rush hours for a specific runway configuration showed that the mean time spent in the runway phase was 9 : 40 minutes as opposed to 4 : 28 minutes in the taxi phase, 3 : 48 minutes in the ramp phase, and 2 : 38 minutes in the pre-pushback phase.

Some of the factors that created inefficiencies in the runway system and reduced runway throughput are discussed along with controller reaction to mitigate effects of these factors

whenever possible. Ensuring safe operations (especially in bad weather) requires a minimum wake vortex and runway separation. Successive flights taking off on diverging routes would ease separation requirements, but due to noise abatement policies jets have to follow certain noise mitigation routes and this reduces throughput unless proper sequencing is adopted. At times arriving aircrafts may have to cross active departure runways. When taxiway capacity becomes full, runway departures will be impeded to allow arrivals to flow. In such situations group runway crossings and simultaneous crossings at multiple points are the intuitively appealing decisions adopted by controllers to reduce the magnitude of departure disruptions. Landings are often given priorities over departures for safety reasons. At times a landing aircraft does not stop short of an intersection with a departure runway. The mean time in the runway phase for aircraft that departed from a runway intersecting with an arrival runway was observed to be about 45% higher than when there was no such intersection. Controller sequencing strategies determines runway system efficiency. Ground controllers preferred to alternate jets and props so that downstream local controllers could easily achieve larger separations and reduce their workload. When flight progress strips pile up at the ATC, pilots experience a delay in receipt of communication. It often happens, that gate controllers are asked to hold aircrafts at the gate when ground controllers are overwhelmed by heavy traffic. Loss of efficiency is also incurred during the course of a runway change. Alternate runways are used during the transition to reduce loss of efficiency. Delays are also incurred due to aircraft preparation for take-off (such as weight and balance calculations, and cabin checks). This holds up the rest of the queue.

Downstream resources such as the terminal airspace and exit fixes tend to have a finite buffer capacity. Hence when blocking is resorted to, throughput is reduced. The effects propagate and affect outbound flow from the airport. These delays are absorbed anywhere on the airport surface depending on the traffic flow management strategy employed. Initiatives are taken to regulate the demand and keep it within capacity. Causes for imposing downstream restrictions include inclement weather, high demand, equipment outages, runway non-availability, and controller workload. The restrictions can affect flights in one

of three possible ways - (1) a take-off time window is specified, such as an EDCT (Expected Departure Clearance Time) for GDP (Ground Delay Program) flights or through a Departure Sequencing Program (DSP), (2) take-off spacing is specified either as distance (Miles-In-Trail of MIT) or as time (Minutes-In-Trail MINIT), (3) time delay, through a GDP (long term) or a Ground Stop (GS) which is short term. Local restrictions (predominantly through Ground Stops) were found to have more influence on the throughput (especially when the number of Ground Stops was large) as opposed to destination related restrictions. Destination restrictions (as well as local restrictions) affected the taxi-out and pushback phases of individual flights. An aircraft's taxi-out (or pushback) was supposed to be influenced by a restriction if the duration of the restriction overlapped with its taxi-out (or pushback). DSP was found to affect taxi-out time but not considerably. EDCT and local in-train restrictions were absorbed at the gate. Delays also increased as a function of number of restricted aircraft.

Flow constraint manifestations at the gate were not found to be as significant as those caused by the previous two factors. Several causal factors were however identified. Sharing of gates between departures and arrivals and limited gate capacity resulted in several arrival flights finding gates occupied by departure aircrafts. This is aggravated when aircrafts absorb GDP departure delays at the gate or experience mechanical problems. There are also gate and aircraft type compatibility issues to contend with. Ground controllers are usually in the dark about turnaround operations at the gate prior to the pilot's call for "clearance to pushback". This introduces an uncertainty regarding the pushback time as far as the ATC is concerned. Their decisions up to the point when the pilot calls are thus made using often inaccurate scheduled departure times. Aircrafts in the same gate alley that request pushback are cleared in a first-come-first-served order. Work load for Ground Controllers increases when resolving gate-occupied situations. This leaves them with less time to manage other phases of departure thus introducing further delays.

Boston Logan International Airport has limited ramp and taxiway capacity. In addition arrivals and departures often share the same taxiway. This resulted in some queue formation

at ramps and taxiways. However these delays were small with respect to the delays at the runway phase. Routing and sequencing strategies based on priorities, or for reducing work load of local controllers was a causal factor for flights to be placed on hold on the taxiway.

Delay Mitigation Strategies and Insights

Conclusions and suggested methods to improve the overall departure flow process are put forth. A proactive as opposed to the currently employed reactive approach to runway configuration selection is recommended. Based on previous work the authors suggest that it is possible to reduce taxi-out times (and thus incur environmental benefits) by regulating flow while keeping throughput the same. Gate capacity is however identified to be an issue for such an implementation. Reduction in queue sizes and delays, it is believed, would require increased flexibility of gate operations. Given the skill of air traffic controllers, it is unlikely that a significant increase in runway efficiency can be achieved without the addition of resources. Currently ground controllers help relieve local controllers' workloads by ensuring convenient sequencing of aircrafts. Automation of these processes would be highly desirable especially since it was observed that lack of area did not allow sequence re-ordering close to the take-off queue. It is suggested that appropriate take-off orders be computed in the presence of downstream restrictions. Average wait times calculated through the developed queuing model could be used for this purpose. Decision support tools should incorporate time control in order to efficiently process time windows specified by EDCT and DSP. Information flow is critical in estimating departure demand and enabling optimal sequencing. Programs such as Collaborative Decision Making (CDM) must be exploited for maximum benefits. An integrated automation tool is thus required to assist in proper decision making. Behavior at hub airports may be significantly different and needs to be studied.

2.5.2 Review of Previous Work Related to Taxi-Out Time Prediction

Many recent studies have proposed different methods to predict taxi-out time and then use the prediction to minimize the taxi-out times.

Previous studies related to departure flow planning include both simulation models and analytical formulations. The DEPARTS [Cooper et al., 2001] model developed as a prototype for Atlanta International Airport (ATL), by Mitre Corporation, attempts to reduce taxi times by generating optimal runway assignments, departure sequencing and departure fix loading. The objective for near-term departure scheduling is to minimize the average taxi-out time over the next 10 to 30 minutes, to get flights into the air from the airport as early as possible without causing downstream traffic congestion in the terminal or en route air space. The input to this decision support tool is a near-real time airport information system and specifically takes advantage of the independent parallel runway operations for arrivals and departures at ATL, and the availability of flight status data (through the Airport Resource Management Tool (ARMT) from the Atlanta Surface Movement Advisor (SMA), the ATL Automated Radar Terminal System (ARTS), and manual scanning of bar coded flight strips at the ATL air traffic control tower. ARMT also captures traffic flow management constraints, airport configuration, and the current weather conditions. Results of their analysis indicate that benefits derived from DEPARTS increase during days with higher actual taxi times, and also that pushback predictability could influence all phases of flight and traffic flow management.

A simulation based study of queueing dynamics and “traffic rules” is reported in [Carr et al., 2002]. The authors conclude that flow-rate restrictions significantly impact departure traffic. The impact of downstream restrictions is measured by considering aggregate metrics such as airport throughput, departure congestion, and average taxi-out delay. A “mesoscopic” airport queueing model is developed in [Pujet et al., 1999]. The Airline Service Quality Performance (ASQP) data collected by the Department of Transportation, and that from the Preferential Runway Assignment system (PRAS) database were used for the purpose of model validation. The model is used to evaluate preliminary control strategies,

and the impact of these schemes on operating costs, environmental costs, and overall delay is presented.

Other research that develops a departure planning tool for departure time prediction is available in [Barrer et al., 1989], [Idris et al., 1998], [Anagnostakis et al., 2000], [Shumsky, 1997], and [Lindsay et al., 2005].

Direct predictions attempting to minimize taxi-out delays using accurate surface surveillance data have been presented to literature [Clow et al., 2004] and [Welch et al., 2001]. Signor and Levy [Signor and Levy, 2006] discuss the implications of accurate OOOI (Gate Out, Runway Off, Runway On, Gate In) data for efficient resource utilization. A pair-wise comparison between Aircraft Situation Data to Industry (ASDI), OOOI data provided by Northwest Airlines (NWA) (Flight Event Data Source: FEDS), and Multi-dependent static surveillance (MDS) data was conducted. It was found that potentially, the surface surveillance radar track data provide independent and most accurate OOOI event times that is also more generally accessible in comparison to proprietary data (from airlines for example). In addition, a bivariate quadratic polynomial regression equation was developed for taxi-out time forecasting. The data used was from the Sensis Corporation's Airport Surface Detection Equipment - Model X (ASDE-X) system at Detroit International Airport (DTW). It is concluded that a standard error of 2 minutes can be achieved for departures up to 10 minutes prior to the aircraft leaving the ramp area. A methodology to estimate total and excess taxi-times from the ASDE-X system has been developed at Sensis Corporation [Levy and Legge, 2008]. Data is currently available for 13 major airports in the United States. Metrics related to environmental factors, such as total and excess fuel burn, fuel cost, and emissions are also estimated. Details of the algorithms used to extract OOOI events and excess taxi times from the surveillance data are presented. Estimated metrics may be reported on a per-airport, per-aircraft, or per-carrier basis.

Algorithms adapted to taxi-time prediction such as space time network search which uses Dijkstra's algorithm and event based A* and co-evolution based genetic algorithm have been compared in [Brinton et al., 2002].

A decision support system, called the Surface Management System (SMS) has been developed at Metron Aviation. The system is used to model demand and delays on and around the airport surface. The airport surface is ‘broken-down’ into several intersections and links. At every intersection, the shortest path to the next allowable node is computed and stored, based on distances and anticipated speeds of aircrafts. A surface trajectory, including a runway assignment is then predicted and updated, based on the current position of the aircraft. The tool also considers the ‘undelayed’ taxi-out time to compute the earliest estimated takeoff time for a flight. This computation assumes that there is no congestion on the ground (note ³). The undelayed times are then used to determine the amount of holding and queuing a flight should experience for proper sequencing and spacing of aircrafts. For the purpose of this surface flow modeling, three models were created, each varying in the degree of fidelity in regard to detail. As and when airborne and surface surveillance data becomes available for flights, the estimates of undelayed times and takeoff for departures are updated. In the highest fidelity model, discrete network event simulations are used to capture dependencies with other flights on the surface, such as propagating downstream constraints, sequencing, and direction of flow. Delays are added at specific intersections based on these dependencies. This algorithm, viewed as a network flow model with flow constraints, is then used to make separation and sequencing decisions. *Initial analysis using the SMS model indicated the necessity for flight prioritization and sequencing starting at the gate area itself, in order correctly represent today’s operation. Moreover, simulation tests run using this model indicated computational challenges, and inaccuracies that were not consistent with the reality of surface operations at the airport. For instance, the number of aircrafts holding at intersections, as computed by the model, was unrealistically large.*

The model that is currently employed in the SMS, termed the ‘SimpleModel’, differs from what is described above. It is lower in fidelity, and assumes that the factors that principally govern airport surface operations are runway constraints and other constraints

³This assumption is similar to that used by the Federal Aviation Administration (FAA) to compute the unimpeded or nominal taxi-out time.

that influence runway times ⁴. It was however concluded that modeling other runway and taxiway restrictions would potentially improve the accuracy of delay estimates. Departure scheduling is effected through a rule-based analysis of spatial constraints and potential runways. It is noted that these sequencing decisions must be adapted as the flight progresses on the airport surface, and relies on information updates related to congestion and runway availability. Hence, as with any departure planning tool, one may expect that the accuracy of predicted departure times would improve nearer to the actual take-off time. The SMS tool aims to blend common situational awareness with Operations Research techniques in an attempt to many unquantifiable uncertainties.

The authors in [Cheng et al., 2001] studied aircraft taxi performance for enhancing airport surface traffic control in which they consider the surface-traffic problem at major airports and envision a collaborative traffic and aircraft control environment where a surface traffic automation system will help coordinate surface traffic movements. In specific, this paper studies the performance potential of high precision taxi towards the realization of such an environment. A state-of-the-art nonlinear control system based on feedback linearization is designed for a detailed B-737 aircraft taxi model. Dareing and Hoitomt [Dareing and Hoitomt, 2002] put forth an interesting discussion on potential benefits that can be derived from specifically assigning accountability (to the FAA, and airlines) for various causes of inefficiency. Alternative ways to measure delays are also suggested.

Other research that has focused on departure processes and departure runway balancing are available in [Atkins and Walton, 2002] and [Idris et al., 1999]. Many statistical models that consider the probability distribution of departure delays and aircraft takeoff time in order to predict taxi-time have evolved in recent years [Tu et al., 2005] and [Shumsky, 1995].

In [Idris et al., 2002], a queuing model for taxi-time prediction is developed. The authors identify takeoff queue size to be an important factor affecting taxi-out time. An estimate of the takeoff queue size experienced by an aircraft is obtained by predicting the amount of passing that it may experience on the airport surface during its taxi-out, and by considering

⁴Such as an Expected Departure Clearance Time (EDCT), a Miles-In-Trail (MIT) restriction, or an Approval Request (APREQ) which is a ‘call for release’

the number of takeoffs between its pushback time and its takeoff time. However, this model requires prior knowledge of actual takeoff times of flights and does not predict taxi-out times in advance of gate pushback. In addition, the model is valid for a specific runway configuration since the runway configuration at the future time of taxi-time prediction is unknown. Suggested extensions to the model include a runway configuration predictor. Test data for a period of one month obtained from the ASQP database showed that the queuing model predicted 66% of the taxi-out times within 5 minutes of the actual time, while a running average model predicted 54% of the flights within the same margin of error.

A queuing model based on simulation to test different emissions scenarios related to duration of taxi-out was developed in [Levine and Gao, 2007]. Some of the scenarios that are considered are redistribution of flights evenly across the day, and variation in number of departures under current capacity. The study showed that lower taxi-out times (and thus lower emissions) are experienced by airlines that use less congested airports and don't rely on hub-and-spoke systems. A Bayesian networks approach to predict different segments of flight delay including taxi-out delay has been presented in [Laskey et al., 2006]. An algorithm to reduce departure time estimation error (up to 15%) is available in [Futer, 2006], which calculates the ground time error and adds it to the estimated ground time at a given departure time. A genetic algorithm based approach to estimating flight departure delay is presented in [Tu, 2006].

The various literature sources reinforce the need for dynamic predictive models that will assist in building a decision support tool to aid ground and local controllers in improving the airport operational efficiency by mitigating surface congestion.

Chapter 3: RESEARCH METHODS

3.1 Addressing Scalability in Approximate Dynamic Programming

In Section 2.4, we saw how the existing Reinforcement Learning (RL) algorithms (Algorithm 3 and Algorithm 4) for solving approximate dynamic programming (ADP) problems presents implementation issues when the dimension ¹ of the problem is even only fairly large. This “curse of dimensionality”, is addressed in the literature through a value function approximation based technique. In this research, we propose to investigate a diffusion wavelet based approximation scheme, to overcome many challenges confronted with, when attempting to approximate the value function ² $V(S)$ (refer Section 2.4.2 for a list of the challenges). In order to introduce the theory of diffusion wavelets, a brief summary of concepts related to classical wavelet theory is presented. This is followed by a discussion of diffusion wavelet theory, and how it can be adapted to approximate the value function in ADP. An algorithm to integrate diffusion wavelet based function approximation with approximate dynamic programming is presented in Section 3.1.4. This approach is then tested on the problem of taxi-out time prediction in air transportation. This integration of the theory of diffusion wavelets and ADP, to enhance the scalability of ADP approaches is the contribution of this dissertation.

The sections that follow begin with a description of classical wavelet theory, and its generalization to diffusion wavelet theory. Its application to value function approximation, and some of the advantages of the method are also discussed.

¹By dimension, we mean the number of state-action combinations.

² $V(S)$ denotes the value of being in state S .

3.1.1 Introduction to Classical Wavelet Theory

One of the many uses of classical wavelet theory is in function approximation. Wavelets are used to represent a function (or signal) via a multiresolution analysis of function spaces. A wavelet can be viewed as a small oscillating function of time or space, which has its energy concentrated in time. Much like the Fourier expansion of a function (using sinusoidal waves), a wavelet expansion, that uses wavelets as the basis functions, can be derived. Wavelets possess a localizing property, which allows a wavelet expansion of a transient function to be modeled with a small number of coefficients. It has been established by [Donoho, 1993] that the wavelet basis system forms what is known as an unconditional basis. This allows for a robust and near-optimal representation of several classes of functions. The properties of an unconditional basis also mean that the coefficients used to represent the function drop off rapidly, allowing for a compact representation. The wavelet expansion is effected through a series of dilations and translations of scaling and wavelet basis functions.

A good introductory treatment of classical wavelet analysis is available in [Burrus et al., 1998], and we reproduce some of the relevant concepts here with several modifications to suit our needs.

Consider the linear decomposition of any real-valued function $f(t)$, which may be expressed as

$$f(t) = \sum_l a_l \psi_l(t) \tag{3.1}$$

where a_l are the real-valued expansion coefficients and, $\psi_l(t)$ are a set of real-valued functions of t called the expansion set. The expansion set is termed a basis for the class of functions that can be expressed as above, if the expansion in Equation 3.1 is unique.

If the basis is orthogonal, then the coefficients may be calculated by the inner product given by Equation 3.2.

$$a_k = \langle f(t), \psi_k(t) \rangle = \int f(t)\psi_k(t)dt \quad (3.2)$$

The wavelet expansion is usually constructed using a two-parameter system, so that,

$$f(t) = \sum_k \sum_j a_{j,k} \psi_{j,k}(t) \quad (3.3)$$

where

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad j, k \in Z \quad (3.4)$$

Here, Z is the set of all integers, j is known as the dilation index, and k is the translation index. The factor $2^{j/2}$ maintains a constant norm independent of scale j . This multiresolution analysis, using the parameterization of time or space by k , and the frequency or scale by j , allows for an effective representation of functions.

The idea behind multiresolution is that, if a set of functions can be represented by a weighted sum of $\phi(t - k)$, then a larger set of functions (including the original), can be represented by a weighted sum of $\phi(2t - k)$. It is noted that, as the scaling index j increases, the wavelet function becomes narrower, and the translation steps in time become smaller. This allows for a finer representation of details of the function, resulting in a better approximation.

These wavelet functions, called the generating wavelet, or mother wavelet, are defined in terms of what is known as the scaling function. The scaling function has the form given by Equation 3.5.

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) \quad j, k \in Z \quad (3.5)$$

The set of all functions that can be expressed using the scaling basis functions is called the function space U , or the span of the basis set.

The span of $\phi_{j,k}$ over k is given by

$$U_j = \overline{\text{Span}_k\{\phi_k(2^j t)\}} = \overline{\text{Span}_k\{\phi_{j,k}(t)\}} \quad (3.6)$$

The wavelet functions $\psi_{j,k}(t)$ span the *differences* between the spaces spanned by the various scales of the scaling function. These wavelet functions allow for a better description of the important features of the function.

The relationship between the various subspaces can be represented by Equations 3.7 and 3.8. A depiction is seen in Figure 3.1. It is assumed that the function spans space U_j . The space U_j is decomposed into several subspaces up to a coarse level. The difference between the subspaces are spanned by the wavelet basis functions denoted by ψ_j while the subspaces themselves are spanned by the scaling basis functions ϕ_j .

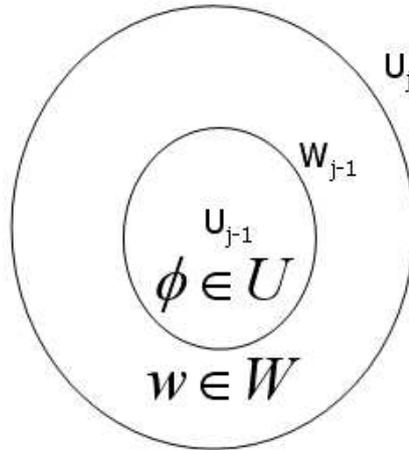


Figure 3.1: Relation between the scaling and wavelet function spaces.

$$U_0 \subset U_1 \subset \cdots \subset U_{j-1} \subset U_j \quad (3.7)$$

Also,

$$U_{j-1} \oplus W_{j-1} = U_j \quad (3.8)$$

The scale of the initial space U_n is arbitrary, and is usually chosen to represent the coarsest level of detail in a function.

The results discussed in wavelet theory are often presented in terms of functions defined on the $L^2(\mathbb{R})$ space which is the set of all square integrable L^2 functions (a function $f(x)$ is square integrable if $\int_{-\infty}^{\infty} |f(x)|^2 dx$ is finite). [Rowland, 2009] provides a note on the L^2 space. However, many of the results hold for larger classes of functions, as well.

In classical wavelet decomposition, any real-valued function $y(t)$ can be expressed as

$$y(t) = \sum_{k=-\infty}^{\infty} c_{j_0,k} \phi(2^{j_0}t - k) + \sum_{k=-\infty}^{\infty} \sum_{j=j_0}^{\infty} d_{j,k} w(2^j t - k) \quad (3.9)$$

where $c_{j_n,k}$ and $d_{j,k}$ are scaling and wavelet coefficients respectively, and are called the discrete wavelet transform (DWT) of the function $y(t)$. If the wavelet system is orthogonal, then the coefficients can be calculated by inner products, and are given by Equations 3.10 and 3.11.

$$c_{j_0,k} = c_{j_0}(k) = \langle y(t), \phi_{j_0,k}(t) \rangle = \int y(t) \phi_{j_0,k}(t) dt \quad (3.10)$$

$$d_{j,k} = d_j(k) = \langle y(t), w_{j,k}(t) \rangle = \int y(t) w_{j,k}(t) dt \quad (3.11)$$

Several orthogonal and biorthogonal wavelet basis functions have been established in the literature, including some popular choices such as the Daubechies, Haar, and Mexican hat.

3.1.2 Diffusion Wavelet Theory

The theory of Diffusion Wavelets is a recent advancement in the field of wavelet analysis (refer [Coifman and Maggioni, 2004]). Diffusion wavelets generalize classical wavelet theory

from one and two dimensional Euclidean spaces, to more general structures such as manifolds. Classical wavelet theory may be viewed as a special case of diffusion wavelet theory. Given a graphical representation of a manifold, the diffusion wavelet transform identifies the best scaling and wavelet basis functions that represent a signal or function on that manifold. The derived basis functions are orthogonal, which provides perfect reconstruction of the signal. The theory is based on deriving diffusion operators that act on functions on the space, and draws on concepts from spectral graph theory. Algorithm 5 describes the steps involved in the diffusion wavelet transform procedure.

Algorithm 5. *Step 1: Obtain the graph (G, E, W) of the m samples of S (domain over which the function is defined), where G represents the set of nodes of cardinality m , E is the set of all edges connecting the nodes, and W is the weights assigned to the edges.*

Step 1a: Compute W using a simple Gaussian kernel function.

$$W_{s_i \sim s_j} = e^{-(\|s_i - s_j\|/\delta)} \quad \forall s_i, s_j \in S \quad (3.12)$$

Step 1b: Define a matrix D such that,

$$D_{s_i \sim s_i} = \sum_{s_j \in S} W_{s_i \sim s_j} \quad \forall s_i \in S \quad (3.13)$$

and,

$$D_{s_i \sim s_j} = 0 \quad \forall s_i \neq s_j \quad (3.14)$$

Step 1c: Obtain the natural random walk on the graph through the matrix operation $P = D^{-1}W$.

Step 2: Obtain the Laplacian L of the graph. For a graph with n vertices, the elements of the $m \times m$ Laplacian matrix are given by Equation 3.15.

$$l_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ adjacent } v_j \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

Step 2a: Compute the diffusion operator $T^1 = T^{2^j}$, ($j = j_0 = 0$) using the relation

$$I - L = D^{-1/2} P D^{1/2} = D^{-1/2} W D^{-1/2} = T^1,$$

where I is the identity matrix of size n .

Step 2b: At level $j = 0$, assume $\phi_0 = I_m$ and $\psi_0 = 0$.

Step 3: Perform a sparse QR-factorization of the diffusion operator T^{2^j} , up to precision ϵ , where ϵ is a small positive number. That is, $\{QR\}_\epsilon = T^{2^j}$. The columns of Q_ϵ provide the orthogonal scaling basis functions ϕ_{j+1} .

Step 4: Perform a sparse QR-factorization,

$$I_{\langle \phi_{j+1} \rangle} - \phi_{j+1} \cdot \phi_{j+1}^* = \{Q' R'\}_\epsilon,$$

where $()$ denotes the complex conjugate. The columns of Q'_ϵ provide the wavelet basis functions ψ_{j+1} .*

Step 5: Increment $j = j + 1$. Obtain $T^{2^j} = R_\epsilon \cdot R_\epsilon^$. Go to Step 6 if no further decomposition of the function space is possible, else go to Step 3.*

Step 6: The output of the algorithm are the scaling and wavelet basis functions at each level of decomposition j .

Now, the reconstruction of the function $f(S)$ may be given by Equation 3.16.

$$\bar{f}(S) = \sum_{k=-\infty}^{\infty} c_{(j_0,k)} \phi_{(j_0,k)}(S) + \sum_{j=j_0}^{\infty} \sum_{k=-\infty}^{\infty} d_{(j,k)} w_{(j,k)}(S) \quad (3.16)$$

For a given sample of size m of the domain S , and corresponding function value $f(S)$, the coefficients at level j are given by Equations 3.17 and 3.18.

$$c_{(j_0,k)} = \langle \bar{f}(S), \phi_{(j_0,k)}(S) \rangle = \frac{1}{m} \sum_{i=1}^m \bar{f}_i(S) \phi_{(j_0,k)}(S) \quad (3.17)$$

$$d_{(j,k)} = \langle \bar{f}(S), w_{(j,k)}(S) \rangle = \frac{1}{m} \sum_{i=1}^m \bar{f}_i(S) w_{(j,k)}(S) \quad (3.18)$$

3.1.3 Value Function Approximation using Diffusion Wavelets

The value function vector $V(S)$, in the field of ADP, can now be approximated using the theory described in Section 3.1.2. Since it is not possible to obtain a vector ρ that forms a complete basis over the space formed by the value function $V(S)$ and the state space S (otherwise we would have $\rho\theta = \bar{V} = V$), we may view \bar{V} as the nearest point projection onto the space formed by the basis functions. The basis functions $\phi(S)$ and $\psi(S)$ are obtained from Algorithm 5. The approximation of the value function, $\bar{V}(S)$ is then given by Equation 3.19.

$$\bar{V}(S|\theta) = \sum_{f \in F} \theta_f \rho_f(S) = \sum_{k=-\infty}^{\infty} c_{(j_0,k)} \phi_{(j_0,k)}(S) + \sum_{j=j_0}^{\infty} \sum_{k=-\infty}^{\infty} d_{(j,k)} w_{(j,k)}(S) \quad (3.19)$$

where, the coefficients $c_{(j_0,k)}$ and $d_{(j,k)}$ may be computed using Equations 3.20 and 3.21.

$$c_{(j_0,k)} = \langle \bar{V}(S), \phi_{(j_0,k)}(S) \rangle = \frac{1}{m} \sum_{i=1}^m \bar{V}_i(S) \phi_{(j_0,k)}(S) \quad (3.20)$$

$$d_{(j,k)} = \langle \bar{V}(S), w_{(j,k)}(S) \rangle = \frac{1}{m} \sum_{i=1}^m \bar{V}_i(S) w_{(j,k)}(S) \quad (3.21)$$

3.1.4 Updating the Value Function: Incorporating the Diffusion Wavelet Scheme in the Iterative ADP algorithm

The ADP algorithm based on updating the values of $V(S)$ over N iterations was described previously in Algorithm 4. Since the ADP solution strategy is based on stepping forward through time (refer Section 2.3.3), Equation 2.27 indicates that when the algorithm enters iteration $n + 1$, the value of being in the future state $S^{n+1} = S^M(S^n, x)$ is required. Now, if the value functions are updated and stored explicitly at each iteration, then $V^{n-1}(S^{n+1})$ is known. However, for large scale problems, it is computationally inefficient (often impossible) to store and access all elements of $V(S)$. Hence, an approximate value of $V^{n-1}(S^{n+1})$ is obtained from Equation 3.19 and used in Equation 2.27. The ADP algorithm based on the diffusion wavelet function approximation scheme may be illustrated by appropriately modifying Algorithm 4. This is described in Algorithm 6.

Algorithm 6. *Step 0: Initialization.*

Step 0a: Initialize $V(S^0)$, S^0 , and set $n = 0$.

Step 0b: Set a limit N_s , on the initial number of states to visit in order to initiate the function approximation scheme.

Step 1: Obtain a sample path ω^n

Step 2: Update S^n if $n > 0$.

Step 3: If N_s distinct states have not yet been visited, go to Step 3e.

Step 3a: If N_s distinct states are being visited for the first time, denote this set of states by S_{N_s} , and go to Step 3b, else go to Step 3c.

Step 3b: Invoke Algorithm 5 to obtain the basis functions $\phi^{n-1}(S)$ and $\psi^{n-1}(S)$ for the sample of N_s states.

Use equations 3.20 and 3.21 to obtain the corresponding coefficients $c_{(j_0,k)}^{n-1}$ and $d_{(j,k)}^{n-1}$.

Go to Step 3d.

Step 3c: Using Equation 3.19 obtain $\bar{V}^{n-1}(S^n)$, the approximate value of being in current state S^n .

Step 3d: Determine the next state $S^{n+1} = S^M(S^n, x, \omega^n)$.

If $S^{n+1} \in S_{N_s}$, then read $\bar{V}^{n-1}(S^{n+1})$ from the stored sample. Set $V^{n-1}(S^{n+1}) = \bar{V}^{n-1}(S^{n+1})$

Set $\phi^n = \phi^{n-1}$ and $\psi^n = \psi^{n-1}$

Go to Step 3e.

Else, add S^{n+1} to the set S_{N_s} , as the last element, and remove the first element of the set S_{N_s} .

Using the updated state space sample, still of size N_s , determine basis functions ϕ^n and ψ^n based on Algorithm 5.

Update the coefficients $c_{(j_0,k)}^n$ and $d_{(j,k)}^n$ using Equations 3.20 and 3.21.

Obtain $\bar{V}^{n-1}(S^{n+1})$ using Equation 3.19. Set $V^{n-1}(S^{n+1}) = \bar{V}^{n-1}(S^{n+1})$.

Go to Step 3e.

Step 3e: Solve

$$\hat{v}^n = \min_{x \in X} (C(S^n, x) + \gamma V^{n-1}(S^M(S^n, x, \omega^n))) \quad (3.22)$$

and let x^n be the value of x that solves Equation 3.22.

Step 3f: Update the value function

$$V^n(S^n) = (1 - \alpha^n)V^{n-1}(S^n) + \alpha^n \hat{v}^n \quad (3.23)$$

Step 4: Find the next state, $S^{n+1} = S^M(S^n, x^n, \omega^n)$

Step 5: $n = n + 1$. If $n < N$, go to Step 1.

Note that the purpose of solving the MDP is to determine the optimal action or decision to be taken in each state. Hence, along with updating the approximation of $V(S)$ in Algorithm 6, the corresponding vector of optimal actions, $X^*(S)$, must also be updated and approximated at each iteration. This is not specifically shown in the description of Algorithm 6 in order to avoid clutter. However, it is easily seen that $X^*(S)$ and $V(S)$ are defined over the same domain, S , and so, the same set of basis functions that is used

to approximate $V(S)$ may also be used to approximate $X^*(S)$. The coefficients used to approximate $X^*(S)$ are obtained by the appropriate use of Equations 3.20 and 3.21.

Advantages of Using Diffusion Wavelets for Value Function Approximation

1. The diffusion wavelet procedure generates the best basis functions as part of the approximation process. The basis functions that are generated form an orthogonal system which allows for accurate and efficient reconstruction of the function. Other regression methods require the basis functions to be pre-specified (linear, quadratic etc.), which may not represent a nonlinear and nonstationary value function accurately.
2. Wavelet transforms are well known to store information in a compact set of significant coefficients. In addition, the number of representative features (basis functions) that is necessary is automatically decided by the diffusion wavelet process.
3. The coefficients (parameters) are updated during the decomposition scheme, therefore a separate recursive parameter updating scheme is not required.
4. It can easily handle multi-dimensional data and compress them efficiently.
5. When states that are not visited in the learning phase are visited during the learnt phase of the implementation, the value of being in this state must be obtained through an interpolation scheme. The diffusion wavelet decomposition procedure provides an effective interpolation method based on the structure of the state space.

3.2 Case Study: Prediction of Taxi-Out Time using Approximate Dynamic Programming

The taxi-out time estimation problem is cast in the probabilistic framework of approximate dynamic programming. In this research, the taxi-out time of a flight is defined as the duration between gate pushback and takeoff. To solve the formulation, a solution strategy

based on the Q-Learning (refer Algorithm 3 for the general Q-learning algorithm) approach is adopted. This forms the baseline for comparing the performance of the diffusion wavelet based value function approximation method. This section discusses details of the taxi-out time estimation problem, including source of the data, the formulation of the problem as a stochastic dynamic programming model, and the Q-learning solution approach.

3.2.1 Data Source

Data from the Aviation System Performance Metrics (ASPM) database, maintained by the Federal Aviation Administration (FAA) is used for this research. The ASPM database records individual flight events such as the time it pushed back from the gate, the time it took off after the taxi-out process, the time it arrived on the runway at the destination airport, and the time it arrived at the gate at the destination airport after the taxi-in process. These events together are abbreviated as OOOI events, to represent gate-Out, wheels-Off, wheels-On, gate-In. Both the *scheduled* and *actual* observed OOOI events are recorded in the ASPM database. In addition, the nominal (or unimpeded) taxi-out time of flights is provided. This quantity represents the average time that an aircraft takes to taxi from gate to takeoff under unimpeded conditions (in the absence of congestion). The nominal taxi-out time is a seasonal average, and is specified for each airline, rather than for each individual flight. Additionally, it is an average across all possible runway configurations. Similar data is available for the nominal taxi-in process of flights, upon arrival at the destination airport. Appendix B describes the method adopted by the FAA to compute these nominal taxi-times.

A snapshot of the ASPM data, for the departure phase and arrival phase, is shown in Figures 3.2 and 3.3.

The various attempts in the literature, to model the airport departure process, use different inputs with distinct levels of detail depending on data availability and accessibility. In this context it is noted that the proposed use of ASPM data in this research does not capture the dynamics of the departure process between the gate pushback and takeoff events.

FLTNO	SCHOUTTM	ACTOUTTM	NOMTO	ACTTO	SCHOFFTM	ACTOFFTM
Flight Number	Scheduled Out time	Actual Out time	Nominal TO time	Actual TO time	Scheduled Off time	Actual Off time
275	07:19	07:30	14.4	13	07:33	07:43

Figure 3.2: Flight data for the departure process:ASPM.

ACTONTM	NOMTI	ACTTI	ACTINTM
Actual On time	Nominal TI time	Actual TI Time	Actual In Time
10:11	5.5	6	10:17

Figure 3.3: Flight data for the arrival process:ASPM.

Also, it is possible that an aircraft pushes back from the gate and for varying reasons (such as refuel after excess taxi) may have to return to the gate and pushback again. It is unclear as to whether the actual pushback time reported by the airlines indicates the first pushback or the second pushback. The Bureau of Transportation Statistics (BTS) recently issued a directive [Source, 2007] effective April 2007 requiring all airlines reporting data to ensure that the first pushback time be recorded as the actual gate-out time. This clearly influences the measured taxi-out time.

3.2.2 Adapting Approximate Dynamic Programming to Taxi-Out Time Estimation

In order to predict taxi-out time using Approximate Dynamic Programming (ADP), we must establish a system state vector, by identifying and quantifying factors that potentially influence the taxi-out process. First, the actual taxi-out times of all flights in a given day, with respect to time of day was plotted. The number of flights scheduled to arrive at the airport, and scheduled to depart from the airport, were also plotted as a function of the time of day. An example of such a plot is seen in Figure 3.4.

A noticeable pattern was observed when comparing the trend in taxi-out times, and the changes in the number of arriving and departing flights with respect to time. For example, it is observed that, as the number of departing flights increases and decreases during the

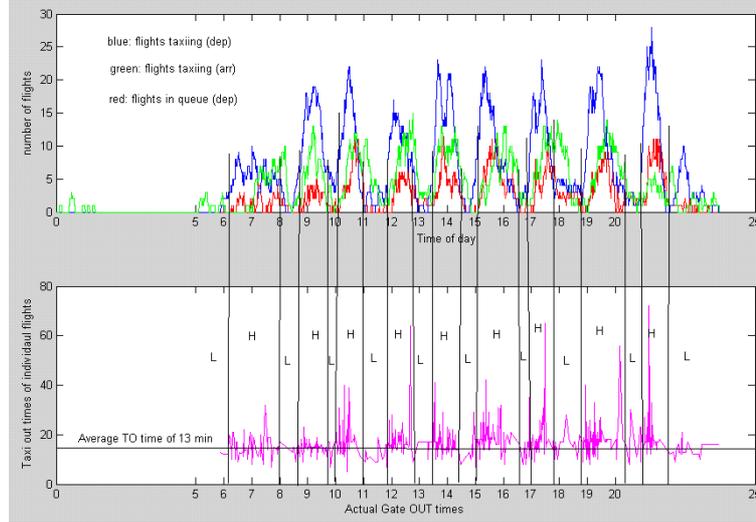


Figure 3.4: Factors influencing taxi-out time.

course of the day, the taxi-out time also correspondingly, increases and decreases, with a time lag of about 30 minutes. This suggests that congestion begins to build up with the number of departures.

Using the available data, the evolving airport system is mapped into system states for a sixty minute period from current time t to time $t + 60$. The system state is obtained for every minute in this 60 minute time period.

Based on the analysis above, the following state variables were represented in the system state of the airport:

1. The number of arriving flights that are taxiing, S_1^t : we say that an arriving flight is taxiing *at time t*, if the following conditions are satisfied -
 - its *On* time is prior to time t ,
 - but its gate-in event is not recorded as yet.
2. The number of departing flights that are taxiing, S_2^t : we say that a departing flight is taxiing *at time t*, if the following conditions are satisfied -
 - its *Out* time is prior to time t ,

but its wheels-off event is not recorded as yet, and its nominal taxi-out time is not yet completed.

3. The number of flights in the runway queue, S_3^t : we consider a flight to be in the runway queue *at time* t , if it has completed its nominal taxi-out time, and has not yet taken off.

Now we consider a 15 minute window starting at current time t , and for every flight scheduled to depart in the interval $[t, t + 15]$, we identify its system state, with the three attributes discussed below. The interval of time considered for the computation of each of these state variables is shown in Figure 3.5.

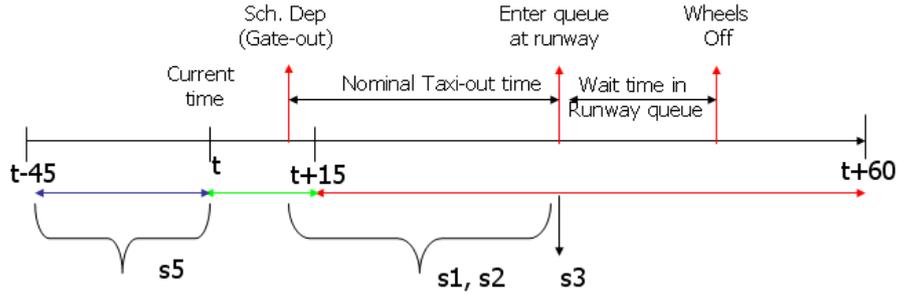


Figure 3.5: Taxi-out time prediction: state variables.

1. The average number of *arriving* flights on the ground that are co-taxiing s_1 : We consider the time interval between the scheduled gate out time and the completion of its corresponding nominal taxi-out time. Letting each minute in this interval be denoted as t_{now} , we now take the average of all the $S_1^{t_{now}}$ values in this time range to be the value of the state variable s_1 .
2. The average number of *departing* flights on the ground that are co-taxiing s_2 : In a manner similar to that explained in (1) above, we consider the time interval between the scheduled gate out time and the completion of its corresponding nominal taxi-out time. Letting each minute in this interval be denoted as t_{now} , we now take the average of all the $S_2^{t_{now}}$ values in this time range to be the value of the state variable s_2 .

3. The number of flights in the runway queue at the end of its nominal taxi-out time s_3 :
We consider the time instant that the flight completes its nominal taxi-out time, say t_{now} . We assign the value of S_3^{tnow} to the state variable s_3 .

It is noted that at some airports (such as Detroit International Airport (DTW)), for a given time period, there may be more than one runway that is used for departure operations. In such a situation, the algorithm assumes that all runways are utilized to a similar extent. Thus, to obtain the runway queue length for that instant in time, we divide the computed total number of aircrafts in queue by the number of departure runways in operation at the airport.

It is useful to note that the state variables s_1 , s_2 , and s_3 involve quantities that may not be known at current time t since the system is yet to evolve. Therefore, for instance, the actual takeoff times of flights beyond time t (for variables s_1 and s_2) must be estimated, rather than assumed known, in the operational phase of the algorithm. In order to determine the takeoff time of a flight scheduled to depart in the near future with respect to current time t , we add the average actual taxi-out time in a 45 minute interval before time t to the flights' scheduled out time. During the training phase of the algorithm, it is reasonable to assume that historical data defining these quantities are available to evaluate the system states. The two phases of the taxi-out time estimation algorithm are discussed in the section that follows. Further, each of the 3 variables is discretized in a pre-determined number of steps. The discretization must be chosen carefully, to ensure that the adopted reinforcement learning solution strategy remains tractable. The exact range of values and discretization may depend on the particular airport studied, since some airports are simply more congested than others and may require that the airport dynamics be captured more rigorously. As an example, at DTW airport, we assume that s_1 and s_2 can each take on values between 0 aircraft and 27 aircraft (and greater) in steps of 2, resulting in 15 possible values (the final discretization step, 15, is used to capture any possible value beyond 26). Similarly, the variable denoting queue length, s_3 can take any of 16 possible values, between 0 flights and 16 flights, in steps of 1.

3.2.3 Q-Learning Implementation of the Taxi-Out Time Prediction Problem

The input to the taxi-out time estimation algorithm is the data from the ASPM database that we described in Section 3.2.1. There are two phases to the implementation - a learning phase, using historic data, and the learnt phase that is then used to obtain taxi-out time predictions on test data.

The Learning Phase

Initially, the current time, t , is set to zero. The airport system states, S_1^t , S_2^t , and S_3^t , are computed for each minute in the interval $[t, t + 60]$. For each flight scheduled to depart in the 15 minute window $[t, t + 15]$, we identify its system state vector, s (consisting of three attributes), as described in Section 3.2.2. A prediction of the taxi-out time is then made, and we denote this by x . Letting $x \in X$ be the set of all possible taxi-out time predictions, we note that theoretically, there are an infinite number of possible values that the taxi-out time of a flight can assume. However, for a convergent process, and for the feasibility of the implementation, we discretize the taxi-out time prediction space. In this research, we assume that the taxi-out times could take on values between 5 minutes and 65 minutes in steps of 1 minute. Reasonably, any observed taxi-out times that are not in this range, is treated as an anomaly in the data. There are thus 60 possible values for the predicted taxi-out times.

The underlying state transition process (the airport dynamics in the $[t, t + 15]$ window) S_t , may be viewed as a Markov chain. At each state, a decision is made by the Air Traffic Controller (ATC), and the Markov decision process (MDP) evolves to the next state. We then view the taxi-out time prediction problem as a MDP in which a taxi-out time prediction is made for each departing flight, based on its system state. The objective of the problem is to minimize the absolute value of the error in the prediction, with respect to the actual observed taxi-out time for that particular departing flight. The cost or error function can be expressed as $C = |\text{actual } TO - \text{predicted } TO|$. The prediction process is solved using

a Reinforcement Learning (RL) based stochastic approximation method with a discounted reward criteria (the Q-learning approach). At the beginning, the elements of the Q-factor matrix are initialized to zero. As the system evolves (using the input data), for each state visited, a prediction is made. The corresponding Q-factor for that state-prediction combination is then updated using the Q-learning version of Bellman's optimality equation, which is given by Equation 3.24.

$$Q(S, x) = (1 - \alpha) Q(S, x) + \alpha \left[C(S, x) + \gamma \{ \min_b Q(S, b) \} \right] \quad (3.24)$$

Here, $C(S, x)$ is the immediate contribution of making prediction x in state S , and $\min_b Q(S, b)$ denotes the minimum value of the Q-factor with regard to state S , obtained by considering all possible predictions $b \in X$.

The parameter α is the decay factor or stepsize, and it is easy to see from Equation 3.24 that, as α decays to zero, the value of $Q(S, x)$ converges. The stepsize rule for α must be carefully investigated since it is critical to the accuracy of the convergence of the Q-factors. Here, γ is the discount parameter, and in some sense, guarantees convergence of the Q-learning algorithm. The learning phase of the taxi-out time estimation algorithm is depicted in Figure 3.6. The final output of the learning phase is the optimal Q-factor matrix.

The algorithm for the implementation of the taxi-out time estimation model is described in Algorithm 7.

Algorithm 7. *Step 0: Initialization.*

Step 0a: Initialize the Q-factors $Q^0(s, x), \forall s \in S, x \in X$.

Step 0b: Set $n = 1$ Step 0c: Set current time $t = 0600$ hours.

Step 1: Map the evolving airport system into system states S_1^t, S_2^t , and S_3^t for the time period $[t, t + 60]$.

Step 2: Determine the number of flights, $i = 1, 2, \dots, n_d$, that actually departed (gate

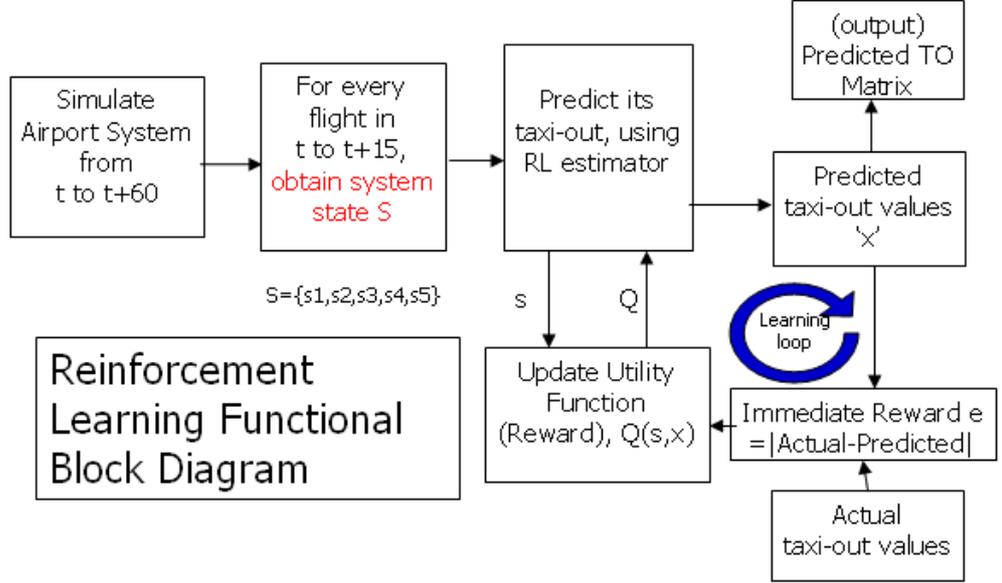


Figure 3.6: Reinforcement Learning Functional Block Diagram.

pushback) in the time interval $[t, t + 15]$.

Step 2a: Set $i = 1$.

Step 3: Determine the system state vector, S^n , for flight i .

Step 3a: Compute

$$\hat{q}_x^n = C(S^n, x^n) + \gamma Q^{n-1}(S^n, x^n) \quad \forall x^n \in X \quad (3.25)$$

where $C(S^n, x^n) = |\text{Actual taxi-out time of flight } i - x^n|$

Step 3b: Update the values of Q^n using

$$Q^n(S^n, x^n) = (1 - \alpha^n)Q^{n-1}(S^n, x^n) + \alpha^n \hat{q}_x^n \quad (3.26)$$

Step 4: $i = i + 1$, $n = n + 1$. If $i < n_d$ go to Step 3.

Step 5: Increment t by one minute, $t = t + 1$. If $t \leq 2300$ hours, go to Step 1.

Step 6: Output final updated Q -factor matrix.

Stepsize Rule for the Taxi-Out Time Estimation Model

The stochastic approximation method introduced in Equation 3.26 requires us to find a stepsize rule for the decay of the learning parameter α which will provide a good rate of convergence. For stochastic and computationally intractable problems, it is impossible to find an optimal stepsize at each iteration. The choice of stepsize rule is excruciatingly problem specific. Note for instance, that the cost function $C(S^n, x^n)$ defined for our taxi-out time estimation problem may exhibit significant variation between successive iterations. This is partially because, there are other unquantifiable factors that influence the actual taxi-out time of each individual flight in the system. For instance, this may include the imposition of downstream restrictions, the adoption of a specific runway and taxiway sequencing strategy based on aircraft size, or a change in the runway configuration during a particular time quarter of the day. Hence, an additional challenge is to choose a stepsize rule that provides a reasonably good tradeoff between minimizing the approximation error (which requires a smaller stepsize), and its ability to respond to the nonstationary data (which works best with a larger stepsize). Several schemes which include deterministic and stochastic rules have been studied in the literature. A detailed review of different approaches along with some necessary proofs is available in [Powell, 2007]. Research over the past several years has identified three necessary conditions to prove convergence of stochastic gradient approaches. It is noted that these conditions are necessary, but may not be sufficient to prove convergence. However, in practice, they provide good rates of convergence and reasonable accuracy for the estimation, when cautiously applied. These three basic conditions are elaborated below [Powell, 2007].

$$\alpha^{n-1} \geq 0 \quad n = 1, 2, \dots \quad (3.27)$$

$$\sum_{n=1}^{\infty} \alpha^{n-1} = \infty \quad (3.28)$$

$$\sum_{n=1}^{\infty} (\alpha^{n-1})^2 < \infty \quad (3.29)$$

First, the stepsizes must be nonnegative, and this is described by Equation 3.27. Equation 3.28 requires that the infinite sum of the stepsizes must be infinite. Observing Equation 3.26, we note that if α decays to zero, then no further learning takes place. The second condition is necessary to prevent the algorithm from stalling prematurely. Thirdly, Equation 3.29 indicates that the infinite sum of the squares of the stepsizes must be finite. This condition ensures two desirable properties, (1) a reasonable rate of convergence, and (2) a guarantee that the variance of our estimate of the optimal value functions goes to zero in the limit.

For our taxi-out time estimation model, we adopt a deterministic rule which depends only on the iteration number (or the number of times the value function is updated). The scheme, based on a heuristic approach is determined by Equations 3.30 and 3.31.

$$\alpha^n = \frac{\alpha^{n-1}}{1 + a^n} \quad (3.30)$$

$$a^n = \frac{n^2}{b + n} \quad (3.31)$$

where, b is a very large positive number, that is set based on the total number of iterations.

We must now verify if Equations 3.30 and 3.31 satisfy the three basic conditions given by Equations 3.27, 3.28, and 3.29. We avoid a rigorous approach based on proofs from mathematical analysis (an excellent treatment of this field may be found in [Rudin, 1976]). Rather, we graphically compare the properties of the stepsize rule given by Equations 3.30 and 3.31, with an alternative stepsize rule that we know satisfies the three necessary conditions given by Equations 3.27, 3.28, and 3.29.

We start, by noting that these conditions require that the stepsizes decrease proportionally by a sequence such as $\alpha^{n-1} = \frac{1}{n}$. However, in practice this sequence decays very quickly, resulting in what is called “apparent convergence”. Hence, we compare our stepsize rule (given by Equations 3.30 and 3.31) with the extension of the basic harmonic sequence $\alpha^{n-1} = \frac{1}{(n)^\beta}$, which is a generalization of the $1/n$ rule. Here, $\beta \in (\frac{1}{2}, 1]$.

Figures 3.7 and 3.8 depict the decay rates for the two schemes, over 1000 iterations. We choose $b = 5000000$, for our scheme, and $\beta = 0.6$ for the harmonic sequence. It is noted that the stepsize rule chosen for our taxi-out time estimation model provides a more gradual decay than the one based on the harmonic sequence. As discussed in Section 2.4.2, a gradually changing decay rate is also crucial for overcoming challenges in value function approximation methods (such as the one described in Algorithm 6).

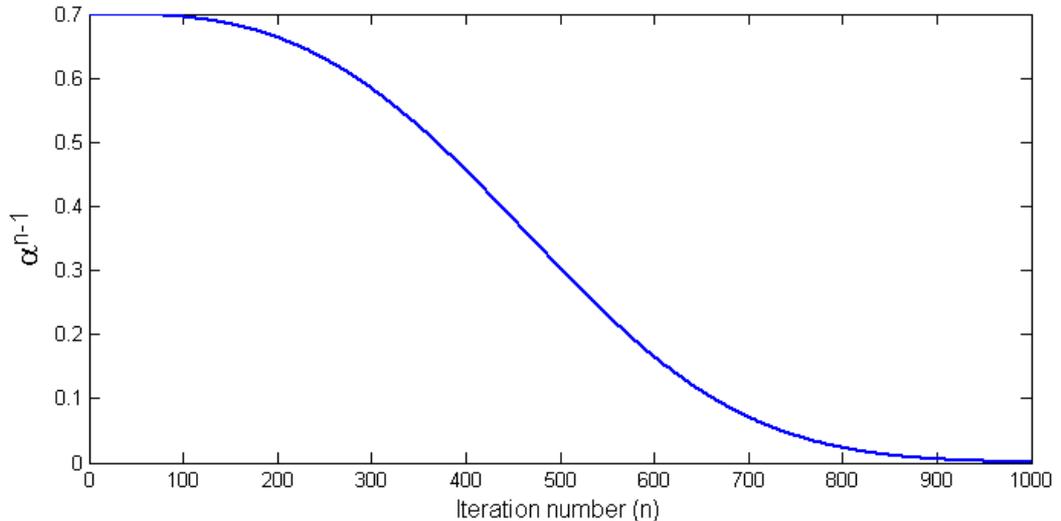


Figure 3.7: Decay of stepsizes for the taxi-out time estimation algorithm.

Figures 3.9 and 3.10 represent the two conditions given by Equations 3.28 and 3.29. Equation 3.28 is the most important condition for convergence, and based on the discussion presented above, it can be easily argued that an appropriate selection of values for the parameters b or β , based on the total number of iterations, will guard against the undesirable outcome of “apparent convergence”. Additionally, both schemes provide a plateau region

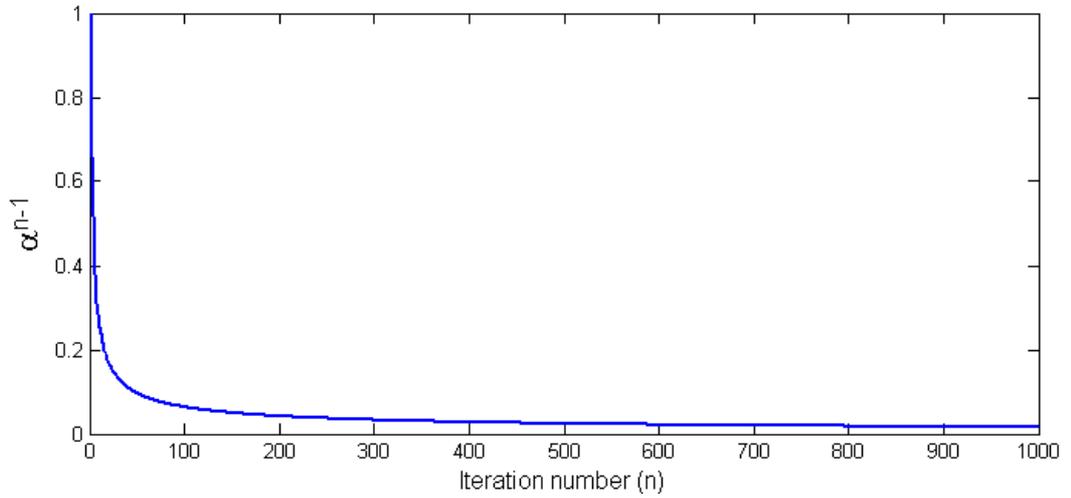


Figure 3.8: Decay of stepsizes for the generalized harmonic sequence.

for the condition given by Equation 3.29, as the number of iterations increase.

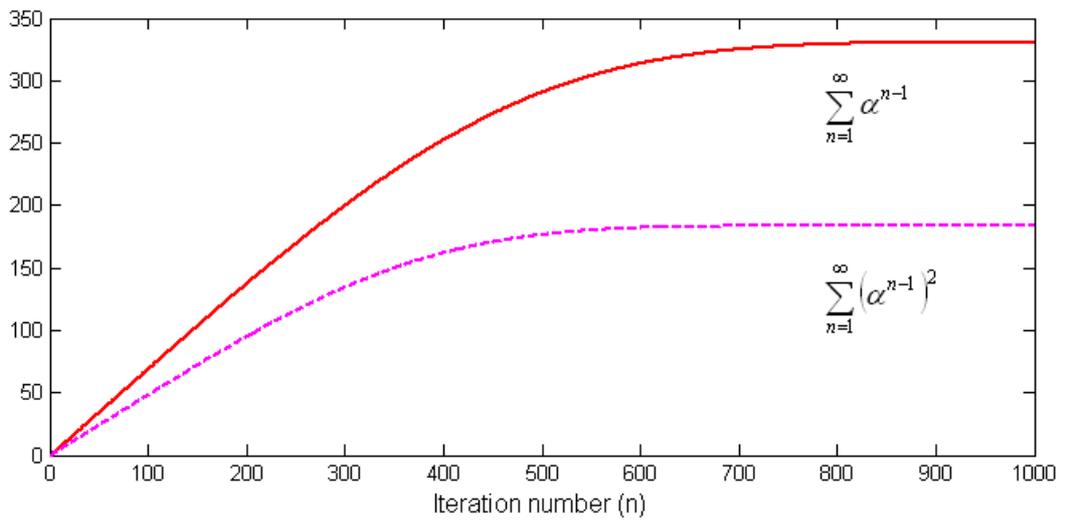


Figure 3.9: Convergence properties of stepsizes for the taxi-out time estimation algorithm.

The Learnt Phase

The “learnt phase” of the algorithm now uses *test* data, for which taxi-out time predictions are desired. Much the same way as the learning phase, for each flight that is scheduled to depart, its system state, say S^f is obtained, as described in Section 3.2.2. Then, the

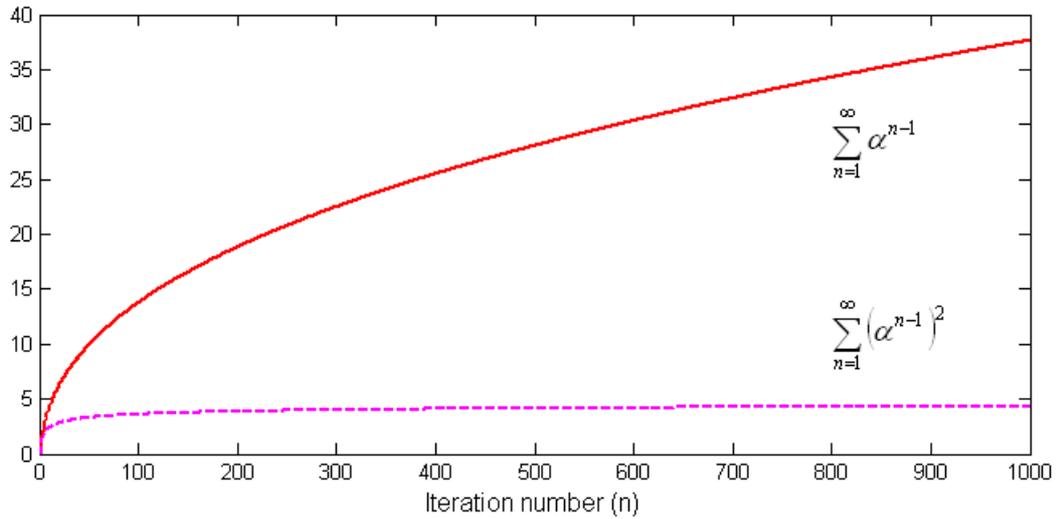


Figure 3.10: Convergence properties of stepsizes for the generalized harmonic sequence.

corresponding row is identified in the optimal Q-factor matrix which is of the form $Q(S, x)$. The optimal prediction for the flight under consideration is given by $x^* = \arg \min_{x \in X} Q(S^f, x)$.

A visual representation of this process is shown in Figure 3.11.

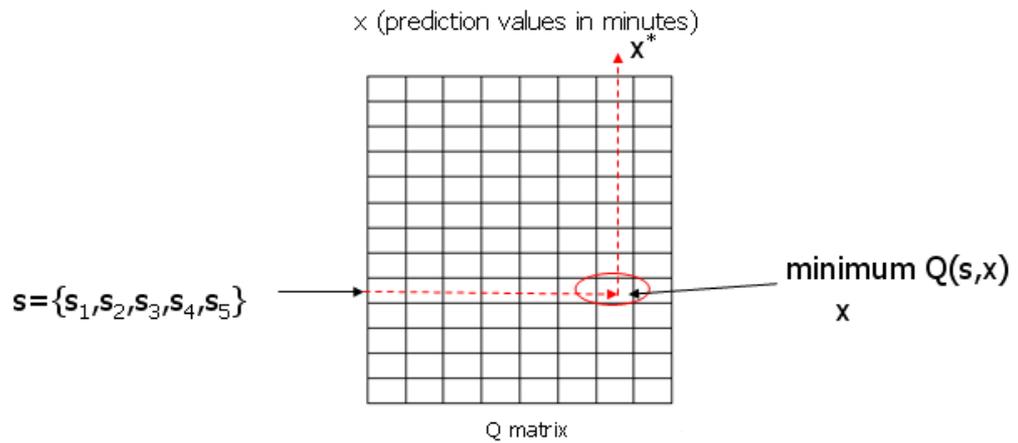


Figure 3.11: Learnt Phase: Q-factor matrix.

3.2.4 Advantages of using RL for Taxi-Out Time Estimation

1. The problem of sequential predictions is well-suited to the framework of stochastic dynamic programming.
2. Due to the uncertainties involved and the complex nature of airport operations, it is often difficult to obtain differential equation based mathematical models and closed form solutions that completely describe airport dynamics.
3. RL is a model free approach (data-driven) that is adaptive to changing airport dynamics.
4. RL learns by interacting with the environment (alleviating need for good training data such as in neural networks).
5. RL is suitable for large-scale optimization due to its simple recursive formulation.
6. RL is suitable for sequential decision making under uncertainty due to its strong mathematical construction using stochastic optimization theory

3.2.5 Integrating Diffusion Wavelet Based Value Function Approximation and Taxi-out Time Estimation

The previous section described the development of an ADP based model for taxi-out time estimation. A Q-Learning based solution strategy was presented, along with details of the algorithm and implementation. The Q-Learning approach results in a look-up table scheme which stores the value of being in each state-action pair. In many real world applications, including our problem of taxi-out time estimation, the number of possible states and actions (and hence state-action combinations) becomes prohibitively large, making the Q-Learning approach computationally inefficient in terms of storage capacity, and run-time speeds. In this section, we discuss an alternative solution approach based on the value function approximation method using diffusion wavelet theory (developed in Section 3.1.4).

As a first step, the Q-Learning implementation of the taxi-out time estimation model (refer Algorithm 7) is converted to a value function based strategy (this transition is described in detail in Section 2.4). Now, the diffusion wavelet based value function approximation scheme (detailed in Algorithm 6) can be suitably adapted to our problem. A simplified schematic representation of the algorithm is shown in Figure 3.12. Algorithm 8 provides a more detailed description of the implementation procedure. Note that there is a subtle difference between the implementation in Algorithm 6 and Algorithm 8: In Algorithm 6 we use a predetermined sample size for the number of distinct states to be visited before the value function is approximated, while in Algorithm 8, we allow the process to evolve for a specified time period, and develop the sample size based on the unique set of states visited in this time period.

Algorithm 8. *Step 0: Initialization.*

Step 0a: Initialize $V(S^0)$, and set $n = 1$, $N_s = 0$. Initialize the set of distinct states, $S_{N_s} = []$.

Step 0b: Set the number of days in the learning time period, $j = 1, 2, \dots, N_{days}$. Initialize $j = 1$.

Step 0c: Set current time $t = 6000$ hours.

Step 1: Map the evolving airport system into system states S_1^t , S_2^t , and S_3^t for the time period $[t, t + 60]$.

Step 2: Determine the number of flights, $i = 1, 2, \dots, n_d$, that actually departed (gate pushback) in the time interval $[t, t + 15]$.

Step 2a: Set $i = 1$.

Step 3: Determine the system state vector, S^n , for flight i .

Step 4: If $j = 1$ and $t < 2200$ hours, set indicator = 1, go to step 7.

Step 5: Set indicator = 0.

Step 5a: If $S^n \in S_{N_s}$, go to Step 7.

Step 6: Invoke Algorithm 5 to obtain the basis functions $\phi^{n-1}(S)$ and $\psi^{n-1}(S)$ for the sample of N_s states.

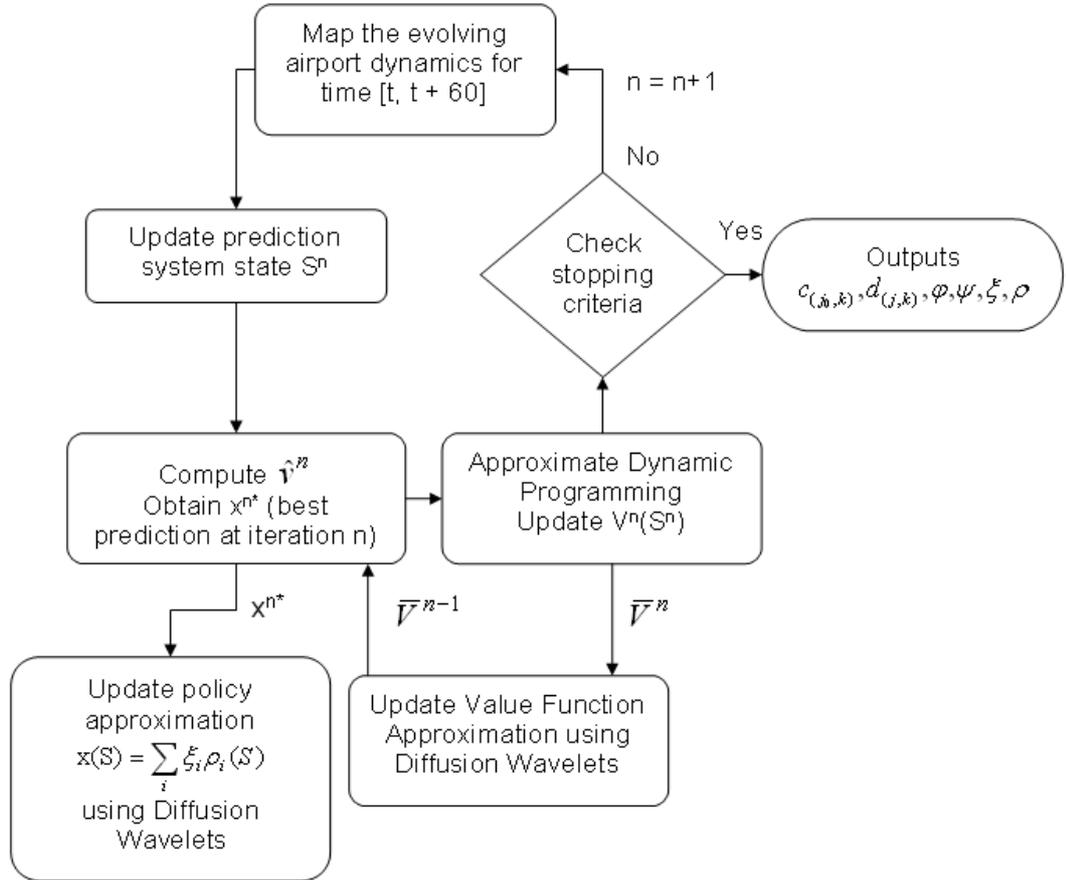


Figure 3.12: Schematic representation of the diffusion wavelet based taxi-out time estimation algorithm.

Step 6a: Use equations 3.20 and 3.21 to obtain the corresponding coefficients $c_{(j_0,k)}^{n-1}$ and $d_{(j,k)}^{n-1}$.

Step 6b: Add S^n to the set S_{N_s} , as the last element, and remove the first element of the set S_{N_s} . Using the updated state space sample, still of size N_s , determine basis functions ϕ^n and ψ^n based on Algorithm 5.

Step 6c: Using Equation 3.19 obtain $\bar{V}^{n-1}(S^n)$, the approximate value of being in current state S^n . Set $V^{n-1}(S^n) = \bar{V}^{n-1}(S^n)$

Step 6d:

Set $\phi^{n+1} = \phi^n$ and $\psi^{n+1} = \psi^n$

Update the coefficients $c_{(j_0,k)}^n$ and $d_{(j,k)}^n$ using Equations 3.20 and 3.21.

Step 7: Solve

$$\hat{v}^n = \min_{x \in X} (C(S^n, x) + \gamma V^{n-1}(S^n, x)) \quad (3.32)$$

and let x^n be the value of x that solves Equation 3.32.

Step 7a: Update the value function

$$V^n(S^n) = (1 - \alpha^n)V^{n-1}(S^n) + \alpha^n \hat{v}^n \quad (3.33)$$

Step 7b: If indicator = 0, go to step 9.

Step 8: If $S^n \notin S_{N_s}$, then, add S^n to S_{N_s} as the last element, and, $N_s = N_s + 1$.

Step 9: $i = i + 1$, $n = n + 1$. If $i < n_d$, then go to Step 3.

Step 9a: Increment t by one minute, $t = t + 1$. If $t \leq 2300$ hours, go to Step 1.

Step 9b: $j = j + 1$. If $j < N_{days}$, go to Step 0c

It must be noted again, that the policy obtained at each iteration n , which is also a function of the state space, S , must be approximated along with $V(S)$ at each iteration. This is a straightforward extension, and we do not explicitly present it in the description of Algorithm 8. The output of the algorithm is the final set of sampled states of size N_s , and the corresponding value function vector, and policy vector.

The learnt-phase (or the test-phase) algorithm differs significantly from that of the Q-learning algorithm since we now do not have a complete look-up table for the prediction policy corresponding to each state. Instead, in a manner very similar to the idea presented in Algorithm 8, at the n^{th} iteration, when a state S^n is visited; if $S^n \in S_{N_s}$, then the optimal prediction is simply read out of the policy vector corresponding to the stored samples. However, if $S^n \notin S_{N_s}$, then S^n is added to set S_{N_s} as the last element, and the first element of S_{N_s} is removed. The diffusion wavelet basis functions are updated using Algorithm 5, and the appropriate use of Equation 3.19 provides the optimal action for state S^n . Subsequent

to this step, Equations 3.20 and 3.20 are used to update the scaling and wavelet coefficients for the prediction policy vector, based on the revised set of system states, S_{N_s} . This process is repeated until a prediction is obtained for all departing flights in the test-phase.

Note that while the Q-factor based implementation in Algorithm 7 required the storage of a two-dimensional look-up table with the number of elements equal to all possible state - action combinations, the value function approximation method described in Algorithm 8 requires the storage of only a small sample of the $V(S)$ vector, and a compact set of basis functions and coefficients to reconstruct the value function vector.

By adopting the diffusion wavelet scheme for approximating the value functions, an additional advantage of significant value is obtained. Consider the situation where, a state is never visited during the learning phase of the algorithm. This means that the value of being in this state would never have been updated. However, it is possible that this state may be visited during the learnt (or test) phase, for which an optimal action is desired. This implementation challenge requires an interpolation scheme for the value functions. That is, if the value of being in a state that was not visited during the learning phase is required, it is estimated from the values of being in its neighboring states. Interpolation techniques are again computationally intensive, and the original problem of determining the optimal action for a given state is often extended further, so as to determine the nearest neighbors. The diffusion wavelet procedure (described in Algorithm 5) has an efficient interpolation scheme built-in to the method used to derive the basis functions. Recall that the state space is represented as a graph, and the weights on the edges are computed based on the Gaussian kernel function, which defines the nonlinear distance between any two states. The diffusion wavelet approach thus provides a natural and convenient framework for representing the nearest neighbors of a given state. Moreover, the computation of the value of being in a newly visited state follows directly from the diffusion wavelet decomposition and reconstruction algorithm, eliminating the need for a separate interpolation method.

Chapter 4: ANALYSIS, EXPERIMENTS, AND RESULTS

In this chapter, a detailed case-study of Detroit International Airport is presented. The characteristics of the airport are described in terms of the departure process and is followed by an analysis of the results based on the algorithms developed in Chapter 3. The analysis is extended to include results for John F. Kennedy International airport.

4.1 Airport Analysis

The reinforcement learning (RL) method developed to predict taxi-out times was used to study taxi-out time predictions at two airports in the United States: Detroit International Airport (DTW), and John F. Kennedy International Airport (JFK). Taxi-out time behavior is affected by the complex interactions of several factors that are local and extraneous to the airport. Viewed at a macro level, the impact of some factors on the actual behavior of taxi-out times at each of these airports tend to vary. Principally, these may be identified as (1) the nature of operations instituted by the airlines, such as schedule banking during specific periods of the day, or whether or not the airport is used for hub operations, (2) geographic location, which results in varying influence of climatic and weather phenomena that sets limits on capacity and visibility, and (3) airport infrastructure, such as number and layout of runways and taxiways. The two airports, DTW and JFK were chosen for the analysis so as to represent airports with different infrastructure and operating characteristics. Sections 4.1.1 and 4.3 describe each airports' characteristics more in detail.

The factors incorporated in the taxi-out time prediction model, which capture the extent of surface congestion as experienced by a flight may be adapted and universally adopted at different airports. For a given flight, the factors influencing its taxi-out time were the number of arriving and departing flights that are co-taxiing (s_1 and s_2 respectively), and

the number of flights in the runway queue (s_3). It is noted that the variables s_1 , s_2 , and s_3 are functions of the arrival and departure demand at the airport.

The pattern of actual taxi-out times at each airport was studied in more detail. In general, even at a given airport, the mean taxi-out time across different days tend to vary. In addition, the variance of actual taxi-out times across a given day can be extremely high. Moreover, at an individual flight level, even for a given time of day, there are flights that have lower taxi-out times, and flights that experience high taxi-out times. Specific plots relating to each airport are provided in Sections 4.1.1 and 4.3. These trends suggest that there are factors that influence specific flights, causing them to experience extended taxi-out times. These factors could include the implementation of a Ground Delay Program (GDP) due to a capacity limit at the destination airport, ground stops (usually of shorter duration when compared to a GDP), a miles-in-trail restriction for flights that depart in a certain direction or through a specific departure fix, or simply because, that individual flight was affected by traffic flow management procedures that involve the sequencing of several arriving and departing flights.

The unique operating characteristics of the two airports, as well as the taxi-out time trends across recent years, months, and individual days are studied in detail in the sections to follow. The infrastructure prevalent at each airport during the years for which the taxi-out time prediction analysis is conducted (September 2005 to August 2008) is also explained.

4.1.1 Detroit International Airport (DTW)

The airport diagram for DTW is shown in Figure 4.1. The airport has six major runways. The most frequently used runway configurations are [21L,22R|21R,22L] and [4L,3R|4R,3L], where the runways specified after the symbol ‘|’ refer to the departure configuration. It is seen that the departure and arrival runway operations are fairly independent at DTW, except for some instances when the taxiways intersect with the departure runways.

During the years 2005, 2006, 2007, and for the most part of 2008, the airport had

three terminals - McNamara, Smith, and Berry (In September 2008, the Smith and Berry Terminals were replaced by the North Terminal. The effect of construction during the early months of 2008 on airline operations is not clear, and is not accounted for in the prediction model developed as part of this research). The airport is the second largest hub of Northwest Airlines (operating from the McNamara Terminal), and Spirit Airlines (operating from the Smith Terminal). In Figure 4.1, the McNamara Terminal is the South Terminal, while the Smith and Berry Terminals were located around the same area as the North Terminal marked on the airport diagram.

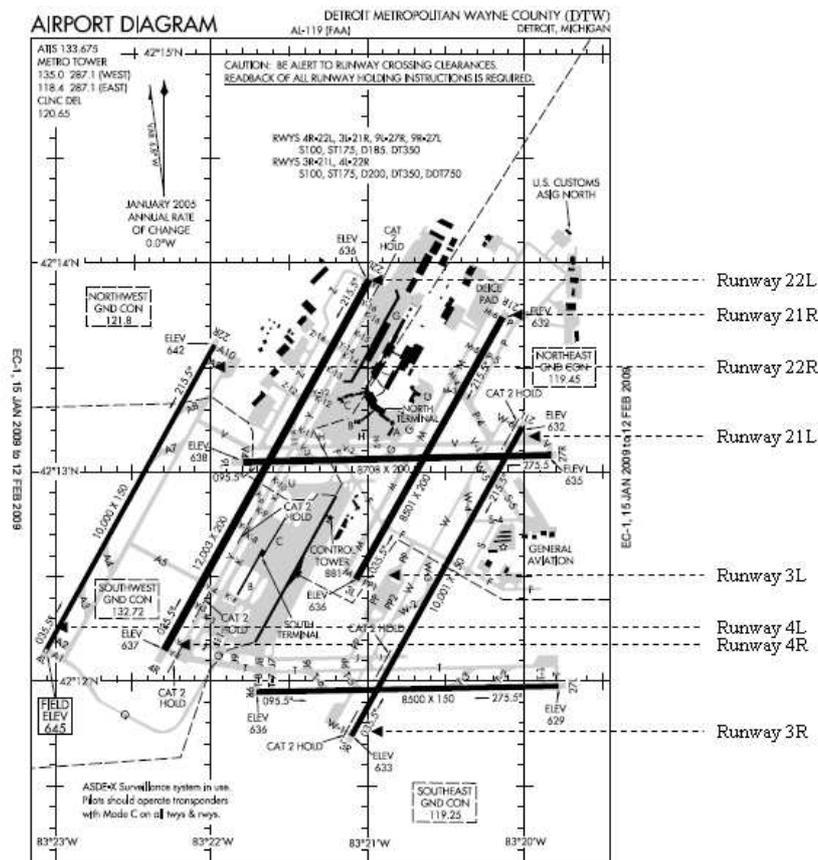


Figure 4.1: Airport Diagram (DTW).

Due to the geographic location of the airport there is a possibility of flights experiencing icing conditions during the winter months. This may require departing flights to taxi to, and hold, at a deicing pad before being ready for takeoff. On the other hand, taxi-out times

during the summer months may be influenced by thunderstorms in the region.

Due to these seasonal effects, the study of long term taxi-out time behavior in each year was divided into two groups - Fall and Winter (September through February), and Spring and Summer (March through August). Several plots were obtained in order to understand the nature of operations at the airport relative to the three time periods of the study. Figures 4.2 - 4.5 show the mean taxi-out time and standard deviation of taxi-out times for each day for the two seasonal groups, across three years.

Consider the period between September and February (Figure 4.5). The mean taxi-out times for each given day remains steady from the first week of September up to about the middle of November. This trend is consistent for all three years of study. Note however, that there is a shift of about 4 to 5 minutes in the daily means for the years (Sep06 - Feb07) and (Sep07 - Feb08), when compared to (Sep05 - Feb06). Beyond mid-November, the trend of the daily means is seen to have several peaks and troughs; these variations being more marked in the plot for (Sep07 - Feb08).

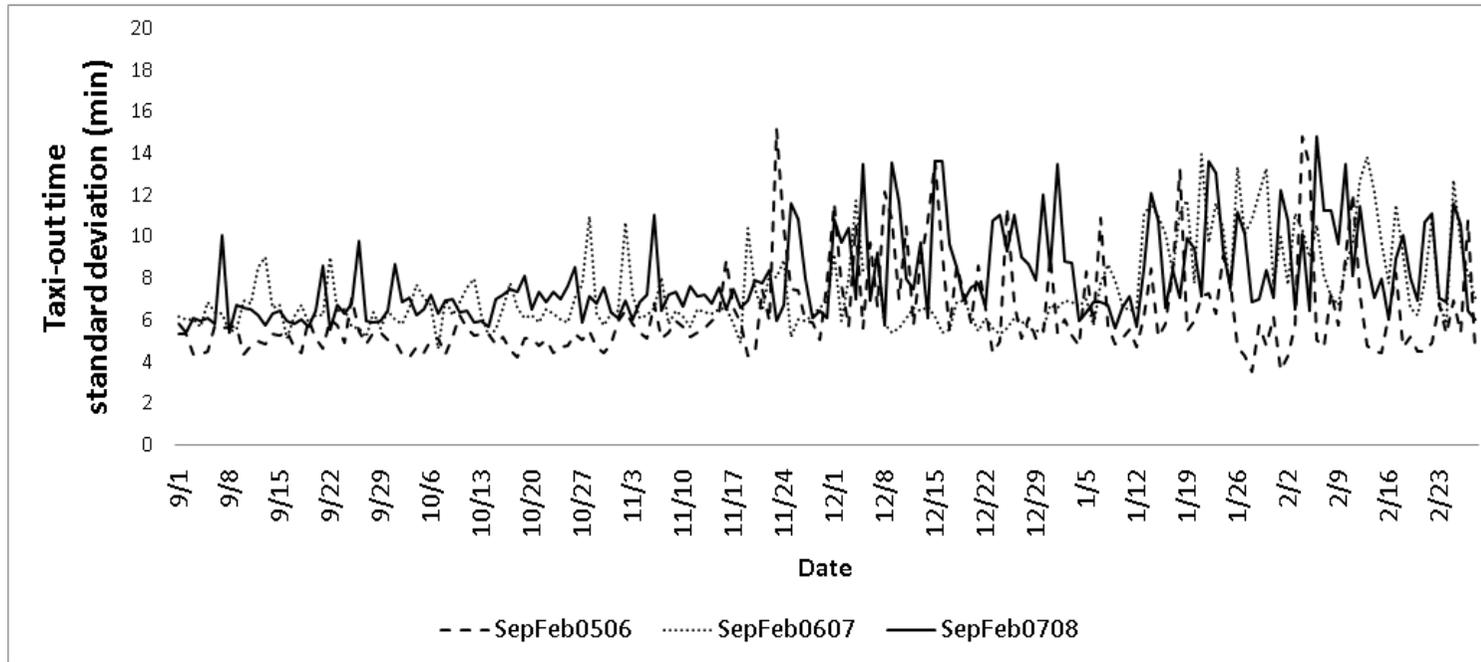


Figure 4.2: Daily standard deviation of taxi-out time (DTW, Mar-Aug).

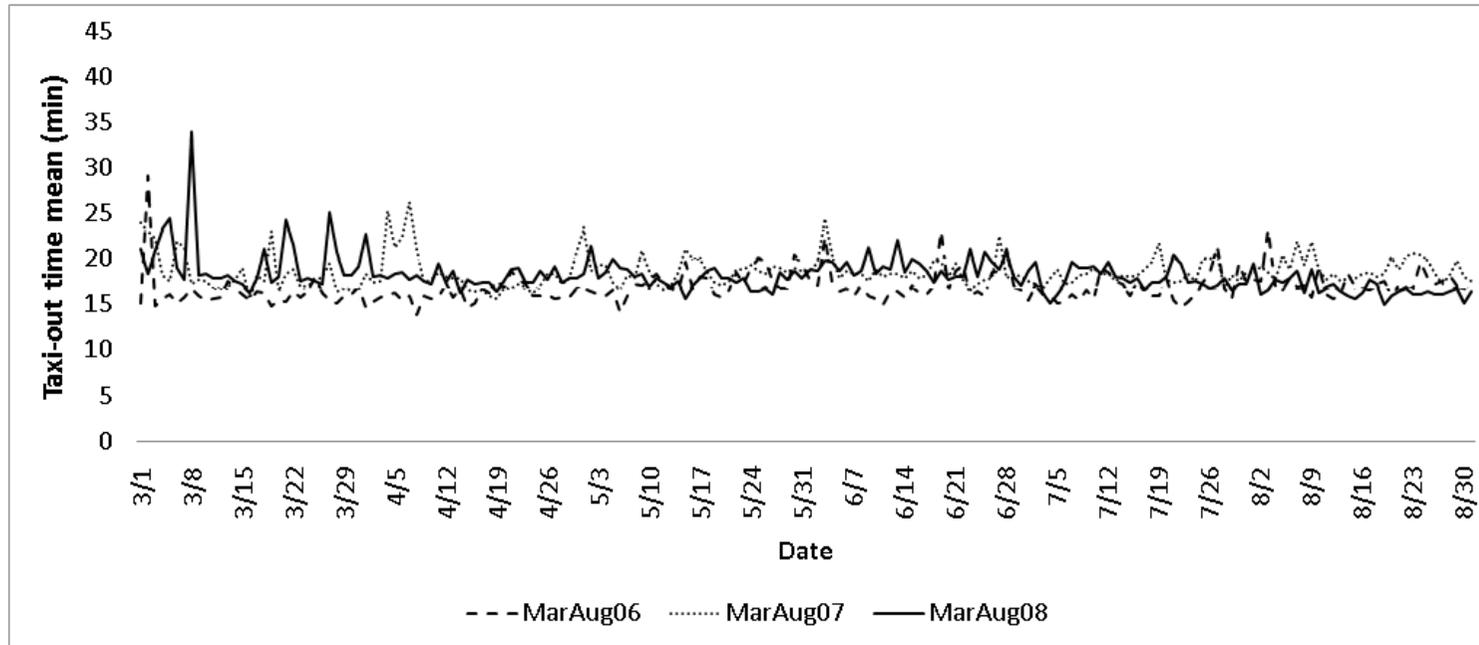


Figure 4.3: Daily means of taxi-out time (DTW, Mar-Aug).

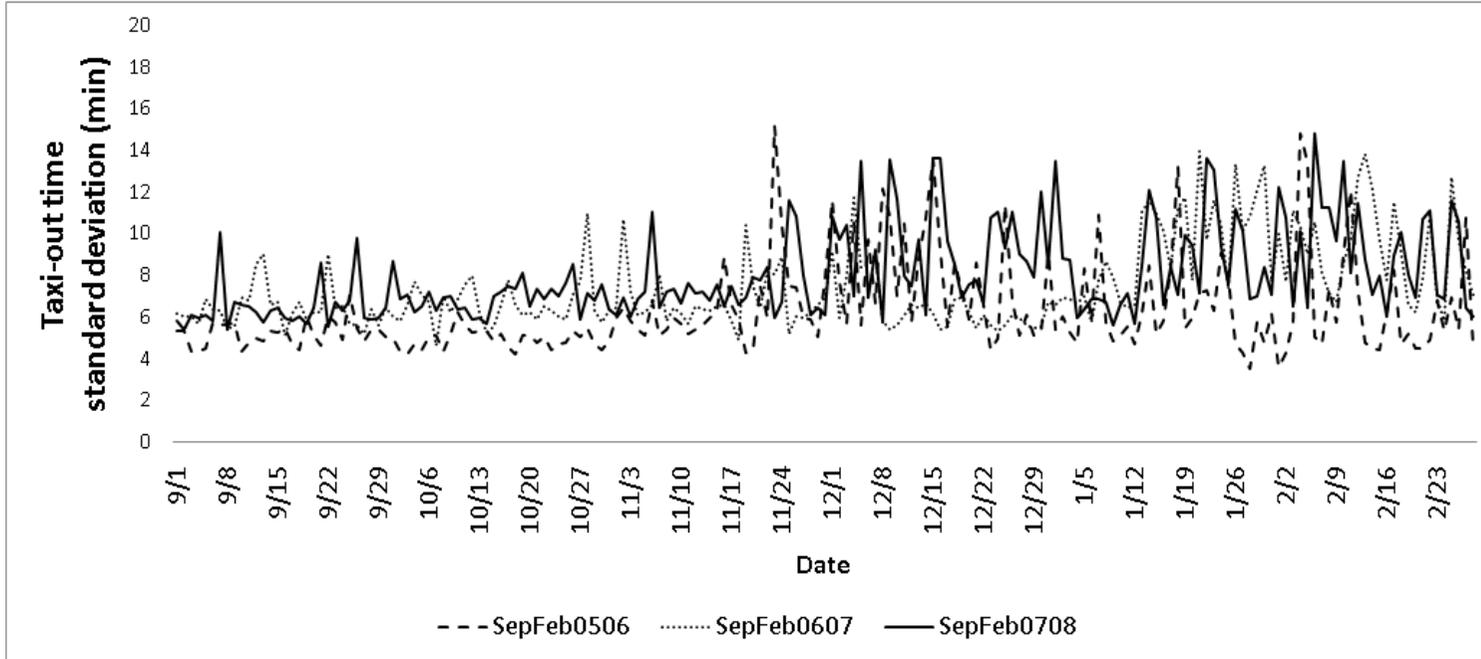


Figure 4.4: Daily standard deviation of taxi-out time (DTW, Sep-Feb).

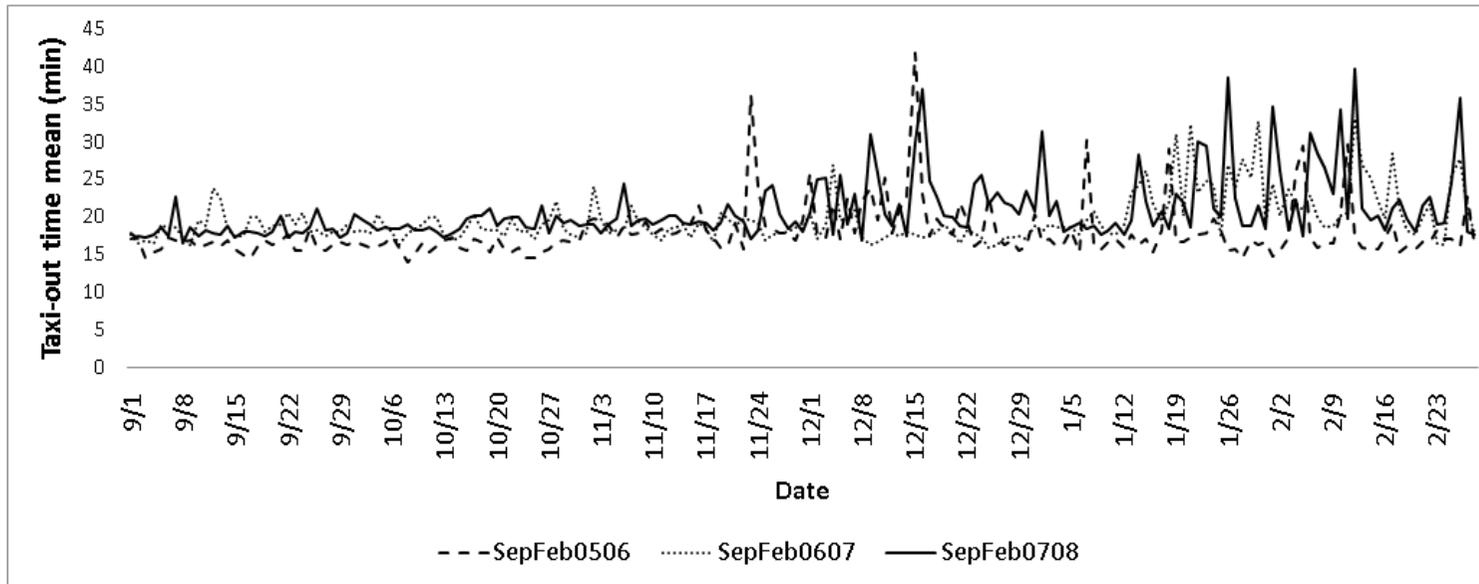


Figure 4.5: Daily means of taxi-out time (DTW, Sep-Feb).

Plots on the demand pattern at the airport over the three years 2005-2006, 2006-2007, and 2007-2008 were also obtained in an attempt to understand the change in overall taxi-out time behavior across the years. The total number of daily operations was plotted for the period between September and February (Fall and Winter) (refer Figure 4.6). There is no indication that the number of operations had changed significantly over the years of study. In fact, for the 3 months in Fall (September - November), the number of operations in the period (Sep05 - Feb06) is higher than that in (Sep06 - Feb07) and (Sep07 - Feb08) by about 60 - 80 flights. The periodic drop in the number of operations for all three years can be attributed to the reduced demand during weekends.

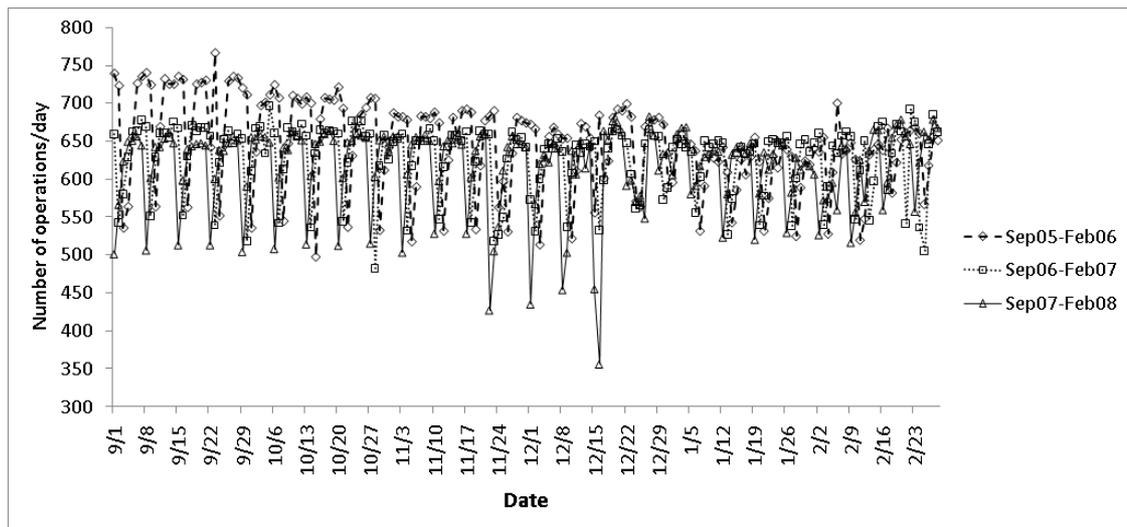


Figure 4.6: Total daily departure demand.

An hour-by-hour plot of the average demand for pushback (Figure 4.7) and departure over all the days between September and February (Figures 4.7 and 4.8) also shows similar demand patterns across the three years.

This analysis suggests that the increase in taxi-out time means and variance that is observed in the recent years (Sep06 - Feb07) and (Sep07 - Feb08) may not be caused by demand pattern changes.

Apart from the variations in taxi-out time across days of the month, which influences

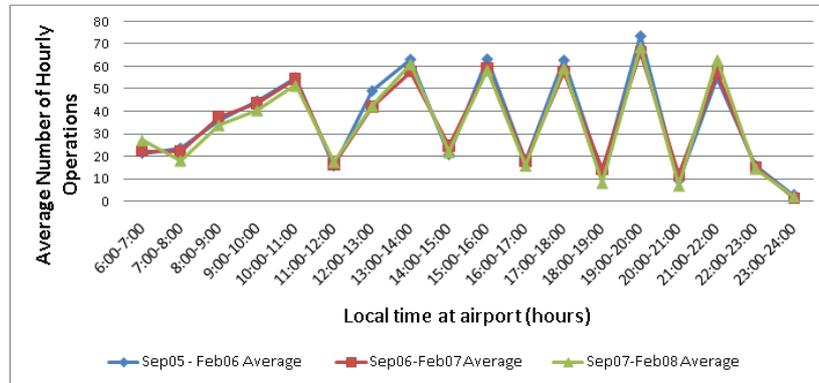


Figure 4.7: Average hourly demand for pushback.

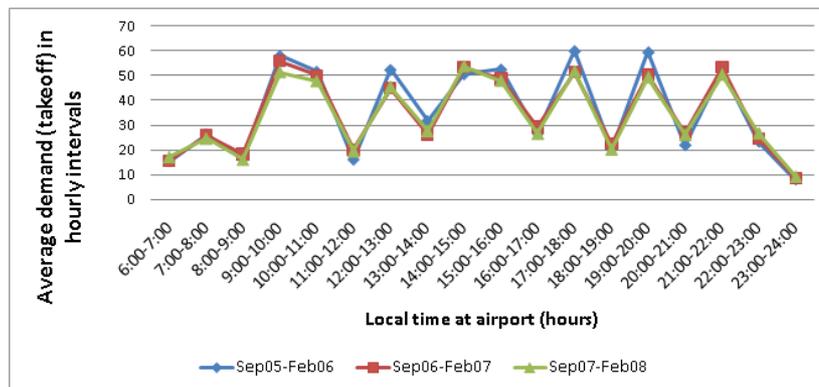


Figure 4.8: Average hourly demand for takeoff.

the training phase of the taxi-out time estimation model, variations may be found within a given day. It is impossible to present plots of taxi-out time trends for every single day in the time period of study. Still, it is useful to view these plots for a few representative days since these trends are unique to each airport and are often informative. These plots are presented and discussed along with the prediction results, when relevant.

4.2 Taxi-Out Time Estimation Results

To achieve the goals of this dissertation, taxi-out time predictions for each period of study were obtained using different methods. These methods are summarized in Figure 4.9.

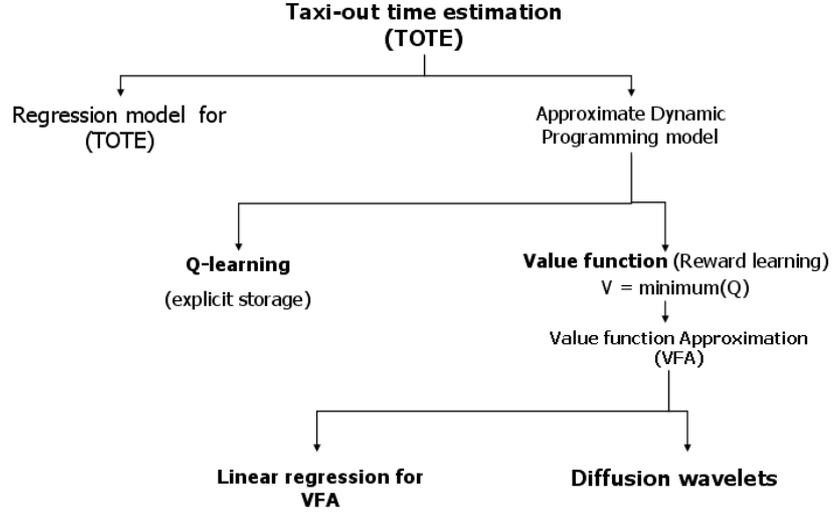


Figure 4.9: Methods compared for taxi-out time estimation.

The diffusion wavelet based value function approximation method described in this dissertation is examined along with state-of-the-art algorithms: a Q-learning approach based on explicit storage of value functions pertaining to each state and a value function approximation method based on linear regression. The analysis aims to address two questions as detailed below.

1. The state-of-the-art is a Q-learning approach based on explicit storage of value functions pertaining to each state. The analysis aims to answer the question - Does the diffusion wavelet value function approximation method provide a good enough solution when examined along with the Q-learning approach. This is an indication of whether the diffusion wavelet based method was successful in identifying representative basis functions to approximate the value function vector. Secondly, an analysis on storage requirements to examine computational feasibility is conducted.
2. The state-of-the-art is a linear regression based value function approximation solution method. The taxi-out time estimation model described has a linear state space structure. This indicates that for this problem, linear basis functions may be a good choice for value function approximation. A second question that the analysis aims to

answer is - Does the diffusion wavelet value function approximation method provide a good enough solution when examined along with the linear regression based value function approximation scheme.

4.2.1 Experiments and Analysis Based on the Q-Learning Algorithm

Training and testing based on the reinforcement learning algorithm was conducted for each of the six month periods of study. In each six month period, seven days were picked at random from each month for the test phase. Hence, there are 42 days taken for testing the accuracy of the algorithm. The remaining days were used to train the algorithm in the learning phase.

The prediction accuracy of the algorithm is represented in two different ways:

1. Individual flight prediction accuracy for a given day - For effective departure planning and scheduling, stakeholders, including airlines and air traffic controllers would benefit from taxi-out time predictions that are accurate within 4 minutes of the actual taxi-out time of the flight (or a root mean square error of 2 minutes). First, for each day in the test phase, the percentage of flights that are predicted within this range of accuracy is computed. This result is then used to obtain confidence intervals for the prediction accuracy across 42 days. Additionally, the percentage prediction accuracy for each hour of the day is also computed. Due to constraints on the size of the page, these results from all the periods of study are spread across 18 tables (Figures 4.29 - 4.46) towards the end of this chapter. Note however, that the confidence intervals for the prediction accuracy are computed for each six month period of study (i.e. across 42 days). The confidence intervals evaluated at 95%, for each of the six periods of study are shown in Figure 4.11. In addition, for each test day, the mean, standard deviation, and median of actual taxi-out times, and for the corresponding predicted taxi-out times were tabulated. These tables are shown in Figures 4.71 - 4.76.
2. Average prediction accuracy in 15 minute intervals of the day - This scenario is

schematically depicted in Figure 4.10. A comparison was made between the average actual taxi-out time per quarter and the average predicted taxi-out time per quarter across the entire day. First all flights that were predicted to take off in a certain quarter are considered and their corresponding mean predicted taxi-out times are plotted. Subsequently, all flights that actually took off in that same quarter are extracted and their corresponding mean actual taxi-out times are plotted. It is to be noted that the flights that actually took off in the quarter being analyzed may not exactly match the set of flights that were predicted to take off in that same quarter. Information regarding downstream restrictions affecting individual flights is not available in this ASPM database used in this research. Hence it is not possible to account for passing of aircrafts in the taxi-out time prediction model. The accuracy of predicted average taxi-out time for a specified time interval of day, which indicates behavior of the airport, is estimated. An analysis of this type is extremely useful in predicting average airport taxi-out time trends approximately 30-60 minutes in advance of the given time of day (specifying the take off quarter). The percentage of time during the day that the average predicted taxi-times matches the average actual taxi-out times within 4 minutes is computed. For each of the days in the test period, the prediction accuracy of the algorithm for the average in 15 minute intervals of the day is shown in the last row of Figures 4.29 - 4.46. Confidence intervals at a level of significance of 95 % were also computed for the average prediction accuracy, in a manner similar to that computed for the individual flight prediction accuracy. These results are seen in Figure 4.12.

In general, the prediction accuracy for the months between March-August is greater than the prediction accuracy for the months between September-February. In addition, within the similar range of months there is a drop in the prediction accuracy between the years 2006 and 2007, 2008. As discussed earlier, the trends in the demand over these years has not changed significantly though the trends in the daily means and variance of taxi-out times shows a shift across the three years of study. It is also possible that the construction

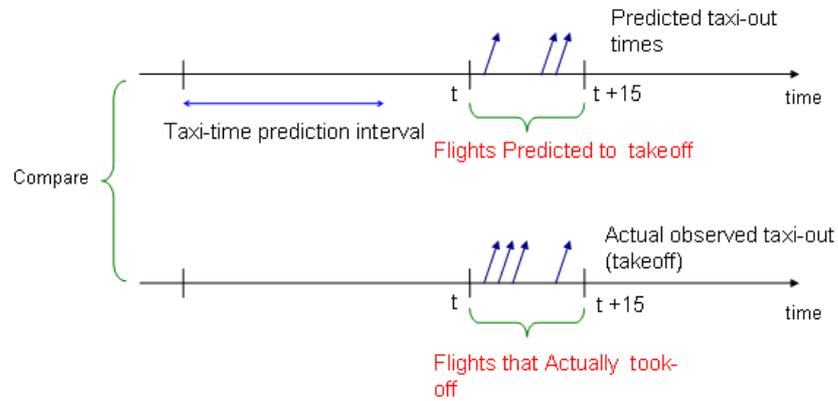


Figure 4.10: Average taxi-out time in 15 minute intervals of the day

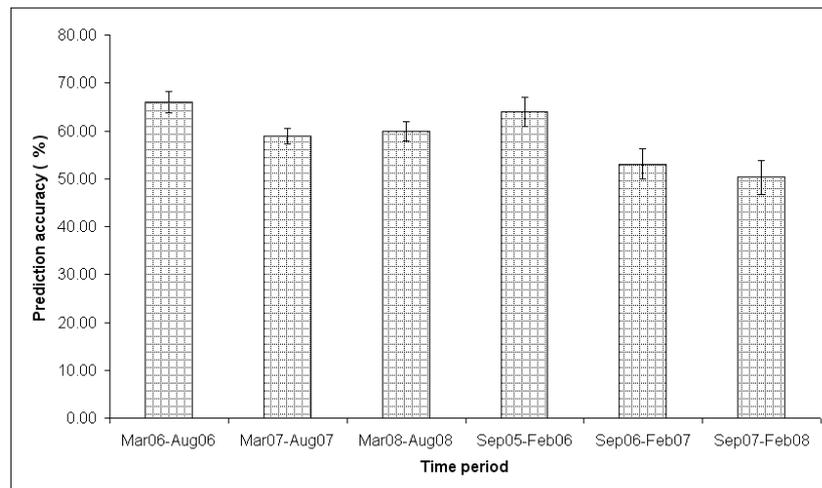


Figure 4.11: Q-learning algorithm (DTW: Individual flights)

of the new terminal at DTW over the year 2008 may have had an effect on the operations at the airport. The impact of this aspect is not immediately quantifiable.

Further, even within each time range of study, the prediction accuracy was higher for some days than for others. This requires an analysis of daily taxi-out time trends over the time of day. Apart from the airport system state that each individual flight experiences due to demand and congestion, several other factors unique to a particular flight influences the amount of time taken by the flight to taxi-out. These include downstream restrictions such as a Ground Stop, or a Miles-In-Trail restriction for flights intending to depart on

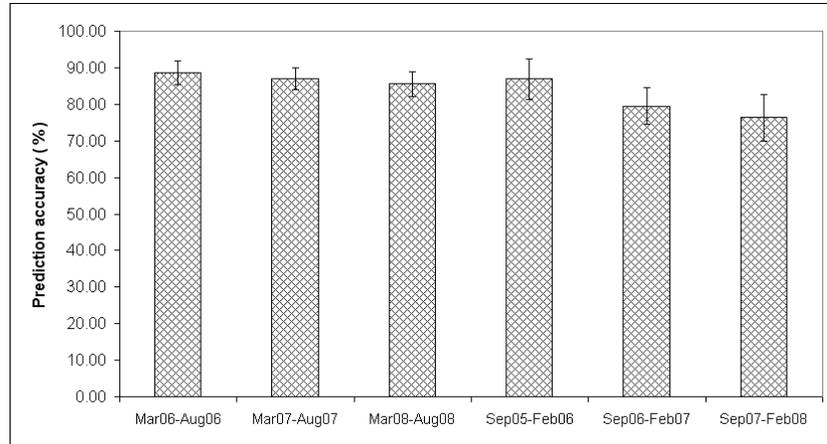


Figure 4.12: Q-learning algorithm (DTW:Averages)

a certain departure fix. Proximity of the departure runway to the terminal, and possible intersections of taxiways and runways are also factors that significantly impact the taxi-out time of a flight (refer Figure 2.5 for a detailed list of other factors influencing taxi-out time). The ASPM database that is used in this research does not provide information on the downstream restrictions imposed on individual flights or the specific runway used for departure. In the absence of this information, to the extent possible, an attempt is made to interpret the prediction results based on either physical phenomena such as weather, or based on a functional analysis of the taxi-out time behavior across the time of day. For this purpose, certain test days were selected from each of the six time periods of study. Two days were represented in this analysis, to include days for which the prediction accuracy could be considered low and high.

As expected, for the days when prediction accuracy was low, the taxi-out times across the day exhibited a high amount of variance relative to most other days in the same time period. This is easily seen by observing the plots of actual taxi-out time behavior across all the days in each time period of study (refer Figures 4.2 - 4.5). For example, the prediction accuracy for July 27, 2006 was low (about 50 %). Figure 4.2 indicates that the standard deviation of taxi-out times for this day was about 10 minutes, which is higher than for most of the other days in the same test period. Similarly, Figure 4.3 also indicates that July 27,

2006 had a higher taxi-out time mean (about 23 minutes) when compared to other days in the same test period.

Further, the airport behavior for each of these days was studied in detail by obtaining a scatter plot of actual taxi-out times of flights with respect to their scheduled gate-pushback times. These taxi-out time values, along with the average actual taxi-out time per quarter, were plotted against the total scheduled and actual demand (both pushback and takeoff) per quarter. In Figures 4.47 - 4.64, the ‘+’ denotes scheduled demand, and the ‘◇’ denotes actual demand. It is not unusual to see a wave-like pattern in the time-periods over which the scheduled demand exceeds actual demand or vice-versa. If the scheduled pushback demand in a quarter is greater than the actual pushback demand (due to either pushback delays or ATC operational procedures), then there is a spill-over of the excess demand into the next two or three quarters. This unscheduled shift in the demand would also cause the pattern of actual taxi-out times to change. It is then likely that the taxi-out time prediction made in advance of scheduled pushback will be based on a different system state than what will actually be experienced by the flight. For example, consider the airport behavior on July 27, 2006 shown in Figure 4.49. The taxi-out times are steady for the first part of the day, but start to increase after about 15:00 hours (local time). It is noted that the spill-over of scheduled demand (both pushback and departure) into the immediately following quarters is a more frequent occurrence after 15:00 hours. The prediction accuracy documented on an hour-by-hour basis also indicates that the prediction accuracy drops soon after 15:00 hours (Figure 4.31). The graph shown in Figure 4.50 compares the prediction accuracy of average taxi-out times in 15 minute intervals of the day. In accordance with the observations above, it is seen that the RL algorithm captures the actual taxi-out time behavior well up to 15:00 hours, and fails to more accurately do so for the evening hours. Similar observations are made for the other days in the test period for which prediction accuracy was low.

As mentioned earlier, another cause for excess taxi-out times is severe weather. Especially in winter, DTW airport is prone to icing conditions, which will require flights to taxi to and hold at a de-icing pad on the surface of the airport before joining the runway

queue. This causes large increases in taxi-out times and additionally influences the variance of taxi-out times across the day. De-icing events at an airport region may be inferred from temperature, precipitation and visibility recordings. Reports generated at Sensis Corporation (www.sensis.com) based on their AEROBAHN® airport automation tool indicate that the following days between December 2007 and February 2008 had de-icing events: 12-05-07, 12-15-07, 01-15-08, 01-22-08, 01-23-08, 02-01-08.

For the purpose of understanding the developed taxi-out time estimation algorithm, these days were included in the test phase of the implementation. The results indicate that the prediction accuracy for these days were very low when compared with the other days in the same time period. It is observed that the trend in the daily means of taxi-out times display a greater level of fluctuation over the Fall-Winter months than in the case of the Spring-Summer months (refer Figures 4.2 - 4.5). Weather conditions are a very likely explanation for these observations. Figure 4.13 shows the actual behavior of the departure process at DTW airport on December 5, 2007. It is observed that during the period between 6:00 and 8:00 in the morning, the actual gate pushback demand per quarter (‘◇’) matched almost exactly the scheduled gate pushback demand (‘+’). The trend in the actual takeoff demand in the same two hour time interval does not match the scheduled takeoff demand, which manifests itself as high taxi-out times (over 50 min) observed in this time range. This mismatch in the takeoff demand is likely to have been caused by de-icing requirements for the aircraft in view of poor weather conditions.

4.2.2 Results Based on Value Function Approximation

The taxi-out time estimation model was solved again using the diffusion wavelet based value function approximation method described in Algorithm 8. The prediction accuracy results for all the time periods of study for the individual flight analysis is shown in Figure 4.14.

A comparison of these results with those obtained using the state-of-the-art Q-learning method (refer Figure 4.15) indicate that in terms of accuracy the diffusion wavelet based value function approximation method was effective as a solution approach to the taxi-out

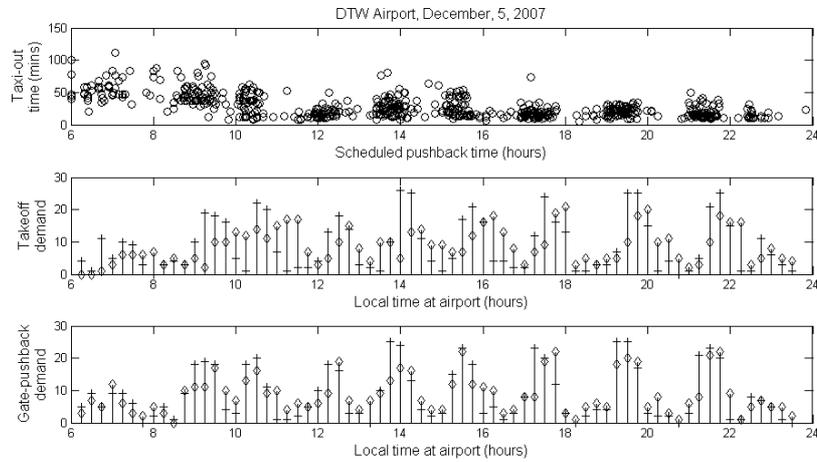


Figure 4.13: Airport behavior (DTW:December 5, 2007)

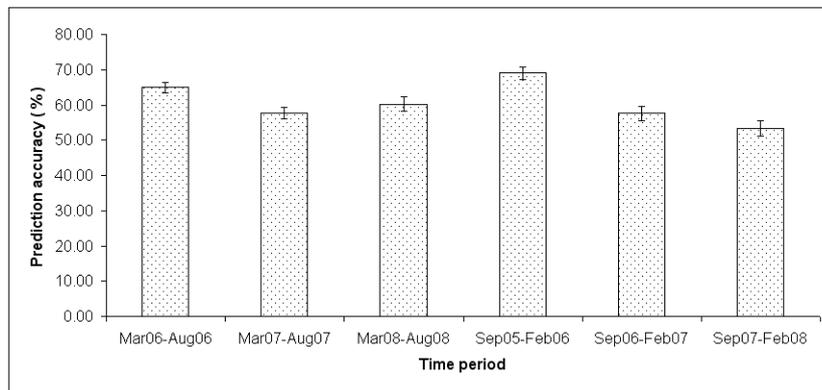


Figure 4.14: Diffusion wavelet based algorithm (DTW: Individual flights)

time estimation model. Based on Algorithm 8, a sample size of approximately 150 to 200 value functions, the corresponding state vectors, basis functions and scaling coefficients were stored at the end of the learning phase. This reduces the storage need from a 3600 element matrix (for the Q-learning approach) to approximately 600 elements. The computational benefit of this compression is hypothesized to be greater for problems of larger scale and will depend on choice of sample size.

A regression based value function approximation method with linear basis functions denoted by the three state variables s_1, s_2 , and s_3 was also applied as a solution strategy. The individual flight prediction accuracy is summarized in Figure 4.16. It is observed that

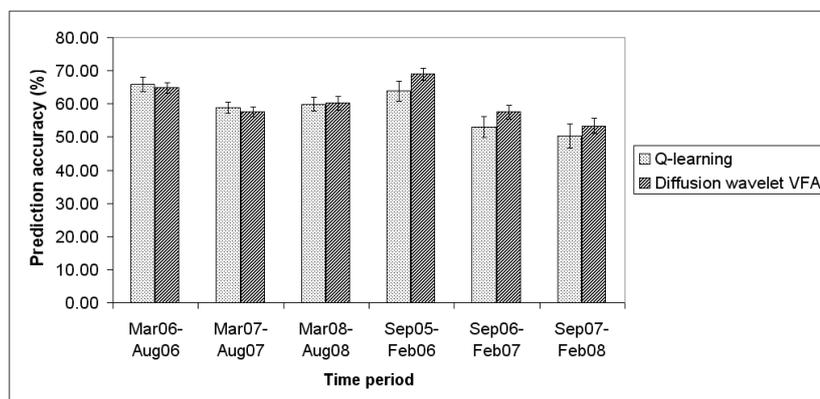


Figure 4.15: Q-learning and Diffusion wavelet based algorithm (DTW: Individual flights)

the regression approach is an efficient solution method for the taxi-out time estimation model considered. This is to be expected since the state space for the problem has a well defined linear structure.

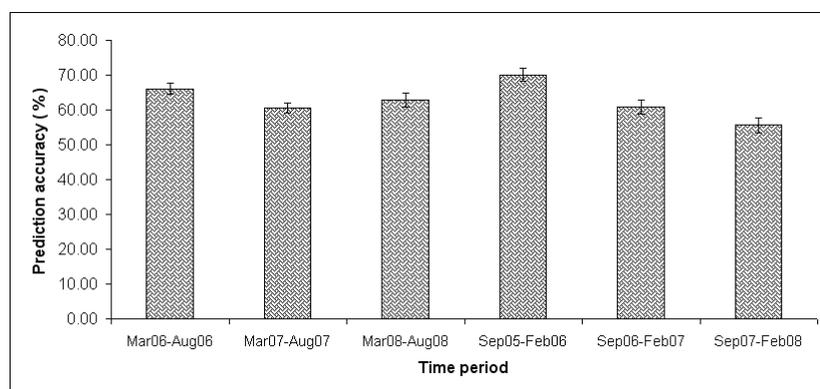


Figure 4.16: Regression approximation based algorithm (DTW: Individual flights)

A comparison of prediction accuracy results of the diffusion wavelet based algorithm with the two state-of-the-art methods is shown in Figure 4.17. The diffusion wavelet based algorithm performs almost as well the linear value function based algorithm. It is often the case in ADP applications that it may never be known in advance if a choice of more general basis functions is needed over other simpler strategies for approximation [Powell, 2007]. For a problem with a more general state space structure, the diffusion wavelet based

value function approximation scheme provides a method to extract good basis functions by exploiting the structure of the state space.

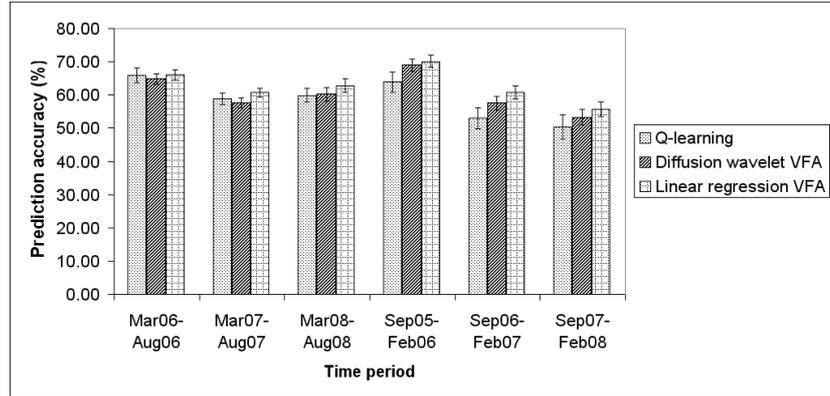


Figure 4.17: Q-learning and value function approximation based methods (DTW: Individual flights)

An additional aspect in ADP applications is related to the problem of interpolation. Reinforcement learning is a simulation-based algorithm. It may happen then that states not visited in the learning phase may be visited during the learnt or test phase. Since a value associated with being in these states is not available, an interpolation method is required to estimate the value of being in these new states. A commonly used approach is to use a measure of distance to identify k nearest neighbors of each of these states whose value functions are known [Gosavi, 2003]. A weighted average of these known values is then assigned as the value of being in a newly visited state. An important feature of the diffusion wavelet based scheme is that an effective interpolation scheme is inbuilt in the basis function extraction procedure. Referring back to the diffusion wavelet decomposition scheme described in Algorithm 5, Equation 3.12 computes the distance between state pairs based on a Gaussian kernel which relates to a random walk such that the probability of transition between states closer to each other is higher than if they are farther apart. This eliminates the need for a separate interpolation scheme and is a salient benefit when working with multi-dimensional state spaces where quantifying distance between states may not be intuitive.

The prediction accuracy results depicted in Figure 4.17 shows that both value function approximation methods at times perform better when compared to the Q-learning approach. Based on the discussion above, it is conjectured that the increase in accuracy may arise due to the interpolation scheme that is efficient in exploiting the structure of the state space. Noise in the approximation is also a factor that cannot be ignored in interpreting the results. An exact quantification of the role of interpolation and approximation noise on the accuracy of the results will not be straightforward.

4.2.3 Taxi-Out Time Estimation Results Based on a Regression Model

A regression model was developed based on the same system states s_1 , s_2 , and s_3 used for the reinforcement learning model. The form of the regression model is shown in Equation 4.1.

$$TO(s) = \beta_1 \cdot s_1 + \beta_2 \cdot s_2 + \beta_3 \cdot s_3 \quad s_1 \neq 0, s_2 \neq 0, s_3 \neq 0 \quad (4.1)$$

where, β_1, β_2 and β_3 are the regression coefficients. The constraints on the state variables being nonzero are laid because the model does not account for nonzero nominal taxi-out times under no congestion conditions. Unlike the reinforcement learning implementation the values assumed by the variables s_1, s_2 and s_3 are continuous.

The same sets of data and test days were considered for fitting the regression model based on the least squares fit, and for computing the prediction accuracy. The results are summarized in Figure 4.18. A comparison of the results from the Q-learning approach (Figure 4.11) indicates a 6 % to 12 % increase in accuracy of the reinforcement learning method over the regression model. It is expected that with the availability of more detailed information on runway and terminal used for departure or other categorical variables influencing taxi-out times, the reinforcement learning approach would provide a means to effectively model and capture these nonlinear relationships. Additionally, the ADP approach provides a sequential prediction capability that may also be extended to aid in departure planning and decision making based on the MDP structure of the problem.

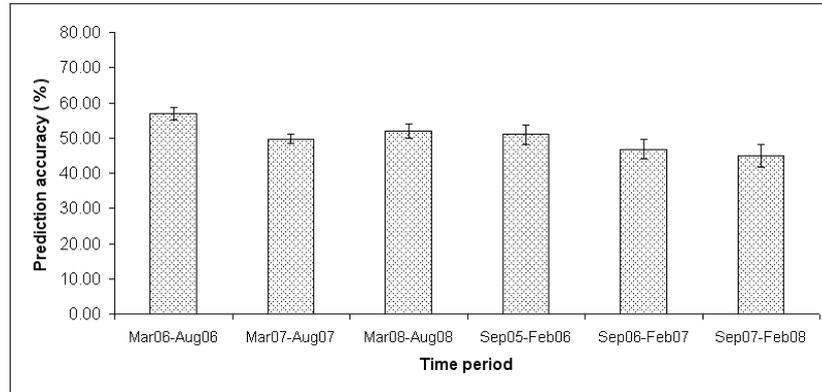


Figure 4.18: Taxi-out time estimation based on regression (DTW: Individual flights)

Figure 4.19 shows a comparison between the prediction accuracy for the following four methods compared:

1. Q-learning
2. Diffusion wavelet based value function approximation
3. Linear regression based value function approximation
4. Regression model for taxi-out time prediction

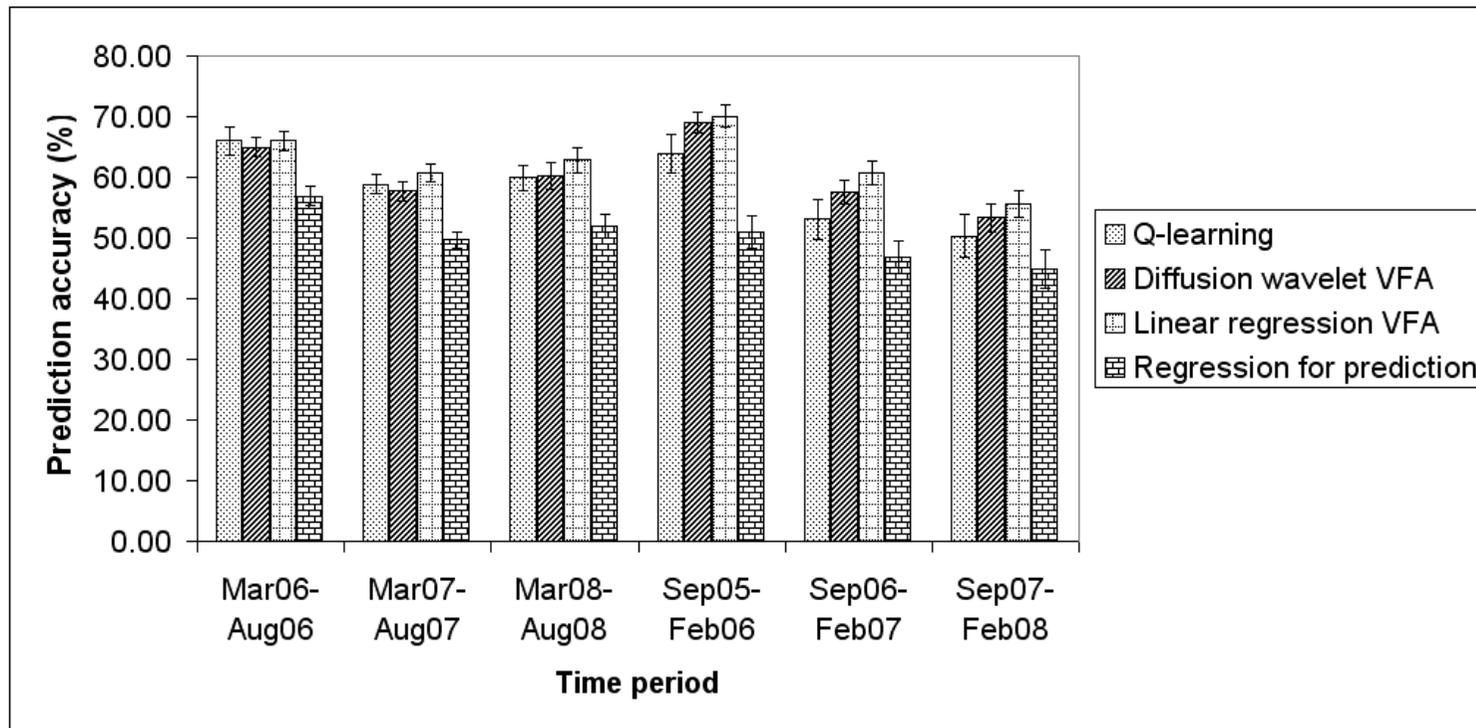


Figure 4.19: Comparison of prediction accuracy based on different methods (DTW: Individual flights)

4.3 Algorithm Extension for John F. Kennedy International Airport (JFK)

John F. Kennedy International Airport (JFK) is one of three major airports in the New York metropolitan area. It is the hub for JetBlue Airways, Delta, and American Airlines. In addition, several international carriers operate to and from JFK International Airport. The airport has eight terminals, servicing over ninety different domestic and international airlines [Source, 2009]. The airport diagram for the airport is shown in Figure 4.21. The most commonly used runway configuration involves mixing departures and arrivals on runway 31L, with runway 31R used exclusively for arrivals.

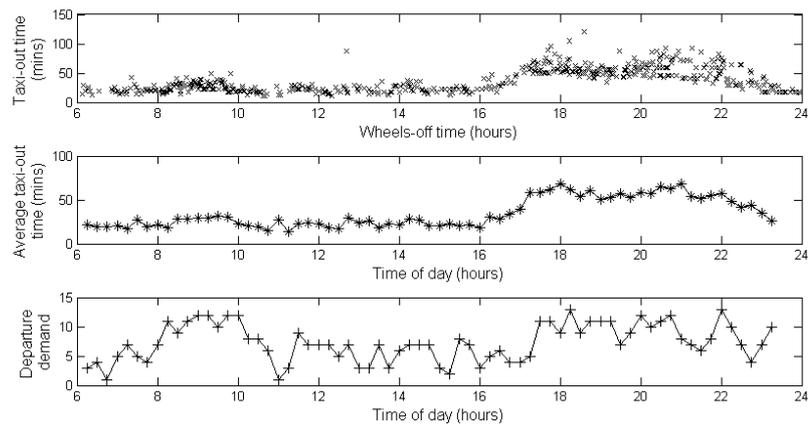


Figure 4.20: Actual taxi-out times and demand, JFK, Dec 4th, 2007

To understand the dynamics at JFK airport, first a discussion of the actual observed departure behavior at the airport is presented. The actual taxi-out times for a single day (December 4th, 2007) experienced by individual flights is represented via a scatter plot (Figure 4.20 top), so as to observe the range and behavior of taxi-out times at the airport. In addition, the average actual (Figure 4.20 middle) taxi-out times in 15 minute intervals of the day were plotted with respect to the time of day. Also, the actual demand per quarter (based on takeoff time of the aircraft) was plotted across all time intervals of the day (refer Figure 4.20 bottom). When the average demand in a quarter increases, correspondingly

the average actual taxi-out times also increase as observed by the peaks in Figure 4.20. A considerable increase in taxi-out times during the morning hours between 7 : 00 A.M and 10 : 00 A.M and during the hours after 4 : 00 P.M was observed. The taxi-out times after 4 : 00 P.M range between 20 min and 130 minutes (refer Figure 4.20 top, for example) which significantly increases the variance across the entire day. The high mean and variance in taxi-out time poses a considerable challenge to its prediction. Similar plots across several days suggested that this is a daily phenomenon seen at JFK airport. An article in Aviation Week [Compart, 2008] also highlights the deterioration in on-time departure rates at the NY airports during the late-afternoons and evenings. A likely cause for this is the cascading effect of taxi-out times across the day combined with an increase in the number of scheduled international flights in the evening with destinations in Europe. These sharp variations in demand and hence taxi-out times during the day make it challenging to track the state of the system and capture the trend in taxi-out times for the purpose of prediction.

In contrast to JFK airport, DTW airport has considerably lower taxi-out time means and standard deviations. For instance, the mean of individual taxi-out times at DTW is about 15 min with a standard deviation as low as 4-5 minutes. It was found that the earlier version of the RL model (described in Section 3.2.2) was not sufficient to capture the dynamics of an airport operating with very high taxi-out times and variance, such as JFK for which the taxi-out times after 4:00 P.M range between 20 minutes to over 130 minutes. The state space for the RL model was expanded by implementing a finer discretization of state variables (refer Section 3.2.3). A new variable capturing the time spent by a flight in the runway queue was introduced, to replace the variable representing the queue length in the earlier version of the model. As mentioned earlier, the ASPM database does not capture much of the dynamics between the gate pushback and takeoff events. Initially, the nominal (or unimpeded) taxi-out time was used as an indicator for whether or not an aircraft is in the runway queue (queues were split equally in the case of multiple runways). This nominal taxi-out time however is a seasonal average provided for a specified carrier across all runway configurations, rather than being specific to an individual flight (refer

Appendix B for a discussion on computing nominal taxi times). With the introduction of the runway ‘queue time’ variable (we define time spent by a flight in the runway queue as Actual taxi-out - unimpeded taxi-out), the effect of individual flight’s actual taxi-out times is captured in the system state of the flight. This was found to enhance the quality of the learning phase of the RL algorithm. In the learnt (testing) phase, since actual taxi-out times are not available at the time of prediction to compute ‘queue time’, the average of actual taxi-out times of flights within the last 30 minutes from current clock time is taken as the best available estimate.

A fourth variable capturing the time of day was included for the JFK taxi-out time model. Due to the corresponding variations in taxi-out time with departure demand, the average actual taxi-out time in a 45 minute interval prior to current time was also introduced as a state variable. These additional variables denoted by s_4 and s_5 respectively, further distinguish two flights that have same values for s_1 , s_2 , and s_3 but very different taxi-out times.

To study the prediction accuracy of the taxi-out time estimation model, the algorithm was tested using data from May 2008-October 2008. The prediction accuracy was computed for 42 test days across the six months. These prediction results on an hourly basis are shown in Figure 4.22, which also indicates that the prediction accuracy drops in the period after 4:00 P.M when demand is significantly high.

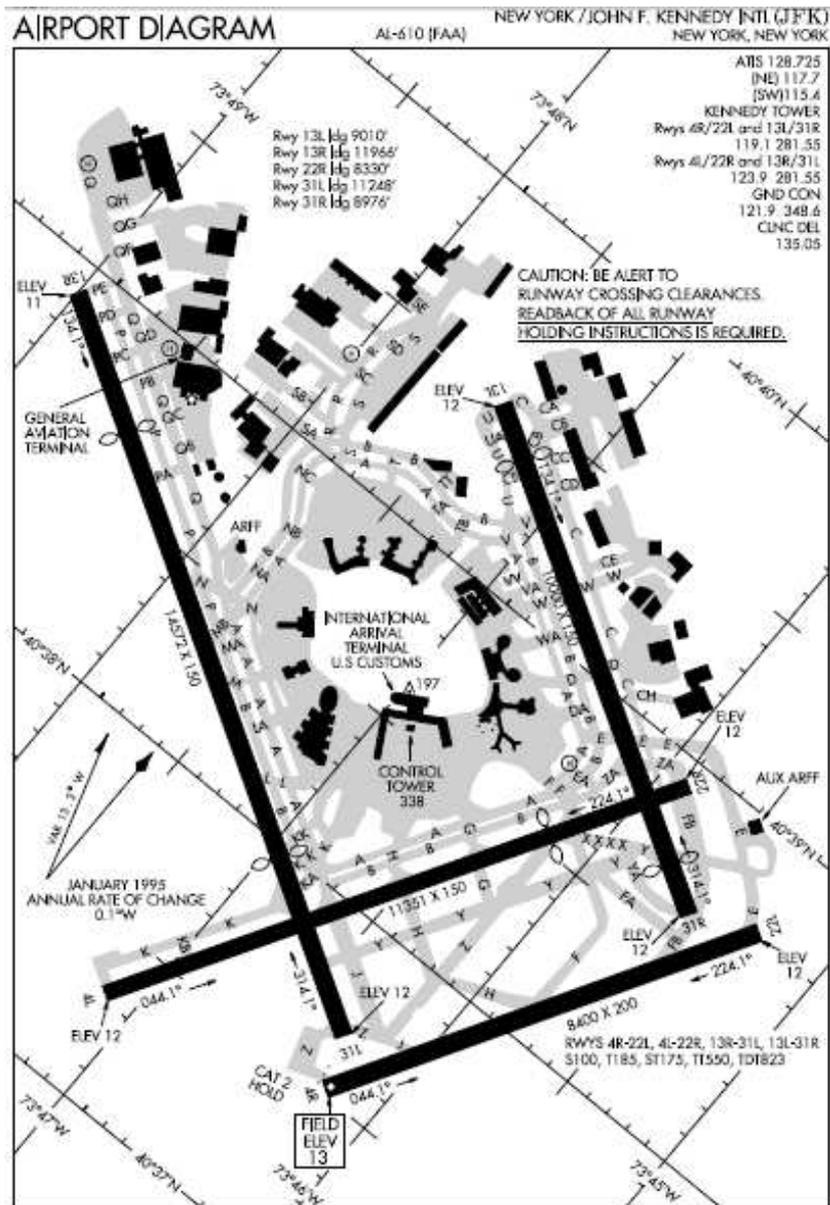


Figure 4.21: Airport Diagram (JFK).

Date	5/8	5/23	6/4	6/19	7/5	7/28	8/14	8/30	9/1	9/3	10/10	10/15	
Day	1	2	3	4	5	6	7	8	9	10	11	12	Mean
Time													
6:00-7:00	71	83	67	76	73	71	68	54	71	70	54	59	68
7:00-8:00	48	42	0	41	38	65	49	21	44	69	52	65	44
8:00-9:00	60	8	6	28	23	41	50	12	16	53	51	65	34
9:00-10:00	72	0	19	47	6	40	67	11	61	71	73	70	45
10:00-11:00	83	11	40	50	23	59	65	35	76	59	76	88	55
11:00-12:00	68	19	32	48	45	26	59	40	59	38	60	100	50
12:00-13:00	60	56	56	76	54	34	28	29	57	73	38	53	51
13:00-14:00	78	65	50	44	40	28	3	31	9	68	15	23	38
14:00-15:00	30	18	9	34	4	11	0	0	20	52	29	13	18
15:00-16:00	6	0	4	30	0	6	0	18	34	23	14	0	11
16:00-17:00	11	4	5	4	0	5	0	32	5	38	42	24	14
17:00-18:00	16	9	3	0	10	17	38	58	8	55	40	39	24
18:00-19:00	23	0	0	0	9	28	35	53	24	79	44	66	30
19:00-20:00	10	11	4	0	3	22	18	41	24	62	32	45	23
20:00-21:00	48	19	16	14	0	27	31	27	37	43	39	45	29
21:00-22:00	44	24	16	28	0	34	3	17	17	39	32	46	25
22:00-23:00	50	36	33	52	41	21	13	27	18	40	34	23	32
23:00-24:00	80	30	80	80	73	67	50	60	83	91	83	47	69
Individual Flt.	43	23	19	37	21	30	34	30	31	55	43	45	34
Average in 15 min	65	14	6	42	19	52	46	38	30	80	68	62	44

Figure 4.22: Prediction accuracy (JFK) within 4 min.

Further insight into the airport behavior was obtained for JFK airport using the AEROBAHN® automation tool developed at Sensis Corporation (www.sensis.com). AEROBAHN® uses detailed surface surveillance data from radar tracking systems as inputs and generates various airport performance reports along with a graphical interface to view the movement of aircrafts on the ground and in the immediate surrounding airspace. Visual observation of the airport and a side-by-side comparison with the data available in ASPM highlighted the following situations that are specific to individual flights. It must be noted that these observations were used only to understand seeming abnormalities in the taxi-out times of specific flights on certain days.

1. A change in runway configuration causes large delays - about a 20 min hold on all departing flights in queue.
2. There is a rare possibility that a flight reaches the runway threshold, and rather than takeoff, returns to the gate and re-taxis to the runway after a considerable amount of time.
3. Some times, close to the runway, different taxi paths are allotted to different flights - this results in some flights moving ahead of others in queue position (this is especially true when flights taxi to runway 4R).
4. Some times, as flights are queueing up for departure at the end of runway 31L, other flights that probably need a shorter stretch of runway are cleared for takeoff from a taxiway intersection. They hence do not experience a hold at the runway queue.
5. At times departures are on hold while an arriving flight lands on the same runway.

As for DTW airport, the prediction accuracy for JFK airport was computed based on the four solution approaches shown in Figure 4.9. As discussed earlier in the description of airport performance, the taxi-out time mean and variance at JFK is almost twice that for DTW airport. A prediction accuracy within a root mean square error of 4 minutes (in

contrast to 2 minutes for DTW) was chosen as a reasonable target. Figure 4.23 indicates that the prediction accuracy based on the Q-learning algorithm is close to 50 %.

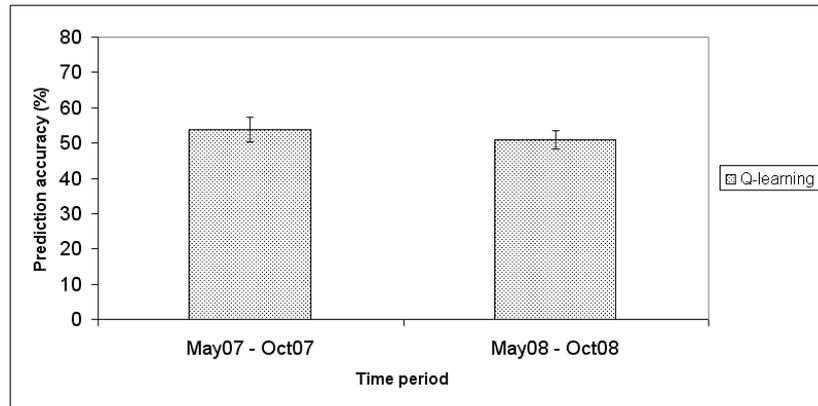


Figure 4.23: Prediction accuracy based on Q-learning algorithm (JFK: Individual flights)

Figures 4.24, 4.25, and 4.26 compare the accuracy performance of the four approaches. The observations on algorithm performance in terms of accuracy are similar to those made for DTW. Figure 4.24 shows that the prediction accuracy based on the diffusion wavelet approximation method falls a little short of the accuracy obtained through the explicit storage Q-learning approach. As for DTW, the structure of the state space for JFK airport is linear. The prediction accuracy based on the choice of linear basis functions will match the results from the Q-learning approach well. This is seen in Figure 4.25. A final comparison seen in Figure 4.26 shows results from the three ADP models and a regression based model for taxi-out time estimation. It is seen that the ADP model was consistent in outperforming the relatively simpler regression based model. This is an indication that the ADP models are more effective in capturing the dynamics of the airport operations.

4.4 Study of Algorithm Complexity

The discussion so far examined the different solution methods in terms of accuracy. An analysis from the perspective of computational storage and time is also of interest when

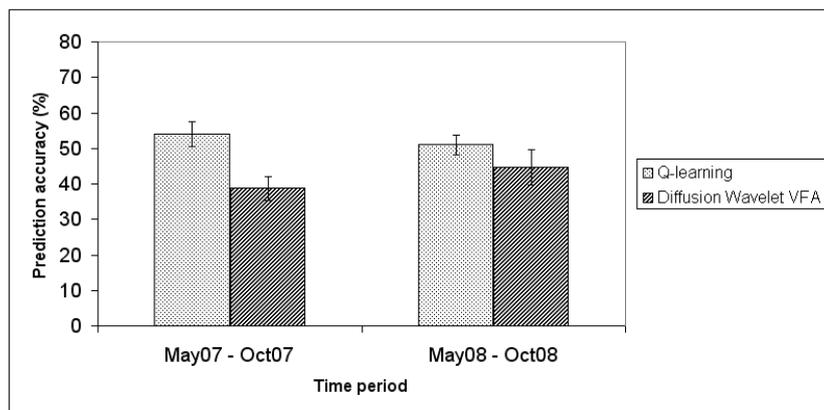


Figure 4.24: Q-learning and diffusion wavelet based algorithm (JFK: Individual flights)

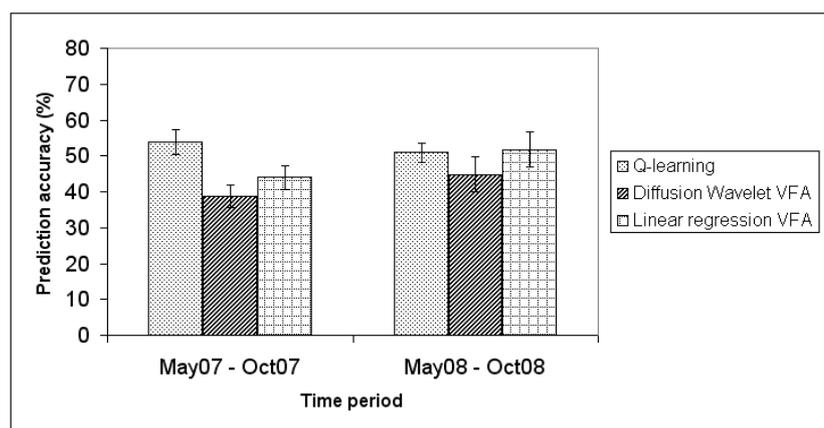


Figure 4.25: Q-learning and value function approximation based algorithms (JFK: Individual flights)

addressing the “curse of dimensionality” in ADP applications. A study on algorithm complexity investigates the scalability of the algorithm in terms of storage and computation time as the size of the input (in this case size of the state space) increases. The state space variables for DTW airport are discretized as follows:

1. s_1 , the number of taxiing arrivals: 0 - 27 in steps of 2, and > 27 resulting in 15 possible state values.
2. s_2 , the number of taxiing departures: 0 - 27 in steps of 2, and > 27 resulting in 15 possible state values.

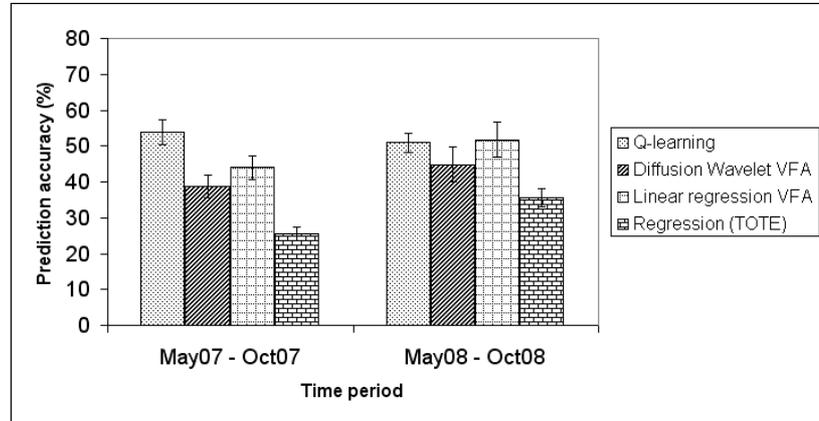


Figure 4.26: Prediction accuracy based on the four solution methods (JFK: Individual flights)

3. s_3 , the number aircrafts in the runway queue: 0 - 15 in steps of 1, resulting in 16 possible state values.

The total number of state combinations with this discretization is $15 \cdot 15 \cdot 16 = 3600$. The discretization of state variables for the JFK airport model results in a total of 216,000 possible states as detailed below.

1. s_1 , the number of taxiing arrivals: 0 - 32 in steps of 3, and > 32 resulting in 12 possible state values.
2. s_2 , the number of taxiing departures: 0 - 32 in steps of 3, and > 32 resulting in 12 possible state values.
3. s_3 , the number aircrafts in the runway queue: 0 - 27 in steps of 2, and > 27 resulting in 15 possible state values.
4. s_4 , the time index: 04:00 hrs to 24:00 hrs in steps of 1 hour resulting in 15 possible state values.
5. s_5 , the average taxi-out time 45 min before current time: < 25 min, 25 - 40 min in steps of 5, > 40 , resulting in 5 possible state values.

The question of computational feasibility arises in both the learning phase and the learnt phase of the ADP algorithms. In the learning phase the value of being in state S^n is updated at each iteration. For the Q-learning algorithm, this requires accessing the last updated value of being in state S^n and the last updated value of being in the next state S^{n+1} (refer Equation 2.23 in Algorithm 3). The effect of an increased number of action combinations is ignored at this time since the number of actions in the taxi-out time estimation model is simply the number of possible taxi-out time prediction values and this number may be considered small (taxi-out time predictions are assumed to range from 5 min to 65 min in steps of one for DTW, and between 5 min and 115 min for JFK). The amount of time taken to access the appropriate elements depends on the position of the state in the Q-factor matrix. Assuming that accessing a single element takes unit time, the worst-case scenario for accessing the value of being in a state is 3600 time units for DTW and 216000 time units for JFK.

In the learning phase of the diffusion wavelet based value function approximation method, when the value of being in state S^n is to be updated, two situations arise. If state S^{n+1} belongs to the set of samples S_{N_s} that was updated in the previous iteration, then the worst-case time to access the appropriate element equals the cardinality of the sample set. Alternately, if state $S^{n+1} \notin S_{N_s}$, the computation time involves appending state S_{n+1} to the sample set, deriving the graph of the state space, computing the diffusion operator, obtaining the basis functions based on the QR -factorization technique, and convoluting the coefficients with the basis functions (these are the steps described in Algorithm 6).

The complexity was also studied based on the experiments run for DTW and JFK. Graphs of computation time for the learning phase and the learnt phase are shown in Figures 4.27 and 4.28 respectively. The results are based on a learning period of 6 months, and a testing phase of 42 days for each of the airports. The first observation is that the Q-learning approach worked better than the diffusion wavelet approach for DTW for which the input size is 3600. The benefit of the diffusion wavelet value function approximation scheme is seen in the results for JFK which has an input size of 216,000. This supports the notion

that when the state space of the problem is relatively small, an explicit storage approach will be reasonably efficient. Also, the diffusion wavelet based value function approximation method involves more algorithmic steps than the Q-learning approach. There will likely be a trade-off point based on the input size, when the benefits of the diffusion wavelet value function approximation method will start to be realized.

In analyzing these graphs, one must be cautious to take note of a few subtle aspects. First, a direct comparison between the results for DTW and those for JFK is not intended. The algorithm for JFK represents a larger input size. However, due the nature of operations, the number of iterations at JFK was also higher. For the same reason, a linear extrapolation or interpolation of the computational benefits of the diffusion wavelet algorithm is not to be attempted. A second aspect is that the computation time of the diffusion wavelet based method will depend on the choice of sample size of the states stored at each iteration. In general, it can be hypothesized that the computational performance of the diffusion wavelet based value function approximation method will be flatter across varying input sizes and is more a function of sample size. The results of the Q-learning approach will vary depending more on input size and the location of states visited in the Q-matrix.

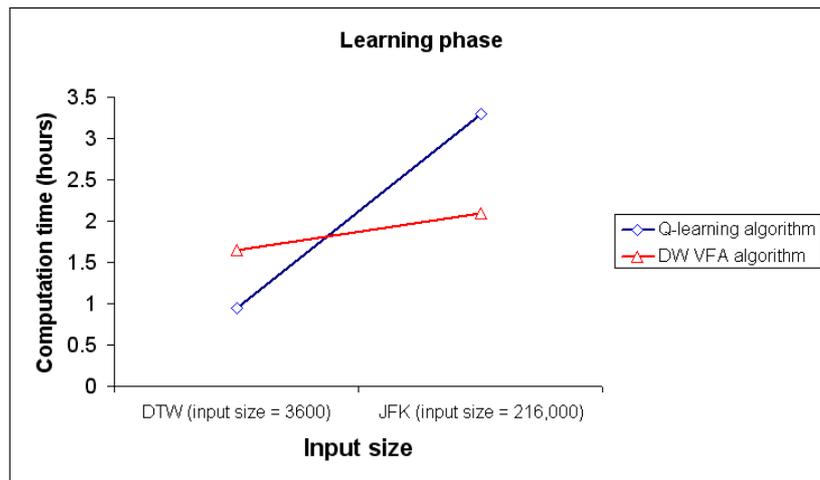


Figure 4.27: A comparison of computation times for the learning phase.

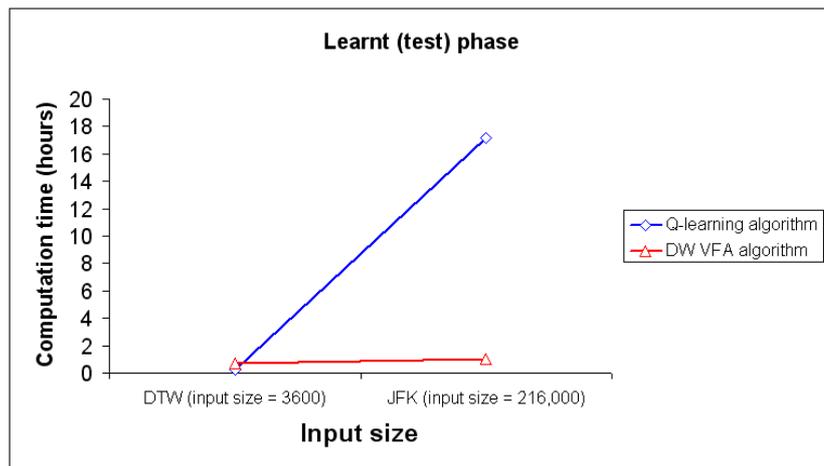


Figure 4.28: A comparison of computation times for the learnt phase.

Date	3/4	3/5	3/10	3/18	3/21	3/26	3/29	4/5	4/6	4/12	4/14	4/16	4/25	4/30
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time														
6:00-7:00	48	56	75	80	73	50	74	30	90	74	37	88	81	87
7:00-8:00	55	67	56	90	63	57	62	44	69	76	54	72	85	82
8:00-9:00	68	66	88	82	85	49	67	55	88	86	76	75	79	74
9:00-10:00	49	73	75	84	65	83	81	75	79	63	69	75	72	78
10:00-11:00	67	85	70	54	64	44	77	82	84	78	67	61	40	85
11:00-12:00	85	61	95	33	72	81	79	74	70	68	65	68	79	83
12:00-13:00	56	72	72	27	73	72	85	77	68	79	81	70	74	75
13:00-14:00	81	57	82	73	84	47	76	66	87	85	71	68	72	72
14:00-15:00	77	50	76	62	83	64	74	83	86	71	89	61	72	52
15:00-16:00	79	77	75	77	75	78	66	41	75	33	81	79	76	66
16:00-17:00	71	82	71	76	68	68	78	74	67	50	90	71	81	63
17:00-18:00	71	68	81	82	64	69	79	72	61	62	76	72	73	73
18:00-19:00	67	44	73	56	75	65	50	69	60	50	78	50	60	57
19:00-20:00	69	78	79	70	68	81	58	71	79	64	79	74	70	17
20:00-21:00	56	69	71	57	87	69	85	69	69	56	71	71	88	56
21:00-22:00	88	82	77	58	71	81	76	80	78	73	85	81	76	66
22:00-23:00	100	92	81	54	80	81	78	89	60	87	90	58	81	67
23:00-24:00	--	--	100	50	86	86	100	100	100	83	100	89	100	100
Individual Fit.	70	71	77	66	73	68	73	68	76	68	75	72	73	67
Avg. in 15 min	93	97	100	91	99	90	96	90	100	87	93	93	97	90

Figure 4.29: Daily % prediction accuracy within 4 minutes (DTW, Mar06-Apr06)

Date	5/2	5/21	5/26	5/27	5/29	5/30	5/31	6/2	6/6	6/13	6/21	6/22	6/23	6/24
Day	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Time														
6:00-7:00	50	76	83	75	77	68	77	85	75	71	57	79	77	89
7:00-8:00	91	81	63	87	78	83	60	87	87	100	23	100	100	33
8:00-9:00	71	78	77	92	61	68	82	83	87	90	27	83	90	76
9:00-10:00	83	77	78	78	74	79	85	96	100	82	93	67	71	70
10:00-11:00	75	85	74	88	82	72	69	94	82	52	94	67	42	64
11:00-12:00	83	47	76	79	78	45	47	42	33	73	63	92	36	68
12:00-13:00	77	63	74	74	81	74	28	11	5	63	67	71	45	77
13:00-14:00	52	73	50	78	80	13	73	73	52	29	97	24	58	71
14:00-15:00	62	78	50	62	26	30	50	93	33	47	69	14	47	89
15:00-16:00	66	50	32	90	68	25	17	48	19	55	55	18	83	52
16:00-17:00	74	81	33	75	67	6	21	35	50	53	44	14	39	58
17:00-18:00	59	68	30	68	63	31	21	45	38	40	42	27	35	77
18:00-19:00	67	47	63	60	59	54	33	48	45	30	34	35	68	57
19:00-20:00	71	71	28	30	77	62	64	60	59	81	61	77	81	77
20:00-21:00	86	58	73	65	75	71	50	73	80	57	56	79	79	57
21:00-22:00	74	72	70	55	79	77	64	66	60	64	66	60	68	69
22:00-23:00	89	85	30	46	61	58	40	55	54	59	62	48	40	54
23:00-24:00	81	81	75	75	75	50	75	67	79	80	73	89	67	67
Individual Ft.	71	71	55	67	72	54	53	61	54	61	59	54	62	67
Avg. in 15 min	100	96	64	87	94	74	71	71	72	87	71	68	84	93

Figure 4.30: Daily % prediction accuracy within 4 minutes (DTW, May06-Jun06)

Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
Day	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Time														
6:00-7:00	100	72	81	65	76	72	65	78	65	75	83	79	69	65
7:00-8:00	74	80	70	73	84	65	72	70	75	89	83	83	88	89
8:00-9:00	82	78	81	64	82	68	97	83	83	90	91	75	65	80
9:00-10:00	69	73	63	65	85	74	82	76	76	75	78	39	70	69
10:00-11:00	80	74	65	74	85	72	86	85	77	82	64	78	63	74
11:00-12:00	68	82	63	74	79	60	47	58	42	60	78	53	59	88
12:00-13:00	60	78	64	76	70	64	20	63	36	67	83	53	64	85
13:00-14:00	53	47	74	35	87	48	43	67	66	71	81	74	53	75
14:00-15:00	29	47	84	29	75	71	74	77	70	83	57	90	74	75
15:00-16:00	71	72	75	63	81	22	15	72	50	81	80	65	76	50
16:00-17:00	55	67	68	50	82	29	40	81	82	64	68	83	58	53
17:00-18:00	69	67	73	53	75	71	40	75	56	70	78	76	58	49
18:00-19:00	47	56	75	72	44	50	65	85	69	63	38	38	39	33
19:00-20:00	80	59	69	76	74	38	56	62	73	66	38	70	51	39
20:00-21:00	73	31	78	50	62	0	71	38	79	50	88	76	40	46
21:00-22:00	72	71	76	41	73	24	64	70	73	69	76	77	40	30
22:00-23:00	50	42	69	43	62	21	47	55	73	45	51	57	67	60
23:00-24:00	75	78	50	100	100	100	100	89	83	100	71	67	71	71
Individual Fit.	68	66	72	59	76	50	56	71	67	71	72	69	61	58
Avg. in 15 min	93	97	100	83	100	58	84	96	88	100	99	97	91	90

Figure 4.31: Daily % prediction accuracy within 4 minutes (DTW, Jul06-Aug06)

Date	3/4	3/5	3/10	3/18	3/21	3/26	3/29	4/5	4/6	4/12	4/14	4/16	4/25	4/30
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time														
6:00-7:00	60	69	65	67	50	73	68	37	78	56	61	85	67	78
7:00-8:00	78	67	63	87	67	90	77	54	77	57	81	68	67	82
8:00-9:00	76	56	77	69	73	74	78	36	64	83	79	61	69	84
9:00-10:00	62	48	68	72	58	69	75	42	65	71	65	65	67	72
10:00-11:00	70	62	77	65	69	67	66	26	52	61	60	53	68	51
11:00-12:00	60	47	71	60	67	82	45	25	68	68	80	56	56	65
12:00-13:00	66	44	84	51	69	37	59	43	62	58	58	62	56	63
13:00-14:00	72	70	67	54	64	58	40	47	45	71	56	69	63	53
14:00-15:00	70	74	63	61	55	58	59	24	38	67	79	62	61	79
15:00-16:00	60	57	55	72	41	66	55	50	38	45	62	57	53	25
16:00-17:00	56	55	47	71	88	55	78	38	41	53	76	63	67	67
17:00-18:00	49	52	52	56	54	31	53	38	61	61	65	68	57	52
18:00-19:00	76	64	50	53	80	53	47	36	43	75	62	62	50	42
19:00-20:00	63	56	49	70	62	54	40	37	18	69	54	62	52	31
20:00-21:00	44	67	75	64	56	63	92	22	22	89	25	82	70	55
21:00-22:00	61	55	75	52	31	53	57	39	12	45	16	58	48	28
22:00-23:00	50	72	59	59	50	47	56	52	28	57	51	49	52	33
23:00-24:00	0	67	50	100	100	50	100	50	75	100	50	75	67	80
Individual Ft.	63	59	63	63	58	58	59	39	47	61	56	62	59	52
Avg. in 15 min	86	96	86	96	84	93	90	64	74	90	88	97	90	72

Figure 4.32: Daily % prediction accuracy within 4 minutes (DTW, Mar07-Apr07)

Date	5/2	5/21	5/26	5/27	5/29	5/30	5/31	6/2	6/6	6/13	6/21	6/22	6/23	6/24
Day	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Time														
6:00-7:00	77	77	64	64	52	84	65	64	72	68	72	85	76	68
7:00-8:00	82	87	44	85	68	55	88	72	81	88	91	59	89	61
8:00-9:00	78	56	70	82	69	72	50	74	64	66	65	78	79	58
9:00-10:00	65	67	56	64	56	58	37	62	68	65	59	83	60	67
10:00-11:00	76	65	61	62	56	56	51	78	46	62	63	53	75	61
11:00-12:00	72	80	67	60	53	71	71	65	53	39	61	61	69	65
12:00-13:00	65	66	52	12	63	55	54	54	57	61	59	65	77	67
13:00-14:00	54	43	50	25	33	57	68	54	21	43	19	52	55	65
14:00-15:00	56	50	33	26	42	71	72	53	40	50	21	67	60	65
15:00-16:00	62	64	65	60	58	51	56	42	70	53	52	63	59	60
16:00-17:00	61	75	69	69	45	55	67	50	47	44	47	74	53	81
17:00-18:00	22	75	51	64	42	24	72	47	53	60	57	34	56	54
18:00-19:00	67	75	67	60	62	67	75	64	36	50	73	75	60	50
19:00-20:00	38	45	32	55	74	70	60	54	71	68	60	50	73	56
20:00-21:00	64	64	50	61	62	83	89	64	75	67	45	75	83	89
21:00-22:00	60	69	61	65	65	58	62	55	78	56	34	75	46	63
22:00-23:00	44	78	81	78	58	76	74	67	73	67	64	74	78	70
23:00-24:00	100	33	33	33	50	67	90	91	71	50	63	73	58	64
Individual Flt.	58	63	55	56	56	59	63	60	59	59	54	63	64	63
Avg. in 15 min	71	96	81	81	80	93	91	88	93	93	84	94	100	97

Figure 4.33: Daily % prediction accuracy within 4 minutes (DTW, May07-Jun07)

Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
Day	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Time														
6:00-7:00	65	53	71	78	88	58	85	63	20	72	72	65	67	68
7:00-8:00	79	68	79	85	73	33	65	61	18	71	79	67	60	62
8:00-9:00	85	84	88	65	53	53	69	61	46	70	75	58	75	73
9:00-10:00	70	54	76	60	35	21	68	51	41	69	80	69	67	70
10:00-11:00	70	52	63	59	59	46	54	54	54	74	71	60	49	65
11:00-12:00	41	73	61	50	69	46	38	53	58	67	56	69	60	56
12:00-13:00	61	62	69	84	73	83	70	50	55	57	66	55	16	64
13:00-14:00	66	51	64	44	73	73	65	56	55	52	55	63	55	55
14:00-15:00	61	56	86	38	65	71	50	17	60	44	67	61	33	71
15:00-16:00	48	64	46	55	67	39	66	57	59	52	56	75	44	55
16:00-17:00	61	72	73	53	60	77	79	61	43	50	33	69	53	88
17:00-18:00	55	50	57	60	50	49	63	17	48	56	67	54	58	60
18:00-19:00	50	55	100	67	78	67	38	86	44	75	67	63	71	33
19:00-20:00	64	71	52	51	30	57	75	57	38	59	44	59	65	55
20:00-21:00	67	100	63	70	82	67	91	73	72	92	91	85	88	78
21:00-22:00	75	72	63	62	58	55	67	75	35	62	72	36	46	53
22:00-23:00	69	68	57	57	73	57	65	88	59	67	57	67	74	67
23:00-24:00	62	77	53	58	55	69	62	71	71	88	70	75	75	75
Individual Flt.	65	64	64	60	60	54	66	55	46	63	64	61	55	62
Avg. in 15 min	99	91	93	94	81	77	93	86	65	97	97	86	61	87

Figure 4.34: Daily % prediction accuracy within 4 minutes (DTW, Jul07-Aug07)

Date	3/4	3/5	3/10	3/18	3/21	3/26	3/29	4/5	4/6	4/12	4/14	4/16	4/25	4/30
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time														
6:00-7:00	48	12	55	36	73	71	36	54	75	86	83	77	69	87
7:00-8:00	71	0	56	48	71	75	57	32	75	83	53	74	76	81
8:00-9:00	75	16	54	70	67	69	41	69	73	88	80	86	65	85
9:00-10:00	66	15	65	54	85	68	26	67	61	71	78	68	39	70
10:00-11:00	62	24	65	70	59	67	57	65	68	62	71	75	60	81
11:00-12:00	69	26	63	57	82	71	64	73	53	93	56	50	60	76
12:00-13:00	63	39	63	60	75	70	71	72	65	71	73	63	46	73
13:00-14:00	50	46	58	55	58	59	67	62	61	62	64	55	69	48
14:00-15:00	50	42	61	56	47	53	61	69	82	72	75	59	59	30
15:00-16:00	39	58	57	59	11	54	69	58	58	67	63	57	52	65
16:00-17:00	36	75	60	29	0	64	57	64	69	71	69	100	67	71
17:00-18:00	9	61	76	56	14	70	55	50	55	58	63	66	60	60
18:00-19:00	0	29	50	63	27	88	100	25	100	50	63	88	50	56
19:00-20:00	5	62	61	43	42	67	75	75	45	72	64	62	67	48
20:00-21:00	0	60	20	20	40	60	86	75	80	75	88	71	63	88
21:00-22:00	0	65	49	26	25	49	48	44	44	51	66	43	69	55
22:00-23:00	0	57	65	35	27	73	42	62	66	50	58	75	65	71
23:00-24:00	--	80	0	100	100	100	100	100	100	100	100	100	100	50
Individual Flt.	44	47	60	50	46	64	56	59	59	65	68	65	62	64
Avg. in 15 min	64	57	87	72	72	93	84	81	84	96	96	94	80	86

Figure 4.35: Daily % prediction accuracy within 4 minutes (DTW, Mar08-Apr08)

Date	5/2	5/21	5/26	5/27	5/29	5/30	5/31	6/2	6/6	6/13	6/21	6/22	6/23	6/24
Day	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Time														
6:00-7:00	68	78	65	79	70	73	82	71	76	80	75	75	71	72
7:00-8:00	85	90	92	90	85	83	67	79	80	74	67	94	65	74
8:00-9:00	47	80	77	82	82	75	67	77	76	23	67	83	45	83
9:00-10:00	40	80	73	75	77	78	62	64	71	0	65	76	45	70
10:00-11:00	71	80	62	81	73	76	76	56	62	33	71	76	61	70
11:00-12:00	35	61	76	71	55	74	78	65	56	35	65	63	56	61
12:00-13:00	58	67	60	67	57	44	68	49	65	42	59	79	20	47
13:00-14:00	44	63	72	69	61	62	65	48	70	56	59	61	33	55
14:00-15:00	60	54	70	59	54	27	45	9	78	58	53	27	19	39
15:00-16:00	23	43	67	57	62	56	63	56	64	50	61	54	16	71
16:00-17:00	9	90	55	91	70	89	63	64	58	21	50	54	29	77
17:00-18:00	42	38	51	67	54	63	51	63	49	43	55	56	26	59
18:00-19:00	57	57	14	43	71	71	67	60	71	86	86	57	63	40
19:00-20:00	58	29	72	64	55	54	52	34	41	54	73	60	55	34
20:00-21:00	50	88	78	56	70	78	71	70	33	88	100	67	60	40
21:00-22:00	61	61	48	67	36	42	36	68	25	58	62	64	71	68
22:00-23:00	69	53	64	62	63	26	52	53	44	46	59	41	57	71
23:00-24:00	67	25	100	100	67	80	40	50	25	50	83	83	60	0
Individual Fit.	52	59	64	69	61	59	57	56	58	48	64	64	46	61
Avg. in 15 min	64	71	96	99	91	77	84	77	90	80	87	96	68	94

Figure 4.36: Daily % prediction accuracy within 4 minutes (DTW, May08-Jun08)

Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
Day	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Time														
6:00-7:00	93	88	64	62	89	75	64	83	77	85	86	81	83	64
7:00-8:00	80	80	55	83	68	71	60	63	74	74	63	83	73	44
8:00-9:00	65	78	45	44	83	79	60	88	78	72	83	78	79	72
9:00-10:00	59	89	49	48	76	70	69	74	72	74	80	78	78	84
10:00-11:00	62	78	47	70	70	73	66	74	78	77	80	70	78	82
11:00-12:00	78	82	50	44	47	79	67	60	53	63	84	67	89	61
12:00-13:00	45	77	41	38	51	80	70	76	56	74	70	62	61	54
13:00-14:00	51	72	54	62	48	67	72	57	46	76	70	77	52	58
14:00-15:00	35	69	50	61	23	53	29	64	75	53	63	56	56	72
15:00-16:00	62	86	58	58	52	70	66	71	62	71	69	73	83	71
16:00-17:00	58	78	67	69	78	80	62	78	36	62	64	73	67	71
17:00-18:00	59	71	75	70	60	63	67	77	21	77	65	64	67	67
18:00-19:00	67	40	57	86	40	75	43	50	0	67	67	50	43	86
19:00-20:00	28	45	40	63	65	53	61	57	45	64	70	78	48	68
20:00-21:00	0	44	86	33	83	14	86	100	80	83	83	67	75	56
21:00-22:00	49	54	56	46	60	61	55	46	61	71	62	49	55	53
22:00-23:00	27	57	68	50	46	57	61	45	64	65	44	45	44	65
23:00-24:00	0	0	50	75	40	60	67	67	75	100	80	60	100	100
Individual Ft.	55	67	54	58	61	66	64	64	58	72	70	66	65	67
Avg. in 15 min	81	100	72	84	86	97	93	100	88	99	97	87	96	97

Figure 4.37: Daily % prediction accuracy within 4 minutes (DTW, Jul08-Aug08)

Date	9/2	9/4	9/8	9/16	9/18	9/23	9/28	10/4	10/8	10/10	10/17	10/19	10/20	10/22
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time														
6:00-7:00	85	33	70	71	83	76	55	77	71	90	57	86	73	63
7:00-8:00	77	75	58	71	93	65	77	77	76	76	61	79	45	71
8:00-9:00	43	92	77	80	47	32	72	87	54	57	79	80	53	76
9:00-10:00	39	78	77	70	67	44	71	76	82	64	75	66	54	65
10:00-11:00	68	74	77	57	78	61	75	76	64	46	76	79	81	69
11:00-12:00	53	88	63	47	81	73	88	89	72	44	88	81	76	25
12:00-13:00	68	76	48	66	75	74	65	66	61	50	78	79	72	23
13:00-14:00	81	61	74	33	89	44	83	76	30	37	87	72	83	69
14:00-15:00	63	85	63	53	88	59	50	87	38	44	53	75	69	71
15:00-16:00	79	43	59	51	76	67	48	68	80	55	79	63	73	82
16:00-17:00	67	58	58	64	62	67	53	81	79	55	42	73	60	80
17:00-18:00	74	82	38	42	65	53	77	71	70	38	59	62	75	64
18:00-19:00	60	70	38	57	77	50	71	64	50	50	40	67	55	33
19:00-20:00	54	56	44	55	71	60	68	70	68	25	67	61	46	29
20:00-21:00	89	73	100	64	80	89	83	67	83	78	80	60	75	70
21:00-22:00	81	80	71	50	58	48	77	82	80	65	82	75	81	79
22:00-23:00	73	77	40	67	43	42	67	74	71	59	70	72	56	62
23:00-24:00	100	100	100	100	100	50	63	36	36	82	62	73	38	31
Individual Flt.	68	68	61	57	71	57	69	74	67	51	72	71	66	58
Avg. in 15 min	96	97	100	90	97	91	97	99	93	80	96	97	96	81

Figure 4.38: Daily % prediction accuracy within 4 minutes (DTW, Sep05-Oct05)

Date	11/5	11/8	11/10	11/17	11/18	11/24	11/28	12/4	12/5	12/12	12/15	12/18	12/29	12/30
Day	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Time														
6:00-7:00	73	68	84	70	75	0	81	0	75	52	8	13	70	79
7:00-8:00	75	70	86	42	64	0	62	11	72	22	8	53	58	68
8:00-9:00	83	68	82	32	79	16	71	4	82	59	0	76	69	73
9:00-10:00	81	71	82	45	67	27	78	14	81	72	0	56	77	63
10:00-11:00	78	64	77	50	62	42	73	22	64	69	0	78	75	81
11:00-12:00	44	94	80	93	50	24	71	58	75	80	0	67	69	85
12:00-13:00	44	74	73	67	71	24	58	52	61	62	0	83	40	59
13:00-14:00	69	54	41	66	81	25	69	78	67	78	0	69	69	63
14:00-15:00	76	23	40	72	73	36	50	63	56	76	10	77	59	65
15:00-16:00	74	67	72	80	59	34	62	67	74	70	8	84	70	66
16:00-17:00	35	76	65	60	73	81	65	50	47	63	0	92	87	86
17:00-18:00	27	58	60	74	72	75	55	78	46	66	5	72	58	63
18:00-19:00	13	36	50	57	71	86	38	29	54	67	10	64	60	70
19:00-20:00	39	58	56	62	73	77	58	66	61	74	16	9	33	66
20:00-21:00	83	90	80	77	78	76	83	73	70	90	56	25	80	100
21:00-22:00	75	72	80	74	68	77	73	74	77	70	60	44	60	53
22:00-23:00	60	72	60	67	56	62	55	69	45	68	50	57	55	49
23:00-24:00	29	41	53	35	58	67	70	43	36	54	59	58	57	58
Individual Flt.	58	65	67	63	69	49	65	57	63	68	23	59	60	65
Avg. in 15 min	81	88	78	86	99	54	97	65	99	90	12	77	93	100

Figure 4.39: Daily % prediction accuracy within 4 minutes (DTW, Nov05-Dec05)

Date	1/9	1/15	1/16	1/18	1/22	1/23	1/31	2/1	2/7	2/14	2/15	2/16	2/19	2/22
Day	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Time														
6:00-7:00	81	31	42	4	20	25	35	52	83	63	77	81	50	63
7:00-8:00	63	59	37	0	38	22	81	55	60	65	83	72	71	75
8:00-9:00	84	78	26	0	3	21	76	89	81	83	82	79	78	73
9:00-10:00	73	67	43	3	15	39	50	91	77	70	84	81	59	73
10:00-11:00	71	82	39	6	16	51	55	82	84	76	77	61	71	80
11:00-12:00	80	82	81	11	80	88	71	64	67	88	82	83	79	74
12:00-13:00	74	70	62	13	57	76	56	71	66	68	66	70	74	82
13:00-14:00	76	81	66	16	71	76	79	75	74	90	84	84	83	80
14:00-15:00	70	84	78	35	74	83	77	78	76	83	79	81	70	48
15:00-16:00	70	69	74	17	58	77	73	78	72	84	57	76	77	23
16:00-17:00	57	67	62	23	79	62	80	63	60	65	84	67	80	86
17:00-18:00	77	72	81	37	77	51	79	87	78	84	81	60	65	71
18:00-19:00	69	56	69	54	79	33	25	62	71	57	78	60	61	71
19:00-20:00	81	62	60	66	78	78	75	85	73	27	76	80	72	76
20:00-21:00	93	55	77	85	82	93	89	89	70	40	70	43	71	80
21:00-22:00	78	62	74	74	78	66	74	80	80	86	76	31	80	81
22:00-23:00	70	70	64	61	74	66	77	77	66	72	64	46	75	64
23:00-24:00	65	62	73	55	59	54	83	63	69	67	62	75	68	63
Individual Flt.	75	69	62	36	62	61	70	78	74	72	75	67	73	69
Avg. in 15 min	99	93	84	25	83	81	91	97	100	96	99	96	93	90

Figure 4.40: Daily % prediction accuracy within 4 minutes (DTW, Jan06-Feb06)

Date	9/2	9/4	9/8	9/16	9/18	9/23	9/28	10/4	10/8	10/10	10/17	10/19	10/20	10/22
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time														
6:00-7:00	76	67	50	33	67	74	73	61	71	60	64	57	70	65
7:00-8:00	82	71	74	52	68	85	79	52	78	67	63	57	73	76
8:00-9:00	69	73	58	66	45	64	78	57	84	61	39	71	66	59
9:00-10:00	72	80	62	73	26	56	85	29	80	51	8	67	65	41
10:00-11:00	72	76	63	54	30	68	72	49	76	53	47	61	58	63
11:00-12:00	61	72	56	69	59	56	72	69	82	67	60	53	56	53
12:00-13:00	61	57	77	76	55	49	79	42	71	66	64	60	55	50
13:00-14:00	28	67	76	38	56	64	55	20	31	56	64	54	55	61
14:00-15:00	10	71	36	21	29	57	13	62	62	36	57	60	61	58
15:00-16:00	52	58	68	55	76	53	48	65	52	46	58	55	57	60
16:00-17:00	64	68	54	71	38	25	60	21	67	47	40	56	69	44
17:00-18:00	41	77	63	78	69	56	58	53	66	33	48	53	66	41
18:00-19:00	54	63	56	50	40	33	53	44	53	53	59	47	53	65
19:00-20:00	73	64	65	69	45	48	31	59	58	46	37	64	65	34
20:00-21:00	30	60	81	68	60	59	63	64	63	64	47	71	64	59
21:00-22:00	70	74	74	75	77	53	66	41	65	69	74	64	58	67
22:00-23:00	38	75	79	71	84	72	58	62	69	71	88	76	74	72
23:00-24:00	--	--	0	0	100	100	100	100	100	100	100	100	60	80
Individual Fit.	58	69	65	62	55	57	61	50	65	54	53	61	62	56
Avg. in 15 min	91	94	91	87	70	88	88	67	91	84	78	94	99	62

Figure 4.41: Daily % prediction accuracy within 4 minutes (DTW, Sep06-Oct06)

Date	11/5	11/8	11/10	11/17	11/18	11/24	11/28	12/4	12/5	12/12	12/15	12/18	12/29	12/30
Day	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Time														
6:00-7:00	43	78	44	64	79	20	42	19	15	50	75	50	74	53
7:00-8:00	78	65	64	81	56	8	74	22	10	73	77	73	71	48
8:00-9:00	67	44	77	50	72	27	76	59	50	63	64	76	69	65
9:00-10:00	59	47	61	61	85	18	78	52	71	67	84	69	78	56
10:00-11:00	58	67	56	68	77	41	72	61	62	56	47	46	60	55
11:00-12:00	56	71	72	44	79	67	41	59	67	64	69	53	71	71
12:00-13:00	50	71	70	31	74	54	53	65	73	50	44	37	63	73
13:00-14:00	57	60	38	46	64	63	55	52	59	42	56	55	47	60
14:00-15:00	50	72	54	63	75	71	80	44	72	58	78	74	42	58
15:00-16:00	63	43	64	53	75	69	52	2	78	53	75	69	72	50
16:00-17:00	75	59	58	73	42	53	40	12	32	44	47	63	53	50
17:00-18:00	76	63	73	59	76	70	34	12	66	51	60	52	72	64
18:00-19:00	56	35	56	44	58	63	64	14	50	53	44	44	58	67
19:00-20:00	65	46	41	50	45	53	60	16	54	68	57	64	41	65
20:00-21:00	35	67	72	38	71	44	55	14	71	64	50	80	60	60
21:00-22:00	50	54	66	64	53	46	62	17	54	55	58	70	66	32
22:00-23:00	70	69	67	80	85	78	69	52	71	62	69	65	63	62
23:00-24:00	80	71	63	100	67	100	80	83	67	60	83	86	83	83
Individual Flt.	60	58	60	56	66	53	60	35	59	57	62	61	62	55
Avg. in 15 min	90	86	87	90	96	71	88	48	83	94	99	91	99	83

Figure 4.42: Daily % prediction accuracy within 4 minutes (DTW, Nov06-Dec06)

Date	1/9	1/15	1/16	1/18	1/22	1/23	1/31	2/1	2/7	2/14	2/15	2/16	2/19	2/22
Day	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Time														
6:00-7:00	32	0	0	57	27	60	21	55	9	0	20	23	50	60
7:00-8:00	17	29	0	73	24	73	20	52	22	0	57	46	69	50
8:00-9:00	46	41	13	71	14	45	44	19	21	14	21	58	63	64
9:00-10:00	52	38	28	55	32	32	61	3	29	14	49	56	67	51
10:00-11:00	68	41	26	50	17	53	49	17	39	14	39	47	52	61
11:00-12:00	69	67	83	53	28	50	59	35	41	33	67	50	53	67
12:00-13:00	42	48	45	61	29	43	70	29	55	53	48	31	53	52
13:00-14:00	59	28	67	54	39	22	70	36	47	27	59	63	58	24
14:00-15:00	61	24	71	52	54	23	70	37	66	48	59	56	64	4
15:00-16:00	78	11	55	52	40	33	62	30	30	48	47	54	70	5
16:00-17:00	50	0	55	68	48	17	72	61	31	44	56	61	57	33
17:00-18:00	63	7	52	46	34	23	53	42	43	31	53	46	71	31
18:00-19:00	38	31	67	42	50	9	64	33	46	69	33	41	43	68
19:00-20:00	37	15	50	14	44	18	62	25	62	59	12	51	57	51
20:00-21:00	82	0	30	44	40	25	55	45	50	83	27	50	57	67
21:00-22:00	34	9	33	2	45	11	54	25	59	44	45	49	56	33
22:00-23:00	79	38	50	29	57	43	43	38	60	50	39	82	67	47
23:00-24:00	67	70	86	44	75	57	67	60	86	55	100	50	75	100
Individual Flt.	54	25	43	45	36	33	56	32	44	37	42	50	60	42
Avg. in 15 min	84	32	71	70	51	46	90	52	78	55	70	87	93	71

Figure 4.43: Daily % prediction accuracy within 4 minutes (DTW, Jan07-Feb07)

Date	9/2	9/4	9/8	9/16	9/18	9/23	9/28	10/4	10/8	10/10	10/17	10/19	10/20	10/22
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time														
6:00-7:00	57	43	68	74	61	67	63	59	54	73	57	72	70	54
7:00-8:00	83	63	64	71	67	87	67	82	70	48	82	68	88	76
8:00-9:00	68	45	71	71	68	61	61	56	71	65	51	49	55	66
9:00-10:00	81	53	74	62	65	55	64	67	78	76	67	64	61	67
10:00-11:00	67	66	72	58	55	55	56	46	51	53	50	36	50	57
11:00-12:00	67	68	67	47	82	67	59	50	53	69	47	38	60	71
12:00-13:00	51	53	50	60	67	48	56	51	55	49	37	38	59	42
13:00-14:00	71	62	65	62	46	52	57	57	44	48	41	43	33	49
14:00-15:00	50	62	64	71	48	32	57	54	43	54	56	28	19	35
15:00-16:00	59	59	55	48	62	62	48	61	55	53	44	56	44	51
16:00-17:00	80	74	83	76	44	76	35	68	58	53	40	50	64	31
17:00-18:00	63	70	63	67	57	61	49	55	66	63	21	40	56	41
18:00-19:00	43	44	80	0	60	33	38	57	63	86	86	50	67	50
19:00-20:00	68	58	52	37	66	46	51	69	54	66	37	43	46	33
20:00-21:00	40	80	64	55	90	100	78	70	60	82	55	64	69	50
21:00-22:00	58	60	64	59	62	61	71	58	55	74	69	49	49	57
22:00-23:00	30	88	67	84	70	63	64	76	70	61	82	52	54	56
23:00-24:00	--	100	100	0	100	33	0	100	75	0	0	75	50	40
Individual Flt.	64	60	63	59	61	58	57	60	58	61	49	48	51	51
Avg. in 15 min	88	94	99	91	91	88	84	93	88	94	84	67	68	83

Figure 4.44: Daily % prediction accuracy within 4 minutes (DTW, Sep07-Oct07)

Date	11/5	11/8	11/10	11/17	11/18	11/24	11/28	12/4	12/5	12/12	12/15	12/18	12/29	12/30
Day	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Time														
6:00-7:00	72	65	67	58	67	52	53	55	4	39	63	27	72	52
7:00-8:00	74	71	94	85	64	62	31	88	0	38	42	33	65	67
8:00-9:00	58	72	81	67	78	39	57	59	3	41	50	35	39	60
9:00-10:00	67	71	68	54	60	76	64	55	5	57	52	53	30	62
10:00-11:00	52	51	42	32	65	65	62	35	10	35	7	42	37	44
11:00-12:00	68	50	56	59	76	61	59	50	39	67	13	67	41	60
12:00-13:00	26	51	45	15	70	74	70	46	38	43	23	51	38	63
13:00-14:00	42	42	33	36	31	56	55	50	24	48	20	46	63	58
14:00-15:00	35	33	41	42	27	62	57	60	15	52	27	50	54	60
15:00-16:00	29	49	45	60	54	68	42	54	28	61	31	47	42	54
16:00-17:00	39	50	73	43	61	63	63	71	53	61	58	44	55	73
17:00-18:00	41	43	69	44	63	63	61	47	34	49	16	46	64	39
18:00-19:00	45	82	0	83	33	43	64	58	60	25	25	42	43	30
19:00-20:00	50	28	40	40	55	72	50	38	37	54	31	53	45	10
20:00-21:00	88	89	29	43	71	100	50	56	71	75	50	50	100	0
21:00-22:00	64	45	57	44	56	51	65	46	25	63	44	52	53	3
22:00-23:00	57	53	54	63	53	43	60	34	31	33	42	48	66	21
23:00-24:00	50	50	67	50	50	50	33	50	60	67	33	33	33	0
Individual Fit.	50	50	52	46	57	60	57	49	26	50	32	47	50	41
Avg. in 15 min	80	68	84	61	84	86	84	86	45	87	54	75	84	77

Figure 4.45: Daily % prediction accuracy within 4 minutes (DTW, Nov07-Dec07)

Date	1/9	1/15	1/16	1/18	1/22	1/23	1/31	2/1	2/7	2/14	2/15	2/16	2/19	2/22
Day	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Time														
6:00-7:00	61	4	32	47	0	52	38	0	19	20	50	60	58	0
7:00-8:00	74	10	61	63	6	50	35	0	31	50	33	55	75	5
8:00-9:00	68	15	44	69	0	39	75	6	18	39	38	59	49	10
9:00-10:00	71	17	48	67	0	45	77	6	23	52	33	65	39	19
10:00-11:00	78	40	62	64	7	66	61	0	25	48	41	69	38	34
11:00-12:00	47	47	35	74	6	68	58	11	20	47	53	73	59	53
12:00-13:00	71	58	64	77	2	56	81	9	31	76	53	59	70	52
13:00-14:00	56	45	61	54	0	16	75	4	22	65	65	65	57	52
14:00-15:00	57	84	70	70	25	5	52	11	43	53	52	53	58	39
15:00-16:00	67	47	69	67	33	22	43	9	18	65	67	57	54	66
16:00-17:00	69	69	67	62	42	8	50	0	57	62	43	50	63	57
17:00-18:00	59	72	34	54	47	3	34	6	34	52	62	71	56	46
18:00-19:00	43	88	67	63	50	0	57	33	50	63	88	57	100	63
19:00-20:00	61	57	64	53	58	9	51	35	20	59	54	70	45	58
20:00-21:00	80	43	100	40	100	0	90	0	67	60	33	80	67	0
21:00-22:00	72	28	74	56	56	1	52	55	26	68	69	61	34	45
22:00-23:00	49	58	48	61	36	30	38	40	39	76	62	57	39	42
23:00-24:00	75	60	75	25	40	0	29	44	50	50	33	67	0	100
Individual Flt.	64	44	58	61	28	27	55	18	27	57	54	63	52	42
Avg. in 15 min	97	62	84	93	33	38	90	4	26	77	84	94	87	70

Figure 4.46: Daily % prediction accuracy within 4 minutes (DTW, Jan08-Feb08)

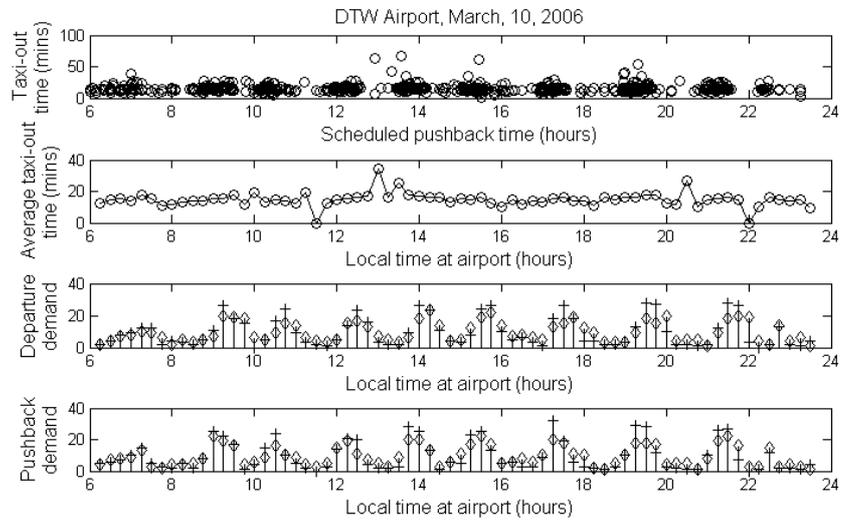


Figure 4.47: Airport trends across time of day

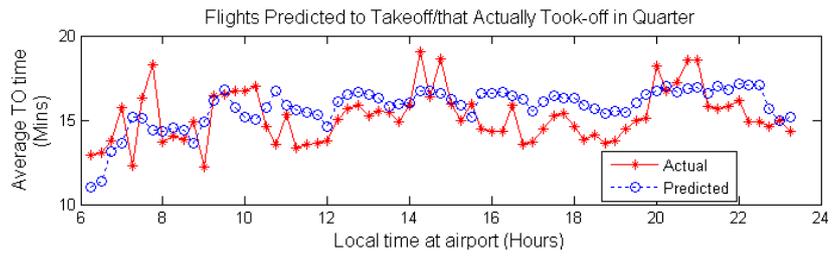


Figure 4.48: Taxi-out time trends (average in 15 min intervals)

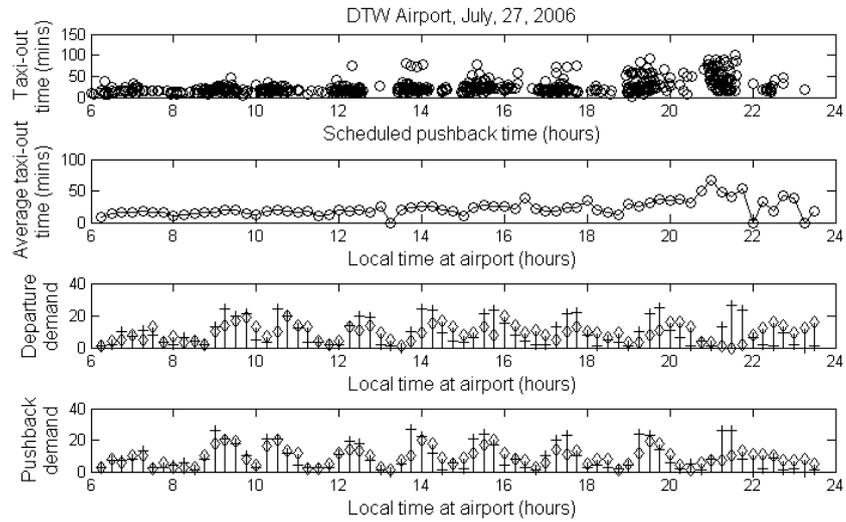


Figure 4.49: Airport trends across time of day

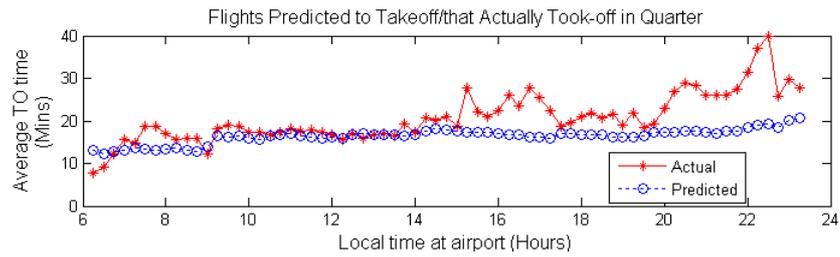


Figure 4.50: Taxi-out time trends (average in 15 min intervals)

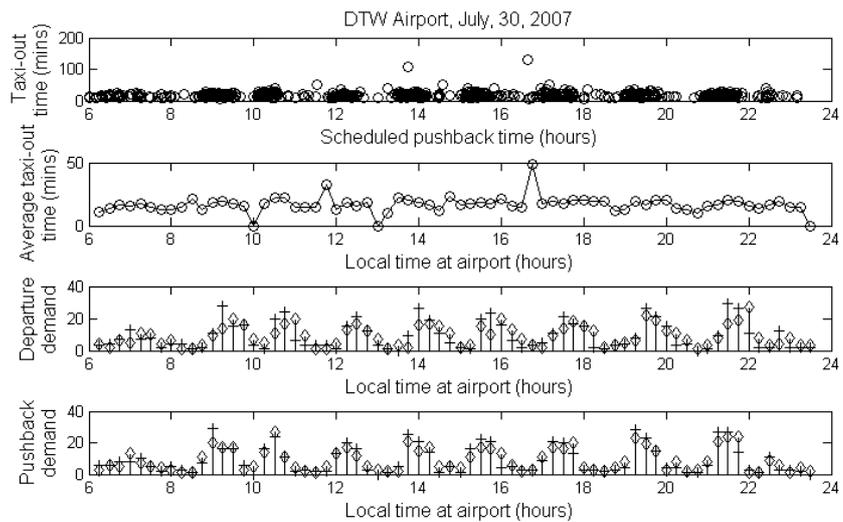


Figure 4.51: Airport trends across time of day

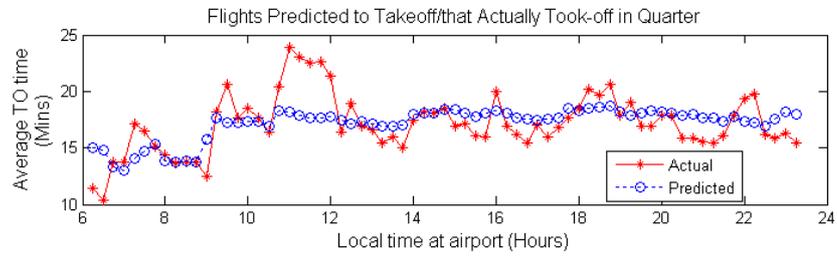


Figure 4.52: Taxi-out time trends (average in 15 min intervals)

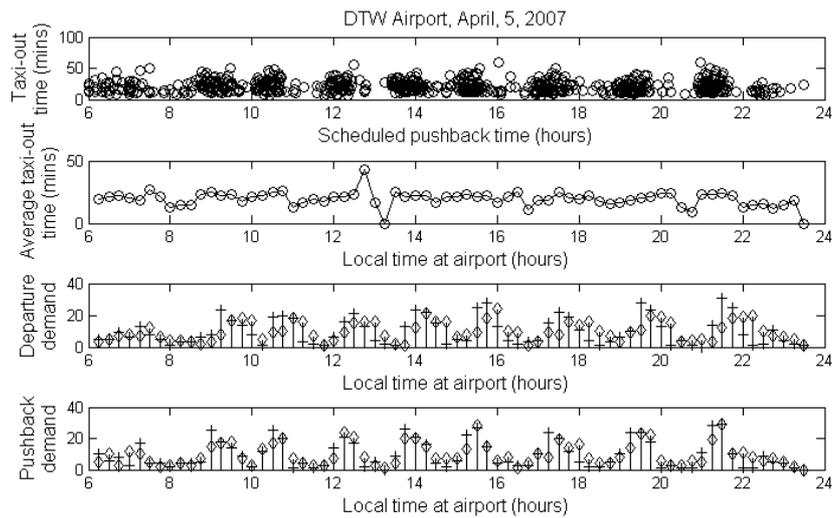


Figure 4.53: Airport trends across time of day

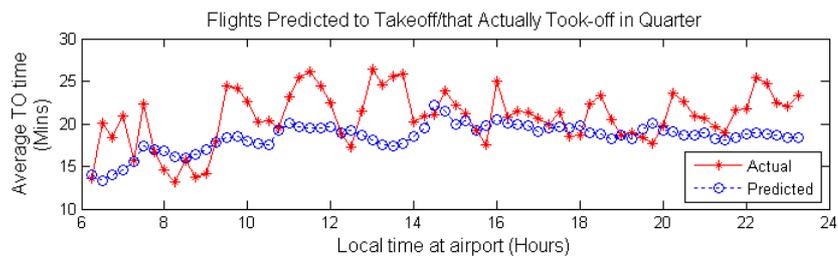


Figure 4.54: Taxi-out time trends (average in 15 min intervals)

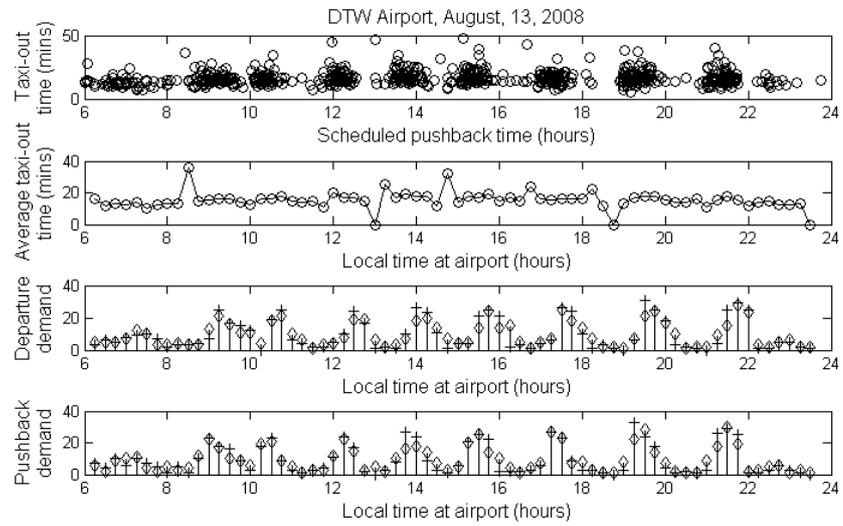


Figure 4.55: Airport trends across time of day

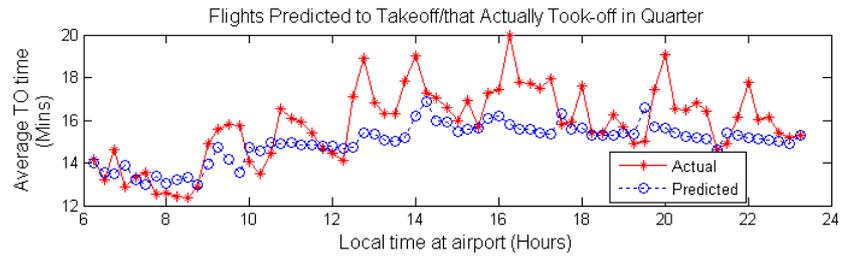


Figure 4.56: Taxi-out time trends (average in 15 min intervals)

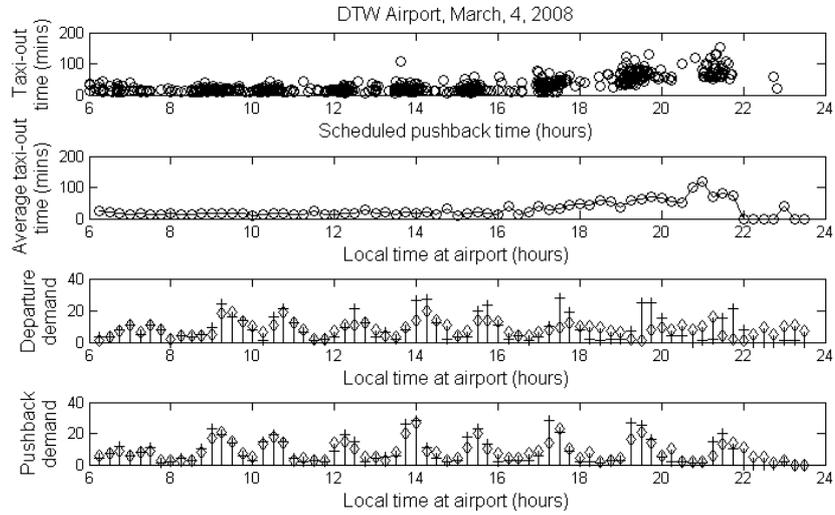


Figure 4.57: Airport trends across time of day

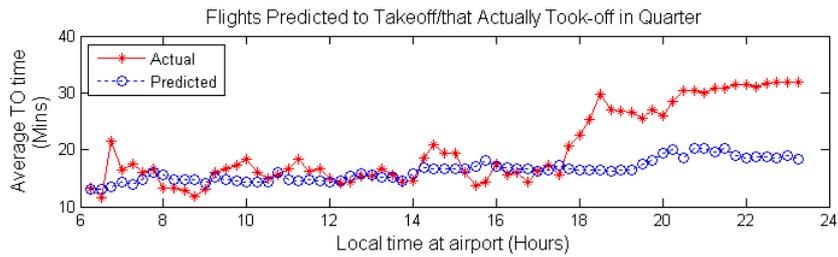


Figure 4.58: Taxi-out time trends (average in 15 min intervals)

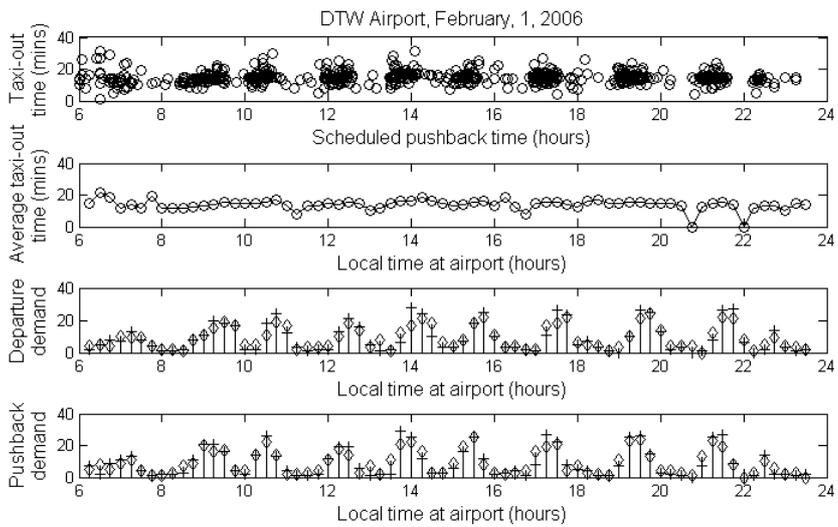


Figure 4.59: Airport trends across time of day

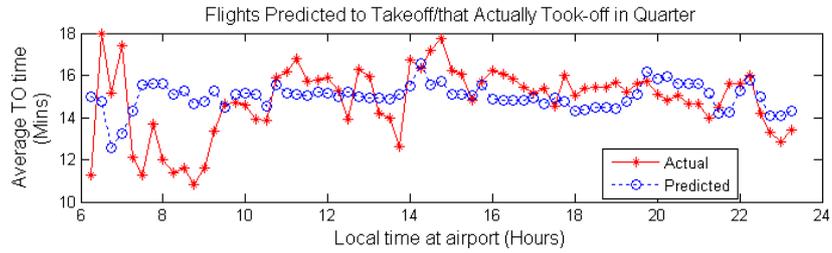


Figure 4.60: Taxi-out time trends (average in 15 min intervals)

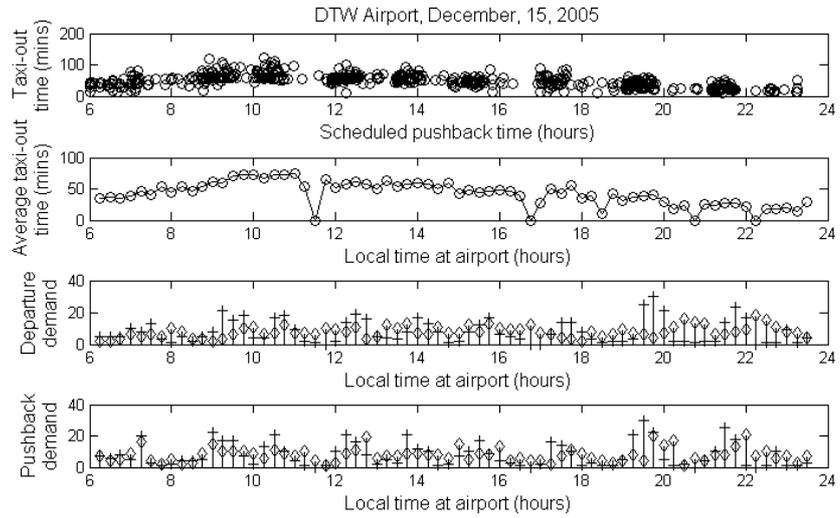


Figure 4.61: Airport trends across time of day

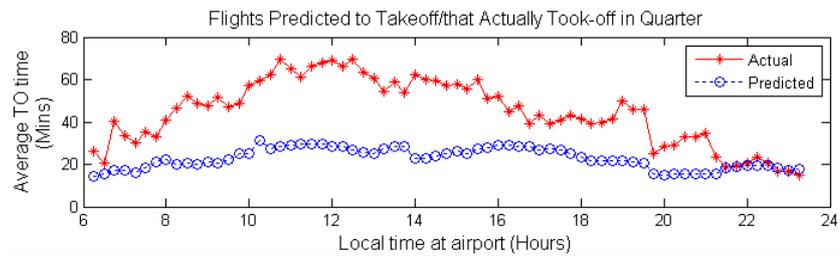


Figure 4.62: Taxi-out time trends (average in 15 min intervals)

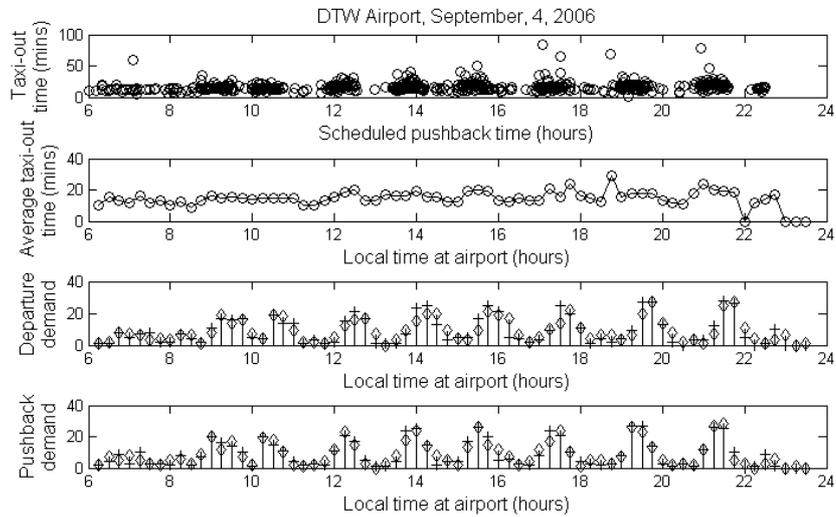


Figure 4.63: Airport trends across time of day

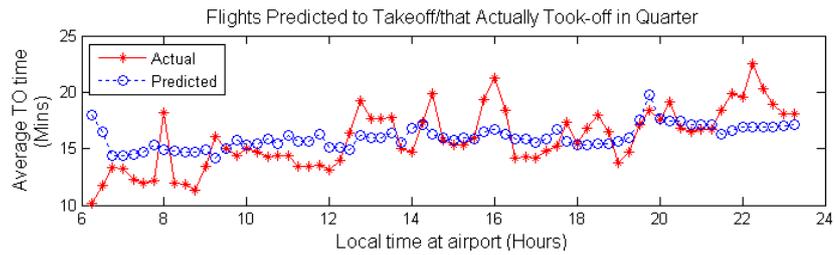


Figure 4.64: Taxi-out time trends (average in 15 min intervals)

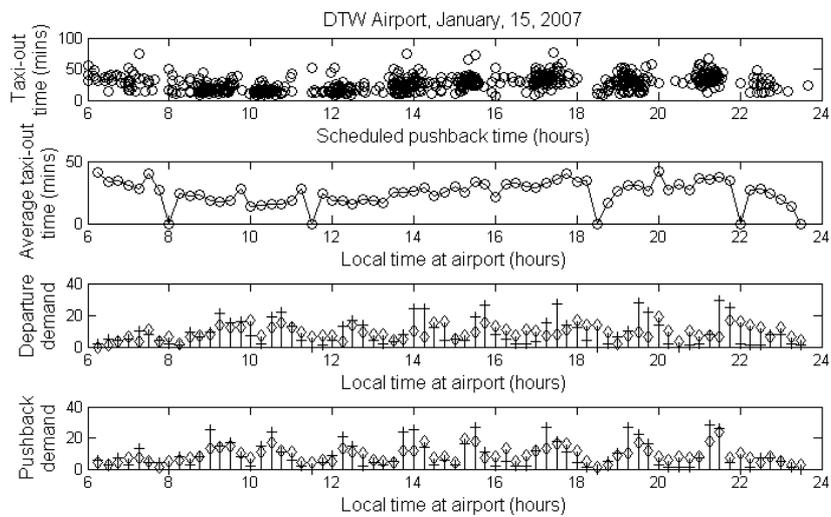


Figure 4.65: Airport trends across time of day

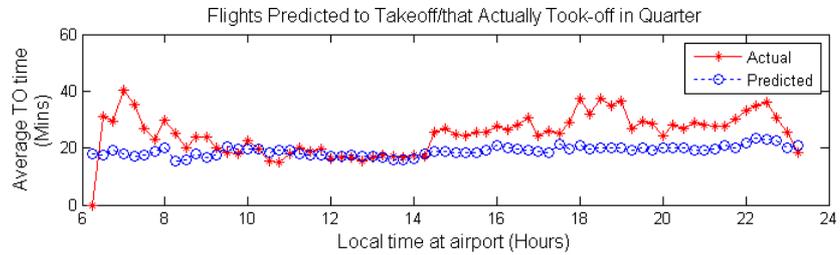


Figure 4.66: Taxi-out time trends (average in 15 min intervals)

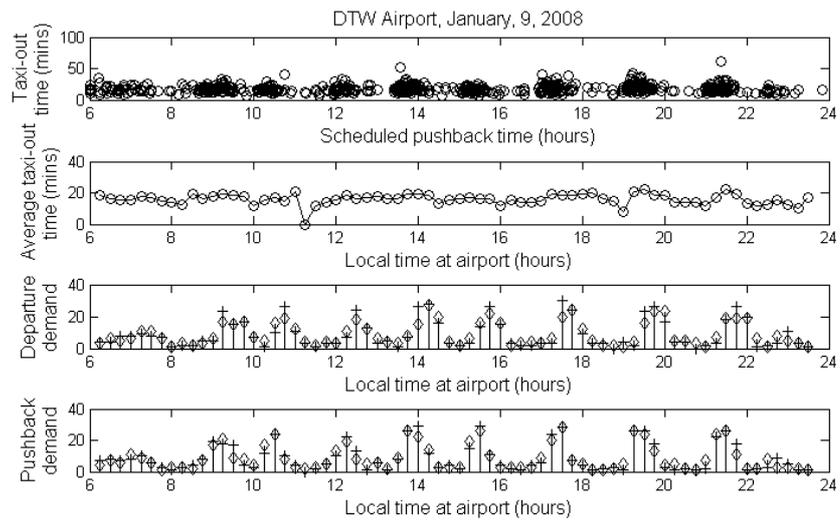


Figure 4.67: Airport trends across time of day

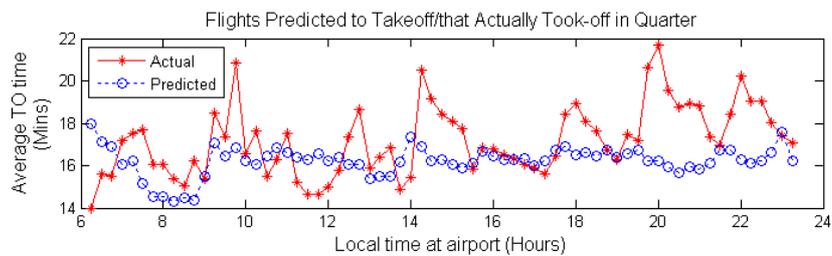


Figure 4.68: Taxi-out time trends (average in 15 min intervals)

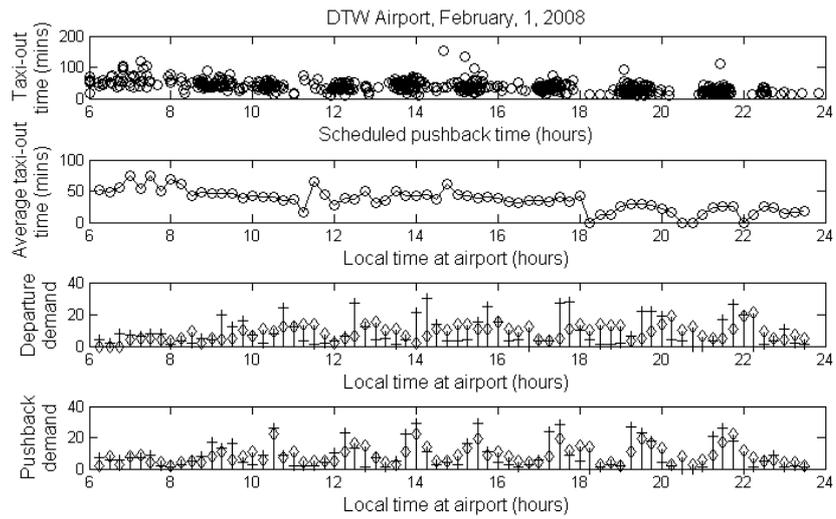


Figure 4.69: Airport trends across time of day

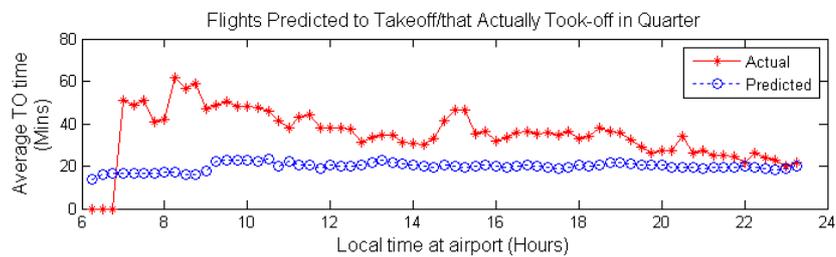


Figure 4.70: Taxi-out time trends (average in 15 min intervals)

Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
MSE	5	5	6	5	5	5	6	7	4	9	6	6	5	6
Mean (Actual)	16	16	16	16	15	16	16	16	15	18	17	15	16	17
Std. Dev. (Act)	5	5	6	5	5	5	5	6	4	8	6	6	5	6
Mean (Pred)	16	15	16	16	15	16	15	15	15	16	15	15	15	15
Std. Dev. (Pred)	2	2	2	2	2	2	2	2	2	2	2	1	2	2
Date	5/2	5/21	5/26	5/27	5/29	5/30	5/31	6/2	6/6	6/13	6/21	6/22	6/23	6/24
MSE	6	6	8	6	7	8	11	11	10	10	11	11	9	6
Mean (Actual)	16	16	20	18	17	20	20	19	19	19	20	20	18	17
Std. Dev. (Act)	6	6	7	6	7	7	10	10	10	9	10	10	8	6
Mean (Pred)	16	15	16	16	16	16	16	16	16	16	16	16	16	16
Std. Dev. (Pred)	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
MSE	6	7	5	8	5	17	12	7	6	5	6	6	7	6
Mean (Actual)	17	16	16	19	16	23	20	18	17	16	17	17	18	17
Std. Dev. (Act)	6	7	5	7	5	15	11	7	6	5	6	6	6	6
Mean (Pred)	16	15	16	16	16	17	16	16	16	16	15	15	16	16
Std. Dev. (Pred)	2	2	2	2	2	3	2	2	2	2	2	2	2	2

Figure 4.71: Daily % actual and prediction taxi-out time statistics (DTW, Mar06-Aug06)

Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
MSE	7	7	6	7	8	7	7	10	12	8	8	7	7	10
Mean (Actual)	18	18	18	18	19	18	17	21	22	18	18	16	17	20
Std. Dev. (Act)	6	7	6	6	7	7	7	9	11	7	8	6	7	9
Mean (Pred)	17	17	17	17	17	17	17	18	18	17	17	17	17	18
Std. Dev. (Pred)	2	2	2	2	2	2	2	3	3	2	3	2	2	4
Date	5/2	5/21	5/26	5/27	5/29	5/30	5/31	6/2	6/6	6/13	6/21	6/22	6/23	6/24
MSE	7	6	7	8	7	9	7	10	8	7	9	7	6	6
Mean (Actual)	19	18	18	19	18	19	19	19	19	18	19	18	17	17
Std. Dev. (Act)	7	6	7	7	6	8	7	8	7	7	8	6	6	5
Mean (Pred)	17	17	17	17	17	17	17	19	17	17	18	17	17	17
Std. Dev. (Pred)	2	2	2	3	2	3	2	4	2	2	3	2	3	2
Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
MSE	6	7	8	8	7	9	6	7	12	7	6	7	8	8
Mean (Actual)	17	18	19	18	18	20	17	19	22	18	18	18	20	18
Std. Dev. (Act)	6	6	7	7	6	8	6	7	9	7	6	6	7	6
Mean (Pred)	17	17	18	18	17	18	17	17	19	17	17	17	18	17
Std. Dev. (Pred)	2	3	2	3	2	3	2	2	3	2	2	2	3	3

Figure 4.72: Daily % actual and prediction taxi-out time statistics (DTW, Mar07-Aug07)

Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
MSE	15	13	7	10	15	6	8	8	7	7	6	6	7	7
Mean (Actual)	23	24	18	21	24	17	18	18	18	18	16	17	19	18
Std. Dev. (Act)	15	11	6	8	13	5	6	6	6	6	6	6	7	6
Mean (Pred)	17	19	17	17	18	15	17	17	16	16	15	15	16	15
Std. Dev. (Pred)	4	4	2	3	4	2	3	3	2	2	2	2	2	2
Date	5/2	5/21	5/26	5/27	5/29	5/30	5/31	6/2	6/6	6/13	6/21	6/22	6/23	6/24
MSE	10	7	6	6	6	8	8	7	10	12	8	7	10	7
Mean (Actual)	21	18	17	16	18	19	18	19	20	22	18	18	21	18
Std. Dev. (Act)	8	6	5	6	5	8	8	6	9	10	7	7	9	6
Mean (Pred)	17	15	16	15	15	16	17	16	17	19	17	17	18	16
Std. Dev. (Pred)	3	2	2	2	2	3	3	2	3	4	2	3	3	2
Date	7/2	7/4	7/10	7/11	7/23	7/27	7/30	8/2	8/7	8/13	8/15	8/16	8/25	8/26
MSE	9	8	8	7	7	7	6	6	8	5	5	5	6	6
Mean (Actual)	19	17	19	18	17	17	17	16	19	16	16	16	16	16
Std. Dev. (Act)	8	7	8	6	6	6	6	5	7	5	5	5	6	6
Mean (Pred)	16	16	16	16	16	16	15	16	16	15	15	16	15	15
Std. Dev. (Pred)	3	3	2	2	2	2	2	3	3	2	2	3	2	2

Figure 4.73: Daily % actual and prediction taxi-out time statistics (DTW, Mar08-Aug08)

Date	9/2	9/4	9/8	9/16	9/18	9/23	9/28	10/4	10/8	10/10	10/17	10/19	10/20	10/22
MSE	6	5	7	8	5	8	6	5	5	7	5	6	6	6
Mean (Actual)	17	16	17	15	16	16	16	16	15	16	17	15	17	16
Std. Dev. (Act)	5	5	6	7	4	7	6	5	5	7	5	6	5	6
Mean (Pred)	16	15	16	16	16	16	16	16	15	16	16	16	16	15
Std. Dev. (Pred)	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Date	11/5	11/8	11/10	11/17	11/18	11/24	11/28	12/4	12/5	12/12	12/15	12/18	12/29	12/30
MSE	6	6	7	8	7	12	7	13	6	7	25	9	8	7
Mean (Actual)	17	17	18	18	17	22	18	22	17	17	38	19	16	16
Std. Dev. (Act)	6	6	7	7	7	10	6	13	6	6	20	9	6	5
Mean (Pred)	16	16	16	17	17	18	17	18	17	16	21	16	17	17
Std. Dev. (Pred)	2	2	2	2	2	3	2	4	2	2	6	2	2	2
Date	1/9	1/15	1/16	1/18	1/22	1/23	1/31	2/1	2/7	2/14	2/15	2/16	2/19	2/22
MSE	6	6	7	19	7	7	7	4	5	5	5	7	6	6
Mean (Actual)	16	17	15	29	17	17	16	15	16	15	15	17	16	16
Std. Dev. (Act)	5	5	6	16	7	6	6	4	5	5	5	6	5	5
Mean (Pred)	16	15	16	19	16	16	16	15	15	15	15	16	16	16
Std. Dev. (Pred)	2	2	2	4	2	2	2	2	2	2	2	2	2	2

Figure 4.74: Daily % actual and prediction taxi-out time statistics (DTW, Sep05-Feb06)

Date	9/2	9/4	9/8	9/16	9/18	9/23	9/28	10/4	10/8	10/10	10/17	10/19	10/20	10/22
MSE	8	8	6	6	7	8	7	9	7	7	8	7	7	7
Mean (Actual)	17	17	17	17	20	19	18	20	18	19	20	18	18	20
Std. Dev. (Act)	8	7	6	5	7	7	7	8	7	7	8	6	7	7
Mean (Pred)	16	16	16	16	18	17	17	18	17	16	17	17	17	17
Std. Dev. (Pred)	2	2	2	2	2	3	2	3	2	2	2	2	2	2
Date	11/5	11/8	11/10	11/17	11/18	11/24	11/28	12/4	12/5	12/12	12/15	12/18	12/29	12/30
MSE	7	7	7	7	6	10	7	18	10	9	7	7	6	7
Mean (Actual)	18	19	18	19	17	19	18	28	20	18	18	18	17	17
Std. Dev. (Act)	6	7	7	6	5	9	6	15	9	9	7	7	5	6
Mean (Pred)	16	17	17	17	17	18	17	20	17	17	17	16	17	18
Std. Dev. (Pred)	2	2	2	2	3	3	2	4	3	3	2	2	2	3
Date	1/9	1/15	1/16	1/18	1/22	1/23	1/31	2/1	2/7	2/14	2/15	2/16	2/19	2/22
MSE	9	14	11	13	11	15	9	12	10	16	11	9	7	16
Mean (Actual)	19	26	22	22	23	25	19	24	20	26	22	20	18	24
Std. Dev. (Act)	8	12	10	12	10	13	8	10	9	14	11	8	7	15
Mean (Pred)	17	19	18	17	19	18	17	18	18	19	19	18	17	19
Std. Dev. (Pred)	2	3	3	3	3	3	2	3	3	3	3	2	2	3

Figure 4.75: Daily % actual and prediction taxi-out time statistics (DTW, Sep06-Feb07)

Date	9/2	9/4	9/8	9/16	9/18	9/23	9/28	10/4	10/8	10/10	10/17	10/19	10/20	10/22
MSE	6	6	6	6	7	7	6	7	9	7	9	10	8	7
Mean (Actual)	17	18	17	18	18	18	18	18	19	18	20	21	19	20
Std. Dev. (Act)	5	6	6	6	7	7	6	6	8	6	8	9	7	7
Mean (Pred)	16	16	17	17	17	17	17	17	17	17	18	18	17	17
Std. Dev. (Pred)	2	2	2	2	2	2	2	2	2	2	2	3	2	2
Date	11/5	11/8	11/10	11/17	11/18	11/24	11/28	12/4	12/5	12/12	12/15	12/18	12/29	12/30
MSE	8	8	8	9	8	8	7	8	18	8	16	10	9	17
Mean (Actual)	20	20	19	20	18	18	19	18	27	19	27	23	20	25
Std. Dev. (Act)	8	7	7	8	8	7	7	7	16	8	14	9	9	16
Mean (Pred)	18	17	18	18	17	17	17	18	20	18	19	20	19	19
Std. Dev. (Pred)	3	3	3	2	2	2	2	3	4	3	4	4	3	3
Date	1/9	1/15	1/16	1/18	1/22	1/23	1/31	2/1	2/7	2/14	2/15	2/16	2/19	2/22
MSE	6	12	7	8	20	19	8	21	16	8	9	7	9	12
Mean (Actual)	18	22	19	18	31	31	18	35	29	20	20	18	20	23
Std. Dev. (Act)	6	11	7	7	16	15	7	15	13	8	8	6	8	11
Mean (Pred)	16	18	17	17	20	19	17	20	20	18	18	17	18	19
Std. Dev. (Pred)	2	3	2	2	3	3	2	3	3	2	2	2	2	3

Figure 4.76: Daily % actual and prediction taxi-out time statistics (DTW, Sep07-Feb08)

Chapter 5: CONCLUSIONS

The scalability of approximate dynamic programming approaches was addressed in this dissertation by investigating a new method based on diffusion wavelet theory for value function approximation. The method was tested on the highly stochastic problem of taxi-out time estimation at airports to establish a proof of concept for the research objective.

5.1 Methodology

The viability of ADP for learning policies in large-scale applications is challenged by the size and dimension of the value function vector. This problem is compounded for continuous state space models. The challenge is to determine the best basis functions that can approximate the value function in real time before the shape of the value function is known. Diffusion wavelets represent the state space as a manifold and effectively exploit the structure of the problem to determine the number and shape of the features that best represent the value function. The characteristics of the basis functions related to orthogonality and minimum entropy allows for accurate and efficient reconstruction of the function. This structure also avoids the need for a separate interpolation scheme for states not visited during the policy learning phase.

Taxi-out time prediction accuracy results based on the diffusion wavelet value function approximation ADP method developed in this dissertation accurately matched the results obtained using the baseline Q-learning approach for several data sets. The method was also compared with a value function approximation method based on regression. For the taxi-out time estimation model, the state space structure is clearly defined and linear. The diffusion wavelet method was able to provide the same level of accuracy as the regression approach to value function approximation. The heuristic choice of value function approximation method

is application specific. It is not possible to know in advance as to which class of problems will benefit from a generalized choice of basis functions for value function approximation [Powell, 2007]. The capability of the diffusion wavelet approach to effectively handle and compress multi-dimensional data makes it an appealing choice for large scale ADP models.

The value function approximation method reduces storage requirements by sampling the state space. Experimental results related to computational time suggest that there may be a trade-off point based on input size (state space combinations) beyond which the computational performance of the diffusion wavelet value function approximation scheme improves over the explicit storage approach. Further investigation on the effect of sample size on algorithm accuracy and computational time may be undertaken as future work.

Proofs of convergence of stochastic approximation methods are not straightforward to obtain. In addition, while this may be of theoretical importance, they do not easily translate into well-defined or reliable stopping criteria. For the tested problem of taxi-out time estimation, the diffusion wavelet based value function approximation approach provided a similar range of prediction accuracy with respect to the Q-learning approach using the same number of iterations. Further studies on rates of convergence of the diffusion wavelet based value function approximation approach would prove useful. The diffusion wavelet based value function approximation method may also be tested on problems of larger dimensions which may require gridding of the state space, and separate value function approximators for each grid.

5.2 Taxi-Out Time Estimation

The first step towards effective departure schedule planning and reduction in departure delays after gate pushback is an accurate taxi-out time prediction capability. The stochastic dynamic programming model developed in this dissertation has the capability to predict taxi-out times sequentially in real time, allowing for updations of the predictions as time evolves.

Prediction of taxi-out times may assist in near-time departure planning, where the

objective is to minimize downstream congestion by getting flights into the air as early as possible. It is expected that control tower operations, surface management systems, and airline scheduling can benefit from this prediction by adjusting schedules to minimize congestion, delays, and emissions, and also by better utilization of ground personnel and resources. Taxi-out time estimates made available for flights affected by a GDP will provide a means to determine if the flight can meet its Expected Departure Clearance (EDCT) time. With airport dynamics changing throughout the day in the face of uncertainties, prediction of airport taxi-out time averages combined with individual flight predictions may help airlines manage decisions such as incurring delays at the gate as opposed to increasing emissions due to longer taxi times. Air Traffic Control may also benefit from this knowledge when making decisions regarding holding flights at the gate or ramp area due to increased congestion. This could improve the performance of air traffic flow management both on ground and in air across the entire NAS in the US and worldwide.

More detailed information on OOOI times and airport dynamics on an individual flight basis may be obtained with the development and deployment of surface surveillance systems such as the ASDE-X system from Sensis Corporation. Future work in the development of surface surveillance systems includes analyzing the radar track information and overcoming challenges in extracting OOOI event times thereby increasing accuracy of the data [Signor and Levy, 2006]. Training the RL estimator with this more detailed information would potentially increase the accuracy of the taxi-out time predictions by more precisely capturing the state of the system, and through standardized and clearly defined OOOI event times.

It is hypothesized that with the availability of more detailed information on departure runway and airline terminals, or other categorical variables influencing taxi-out times, the reinforcement learning approach would provide a means to effectively model and capture nonlinear relationships in the airport departure process. Sensitivity of the results to the prediction look-ahead window (currently set at 15 min) may prove useful. Dynamic systems under uncertainty such as the operation of an airport evolve over time and have to be predicted and adjusted periodically. The ADP approach provides a sequential prediction

capability that may also be extended to aid in departure planning and decision making based on the structure of the model.

Appendix A: The Robbins-Monro Algorithm

The Robbins-Monro algorithm [Robbins and Monro, 1951] is a simple idea that helps us estimate the mean of a random variable from its samples. The following description of the algorithm is summarized from [Gosavi, 2003].

The mean of n samples of a random variable may be obtained by a straightforward averaging process. That is, if X is a random variable, and s^i is the i^{th} independent sample of X , then, by the strong law of large numbers, with probability 1,

$$E[X] = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n s^i}{n} \quad (\text{A.1})$$

where, $E[X]$ denotes the *expected value* of random variable X .

Now, let X^n denote the estimate of X in the n^{th} iteration (after n samples have been obtained). Then,

$$X^n = \frac{\sum_{i=1}^n s^i}{n} \quad (\text{A.2})$$

Now,

$$X^{n+1} = \frac{\sum_{i=1}^{n+1} s^i}{n+1} \quad (\text{A.3})$$

$$= \frac{\sum_{i=1}^n s^i + s^{n+1}}{n+1} \quad (\text{A.4})$$

$$= \frac{X^n n + s^{n+1}}{n+1} \text{ (using Equation A.2)} \quad (\text{A.5})$$

$$= \frac{X^n n + X^n - X^n + s^{n+1}}{n+1} \quad (\text{A.6})$$

$$= \frac{X^n n + X^n}{n+1} - \frac{X^n}{n+1} + \frac{s^{n+1}}{n+1} \quad (\text{A.7})$$

$$= X^n - \frac{X^n}{n+1} + \frac{s^{n+1}}{n+1} \quad (\text{A.8})$$

$$= (1 - \alpha^{n+1})X^n + \alpha^{n+1}s^{n+1} \quad (\text{A.9})$$

if $\alpha^{n+1} = 1/(n+1)$.

Equation A.9 represents the Robbins-Monro algorithm. When $\alpha^{n+1} = 1/(n+1)$, this equation is equivalent to direct averaging. Other definitions of α that provide indirect averaging may also be used. In Reinforcement Learning (RL) algorithms, α is known as the *learning rate* or *step size*, and plays a crucial role in guaranteeing convergence to the optimal solution.

Appendix B: Derivation of Unimpeded Taxi Times

The nominal or unimpeded taxi times provided in the FAA's ASPM database is based on a regression model derived by the Office of Aviation Policy and Plans which is part of the FAA. A summary of the method employed is discussed below.

The inputs to the unimpeded taxi time estimation model are based on considering city pair flights linking the different departure and destination airports. Data relating to the day, month, and year of operation of the flight, along with their corresponding OOOI (Out, Off, On, In) event times are noted. For a specific airport, both departure and arrival flights are categorized into subgroups, based on the airline, and the season of operation (December-February constitutes the Winter months, March-May constitutes Spring, June-August are the Summer months, and Fall is categorized by the months September-November).

The next step is to define departure and arrival queue lengths. The actual gate out time of a departing flight signifies its entry into a departure queue. Its exit from the departure queue is marked by its actual wheels-off event time. In a similar manner, an arriving flight enters the arrival queue when its actual wheels-on time is recorded, and it exits the arrival queue when the gate-in event occurs.

Each minute of the day is now considered a separate bin, and for a departing flight, the number of flights ahead of it in the departure queue, at the time it enters the departure queue, is taken as the departure queue length experienced by the flight. This is represented by x_o . The number of arriving flights taxiing at the time instant that the departing flight enters the departure queue is represented by x_i .

A linear regression model using the taxi-out times of departing flights (y_o) as the dependent variable is fit, so that, $y_o = ax_o + bx_i + c$, where $a \geq 0$ and $b \geq 0$ since, as the number of aircraft on the ground increase, the taxi-out time is expected to increase.

Since this computation aims at estimating the unimpeded taxi-out times under no congestion conditions, only flights with taxi-out times within the 75th percentile are considered for the regression model. A similar model is also established for estimating the unimpeded

taxi-in times for each subgroup based on airline and season.

Finally, in order to obtain the nominal taxi-out time, the value of x_o is set to 1, and x_i is set to 0, in the appropriate model.

Source: Kondo, A. The Derivation of the Unimpeded Taxi-out and Taxi-in Times in the Estimation of the Taxi Time Delays. Technical Note 2009 No. 1, Office of Aviation Policy, Federal Aviation Administration. (Work in Progress)

Bibliography

Bibliography

- [Ahmed and Anjum, 1997] Ahmed, M. S. and Anjum, M. (1997). Neural-net based self-tuning control of nonlinear plants. *International Journal of Control*, 66:85–104.
- [Anagnostakis et al., 2000] Anagnostakis, I., Idris, H., Clarke, J., Feron, E., Hansman, J., Odoni, A., and Hall, W. (2000). A conceptual design of a departure planner decision aid. 3rd USA/Europe Air Traffic Management R and D Seminar, Naples, Italy.
- [Atkins and Walton, 2002] Atkins, S. and Walton, D. (2002). Prediction and control of departure runway balancing at dallas fort worth airport. In *Proceedings of the American Control Conference*, Anchorage, AK. American Control Conference.
- [Balakrishna et al., 2008a] Balakrishna, P., Ganesan, R., and Sherry, L. (2008a). Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of tampa bay departures. In *Proceedings of the 3rd International Conference on Research in Air Transportation*, Washington DC. International Conference on Research in Air Transportation.
- [Balakrishna et al., 2008b] Balakrishna, P., Ganesan, R., and Sherry, L. (2008b). Airport taxi-out prediction using approximate dynamic programming: An intelligence-based paradigm. *Transportation Research Record*, 2052:54–61.
- [Balakrishna et al., 2008c] Balakrishna, P., Ganesan, R., Sherry, L., and Levy, B. (2008c). Estimating taxi-out times with a reinforcement learning algorithm. In *27th Digital Avionics Systems Conference*. DASC.
- [Baras and Patel, 1995] Baras, J. S. and Patel, N. S. (1995). Information state for robust control of set-valued discrete time systems. In *Proceedings of the 34th Conference, Decision and Control (CDC)*, page 2302.
- [Barrer et al., 1989] Barrer, J., Swetnam, G., and Weiss, W. (1989). The feasibility study of using computer optimization for airport surface traffic management. Report, The Mitre Corporation, Mclean, VA. MTR89W00010.
- [Bayard, 1991] Bayard, D. S. (1991). A forward method for optimal stochastic nonlinear and adaptive control. *IEEE Transactions on Automatic Control*, 36:1046–1053.
- [Beatty et al., 1999] Beatty, R., Hsu, R., Berry, L., and Rome, J. (1999). Preliminary evaluation of flight delay propagation through an airline schedule. *Air Traffic Control Quarterly*, 7:259–270.

- [Bellman, 1954] Bellman, R. (1954). The theory of dynamic programming. *Bull. American Math Society*, 60:503–516.
- [Bellman, 2003] Bellman, R. (2003). *Dynamic Programming*. Dover Publications.
- [Bertsekas and Tsitsiklis, 1996] Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific.
- [Brinton et al., 2002] Brinton, C., Kozel, J., Capozzi, B., and Atkins, S. (2002). Improved taxi prediction algorithms for the surface management system. In *AIAA Guidance, Navigation, and Control Conference*. AIAA 2002-4857, Monterey Bay, CA.
- [Burrus et al., 1998] Burrus, C., Gopinath, R., and Guo, H. (1998). *Introduction to Wavelets and Wavelet Transforms: A Primer*. Prentice-Hall, Inc.
- [Callaham et al., 2001] Callaham, M. B., DeArmon, J., Copper, A., Goodfriend, J., Monch-Mooney, D., and Solomos, G. (2001). Assessing nas performance: Normalizing for the effects of weather. In *Proceedings of the 4th USA/Europe Air Traffic Management Research and Development Symposium, Santa Fe, N.M.*
- [Carr et al., 2002] Carr, F., Evans, A., Clarke, J.-P., and Feron, E. (2002). Modeling and control of airport queueing dynamics under severe flow restrictions. In *Proceedings of the 2002 American Control Conference*, Anchorage, AK. American Control Conference.
- [Castillo and Hurwitz, 1997] Castillo, D. E. and Hurwitz, A. (1997). Run-by-run process control: Literature review and extensions. *Journal of Quality Technology*, 29(2):184–196.
- [Cheng et al., 2001] Cheng, V., Sharma, V., and Foyle, D. (2001). A study of aircraft taxi performance for enhancing airport surface traffic control. *IEEE Transactions on Intelligent Transportation Systems*, 2(2).
- [clow et al., 2004] clow, M., Howard, K., Midwood, B., and Oiesen, R. (2004). Analysis of the benefits of surface data for etms. Report, Volpe National Transportation Systems Center. VNTSC-ATMS-04-01.
- [Coifman and Maggioni, 2004] Coifman, R. and Maggioni, M. (2004). Diffusion wavelets. <http://www.math.duke.edu/mauro/diffusionwavelets.html>.
- [Compart, 2008] Compart, A. (2008). No cap on delays at ny airports.
- [Cooper et al., 2001] Cooper, W. J., Cherniavsky, E., DeArmon, J., Glenn, J., Foster, M., Mohleji, S., and Zhu, F. (2001). Determination of minimum pushback-time predictability needed for near-term departure scheduling using departs. Report, The Mitre Corporation, Mclean, VA.
- [Dareing and Hoitomt, 2002] Dareing, S. W. and Hoitomt, D. (2002). Traffic management and airline operations. In *Proceedings of the 2002 American Control Conference*, volume 2, pages 1302–1307.
- [Denardo, 2003] Denardo, E. (2003). *Dynamic Programming: Models and Applications*. Dover Publications.

- [Donoho, 1993] Donoho, D. L. (1993). Unconditional bases are optimal bases for data compression and for statistical estimation. *Applied and Computational Harmonic Analysis*, 1(1):100–115.
- [Donohue and Shaver III, 2008] Donohue, G. and Shaver III, R. (2008). *Terminal Chaos: Why U.S. Air Travel is Broken and How to Fix It*. American Institute of Aeronautics and Astronautics Inc.
- [Fleming, 2008] Fleming, S. (2008). National airspace system. Report, U.S. Government Accountability Office.
- [Futer, 2006] Futer, A. (2006). Improving etms ground time predictions. In *25th Digital Avionics Systems Conference*. DASC,IEEE,AIAA.
- [Ganesan et al., 2007] Ganesan, R., Das, T., and Ramachandran, K. (2007). A stochastic dynamic programming approach to run-by-run control. *IEEE Transactions on Automation Science and Engineering*, 4(2).
- [Goldburg and Chesser, 2008] Goldburg, B. and Chesser, D. (2008). Sitting on the runway: Current aircraft taxi times now exceeds pre-9/11 experience. Report, Dept. of Transportation, Bureau of Transportation Statistics.
- [Gosavi, 1998] Gosavi, A. (1998). *An Algorithm for solving Semi-Markov Decision Problem using Reinforcement Learning: Convergence Analysis and Numerical Results*. PhD thesis, University of South Florida.
- [Gosavi, 2003] Gosavi, A. (2003). *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer Academic Publishers.
- [Ho et al., 1992] Ho, Y. C., Sreenivas, R., and Vakili, P. (1992). Ordinal optimization of dedcs. *Journal of Discrete Event Dynamics*, 2(2):61–88.
- [Howard, 1960] Howard, R. (1960). *In Dynamic Programming and Markov Processes*. MIT Press.
- [Idris et al., 1999] Idris, H., Anagnostakis, I., Decaire, B., Clarke, J., Hansman, R., Feron, E., and Odoni, A. (1999). Observations of departure processes at logan airport to support the development of departure planning tools. *Air Traffic Control Quarterly*, 7(4):229–257.
- [Idris et al., 2002] Idris, H., Clarke, J., Bhuva, R., and Kang, L. (2002). Queueing model for taxi-out time estimation. *Air Traffic Control Quarterly*.
- [Idris et al., 1998] Idris, H., Delcaire, B., Anagnostakis, I., Hall, W., Pujet, N., Feron, E., Hansman, R., Clarke, J., and Odoni, A. (1998). Identification of flow constraints and control points in departure operations at airport system. In *Proceedings AIAA Guidance, Navigation and Control Conference*, Boston, MA. AIAA 98-4291.
- [Idris and Hansman, 2000] Idris, H. and Hansman, J. R. (2000). Observation and analysis of departure operations at boston logan international airport. Report, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, 02139.

- [JPDO, 2008] JPDO (2008). The next generation air transportation system integrated plan. Report, Joint Planning and Development Office.
- [Laskey et al., 2006] Laskey, K., Xu, N., and Chen, C.-H. (2006). Propagation of delays in the national airspace system. Report, George Mason University.
- [Levine and Gao, 2007] Levine, B. and Gao, O. (2007). Aircraft taxi-out emissions at congested hub airports and the implications for aviation emissions reduction in the united states. In *CD-ROM*, Washington DC. Transportation Research Board, Annual Meeting.
- [Levy and Legge, 2008] Levy, B. and Legge, J. (2008). Objective and automatic estimation of excess taxi-times. In *Integrated Communications, Navigation and Surveillance Conference*, volume 5-7, pages 1–10. IEEE.
- [Lindsay et al., 2005] Lindsay, K., Greenbaum, D., and Wanke, C. (2005). Pre-departure uncertainty and prediction performance in collaborative routing coordination tools. *Journal of Guidance, Control, and Dynamics*, 28(6).
- [Mahadevan and Maggioni, 2005] Mahadevan, S. and Maggioni, M. (2005). Value function approximation with diffusion wavelets and laplacian eigenfunctions. Report, University of Massachusetts, Department of Computer Science, Amherst, MA, 01003.
- [Martin et al., 2001] Martin, P., Murray, R., and Rouchon, P. (2001). *Flat Systems, Equivalence and Feedback*. Springer.
- [Matveev and Savkin, 2000] Matveev, S. and Savkin, A. (2000). *Qualitative Theory of Hybrid Dynamical Systems*. Birkhauser.
- [Meier and Eriksen, 2006] Meier, C. and Eriksen, P. (2006). Total airport management: A step beyond airport collaborative decision making. Report, EUROCONTROL.
- [NextGen, 2009] NextGen (2009). Nextgen 2018. <http://www.faa.gov/about/initiatives/nextgen/2018/>
- [Powell, 2007] Powell, W. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience.
- [Pujet et al., 1999] Pujet, N., Delcaire, B., and Feron, E. (1999). Input-output modeling and control of the departure process of busy airports. In *AIAA Guidance, Navigation, and Control Conference*, Portland, Oregon.
- [Robbins and Monro, 1951] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.
- [Rowland, 2009] Rowland, T. (2009). L2-space. <http://mathworld.wolfram.com/L2-Space.html>.
- [Rudin, 1976] Rudin, W. (1976). *Principles of Mathematical Analysis*. McGraw-Hill, Inc.
- [Schaft and Schumacher, 2000] Schaft, V. d. and Schumacher, H. (2000). *An Introduction to Hybrid Dynamical Systems*. Springer.
- [Schumer and Maloney, 2008] Schumer, C. and Maloney, C. B. (2008). Your flight has been delayed again. Report, Joint Economic Committee Majority Staff.

- [Shumsky, 1995] Shumsky, R. (1995). *Dynamic Statistical Models for the Prediction of Aircraft Take-Off Times*. PhD thesis, Operations Research Center, MIT, Cambridge.
- [Shumsky, 1997] Shumsky, R. (1997). Real time forecasts of aircraft departure queues. *Air Traffic Control Quarterly*, 5(4).
- [Signor and Levy, 2006] Signor, D. and Levy, B. (2006). Accurate oooi data: Implications for efficient resource utilization. In *25th Digital Avionics Systems Conference*. DASC.
- [Source, 2007] Source, I. (2007). Technical directive no. 15: Bureau of transportation statistics. www.bts.gov.
- [Source, 2009] Source, I. (2009). John f. kennedy international airport. <http://en.wikipedia.org/>.
- [Spall and Cristion, 1998] Spall, J. and Cristion, J. (1998). Model-free control of nonlinear stochastic systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 43:1198–1210.
- [Tu, 2006] Tu, Y. (2006). *Air Transportation Systemm Performance: Estimation and Comparative Analysis of Departure Delays*. PhD thesis, University of Maryland.
- [Tu et al., 2005] Tu, Y., Ball, M., and Wolfgang, J. (2005). Estimating flight departure delay distributions - a statistical approach with long-term trend and short-term pattern. Report, Robert H. Smith School. RHS 06-034, <http://ssrn.com/abstract=923628>.
- [UMass, 2009] UMass (2009). Reinforcement learning repository. <http://www-anw.cs.umass.edu/rlr/>.
- [Welch et al., 2001] Welch, J., Bussolari, S., and Atkins, S. (2001). Using surface surveillance to help reduce taxi delays. AIAA Guidance, Navigation, and control Conference, AIAA-2001-4360 Montreal, Quebec.
- [Werbos, 1998] Werbos, P. (1998). Stable adaptive control using new critic designs. Report, National Science Foundation, Washington DC. <http://arxiv.org/html/adap-org/9810001>.
- [Wonham, 1979] Wonham, W. (1979). *Linear Multivariable Control: A Geometric Approach*. Faller-Verlag.
- [Xu et al., 2008] Xu, N., Sherry, L., and Laskey, K. (2008). Multifactor model for predicting delays at u.s. airports. *Transportation Research Record*, 2052:62–71.

Curriculum Vitae

Poornima Balakrishna obtained a Bachelor's degree in Electronics and Communication Engineering from Madras University, India, in June 2002. She received her Masters degree in Industrial Engineering and Operations Research from the University of Oklahoma, in May 2005. Her Masters thesis research involved the investigation of spherical form tolerance using support vector machines. In August 2005 she joined the Systems Engineering and Operations Research doctoral program at George Mason University, where she has been a teaching assistant, research assistant, and graduate lecturer for a Systems Modeling Laboratory course. She currently works as a Junior Research Engineer at Sensis Corporation in Reston, Virginia. Her research interests include simulation-based learning, large scale optimization, stochastic processes, wavelet analysis, and air transportation applications.