Modeling and Control of Networked Systems: Applications to Air Transportation

by

Karthik Kavassery Gopalakrishnan
B.Tech., Indian Institute of Technology Madras (2014)
S.M., Massachusetts Institute of Technology (2016)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2021

Author . . . . . .
Department of Aeronautics and Astronautics
June 30, 2021

Certified by . . . . . .
Hamsa Balakrishnan
William E. Leonhard (1940) Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by . . . . . .
Hari Balakrishnan
Fujitsu Professor of Computer Science and Artificial Intelligence
Doctoral Committee Member

Certified by . . . . . .
Richard Jordan
Health Data Analytics Institute
Doctoral Committee Member

Accepted by . . . . . .
Jonathan How
Chair, Graduate Program

Modeling and Control of Networked Systems: Applications to Air Transportation

by

Karthik Kavassery Gopalakrishnan

Submitted to the Department of Aeronautics and Astronautics
on June 30, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Aeronautics and Astronautics

Abstract

Growing air traffic has resulted in congestion and flight delays. Delays not only inconvenience passengers but also have negative environmental impacts and cause monetary losses for airlines. Reducing delays is therefore crucial for operating a sustainable, efficient, and robust aviation infrastructure. Network analysis has been a popular tool to study large-scale interconnected systems due to its analytical and computational tractability. However, time-varying topologies, multi-layered interactions, and the inability to model the high variability in flight delays have limited the utility of traditional network models in aviation applications. In this dissertation, we use ideas from switched-systems theory, graph signal processing, and machine learning to develop tools that overcome some of these limitations.

The key idea behind our modeling approach is to (i) simplify the complex network interactions by identifying a small, finite set of representative network topologies, (ii) use data to learn the delay dynamics for each individual topology, and (iii) identify an appropriate topology transition policy to model the dynamics on time-varying networks. We call this the Markov jump linear system (MJLS) model for airport delays. We develop this delay model for the US, validate it, and demonstrate its superior predictive performance in comparison to other benchmarks. Next, we use this model to identify appropriate interventions that can minimize delays, and develop novel controllers for regulating delays. Our findings suggest that (i) optimal interventions that target highly-connected airports at Atlanta, San Francisco and Chicago can provide maximum system-wide benefits, and (ii) the most effective time for such interventions is between 11 and 2 pm ET. The methods developed in this dissertation can help airlines and air traffic managers improve the efficiency and robustness of the air transportation system.

Thesis Supervisor: Hamsa Balakrishnan
Title: William E. Leonhard (1940) Professor of Aeronautics and Astronautics

Acknowledgments

No words are sufficient to express my gratitude to my advisor, mentors, collaborators, friends and family for their support during grad school. I want to start by thanking my advisor Prof. Hamsa Balakrishnan. She has not only helped me mature and develop as a researcher by giving me so many opportunities to learn and explore my interests, but also been extremely kind and caring mentor. Thank you Prof. Hari Balakrishnan and Prof. Mohammad Alizadeh for inspiring me to search hard to find the right question, and introducing me to the fascinating world of computer systems and networking. Thank you Dr. Richard Jordan for guiding me in the early part of my PhD and for volunteering to serve on my committee. A special thanks to Prof. Daniel Delahaye for inviting me to Toulouse during 2019 and for the feedback on my thesis. And finally, thanks Prof. Saurabh Amin for your comments and serving as a reader for my dissertation.

During Spring 2020, I had the wonderful opportunity to be a TA for 16.09 – I am grateful to Prof. Youssef Marzouk for the opportunity. I also had a chance to gain some industrial research experience through a summer internship at Honeywell – thanks Dr. Alberto Speranzon for hosting and mentoring me. It has been a pleasure collaborating with Prof. Lavanya Marla and her group from UIUC during the last few years of my PhD. During my PhD, I also had the opportunity to collaborate with several groups outside MIT. This includes Dr. Richard DeLaura and Dr. Tom Reynolds from MIT Lincoln Labs; Dr. Husni Idris at NASA; Luis Mesen, Mark Talaga and Marc Brodbeck from United Airlines; and Dr. Antony Evans, Maxim Egorov and Vanessa Kuroda from Airbus. I really appreciate the opportunity to learn and work with all of you. I want to thank everyone in the lab, who made it a joy to come in every day. In particular, working with Arjun, Venkat, Max, Chris, Gerard, Siddharth, Baris, Akila and Simran has been really enjoyable – thank you all. Thank you Sandeep, Yash, Jackie and Joao for your company and friendship.

I want to thank friends and family for all the good times over the several years. In particular, thanks to Sandeep, Krishna, Raja, Rushina, Rohit, Arko and Nigamaa. Thank you Krithika for your wonderful company over the past several years. And finally, thanks to my parents, sister, grandparents, cousins, aunts and uncles, especially the ones at New Jersey. Without their encouragement, support, and help, I would not have come this far along to get a PhD from MIT!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Air travel has revolutionized our lives by offering a fast and safe way to travel across the world. More than 4.5 billion passengers, and trade estimated to be valued at $6.7 trillion, were transported on 39 million flights in 2019 [4]. Air traffic demand continues to grow and its connectivity has nearly doubled in the past two decades to cover more than 22,000 unique city-pairs worldwide [4]. Despite the disruption in air travel in 2020-21 due to the COVID-19 pandemic, air traffic is poised for further growth in the coming years [5, 3, 6].

1.1 Motivation

At the most basic level, an effective commercial aviation system transports people and goods from airport A to airport B at a promised time. Unfortunately, this goal is not always satisfied due to delays or cancellations. During 2019, 19% of the flights in the United States (US) had an arrival delay of greater than 15 minutes, and another 2.5% of flights were canceled. Flight delays are estimated to have an economic impact of $30-40 billion each year in the US, and $60 billion worldwide [2, 7, 8, 9] (see Figure 1.1). These statistics translate to a cost of over $300 per minute of flight delay. The problem of high delays and cancellations is only expected to get worse with increasing traffic. Over the next two decades, commercial air traffic poised to increase by 80% [5] and low-altitude flights (which includes package delivery drones, or air taxis) are expected to grow by several orders of magnitude [3].

The large magnitude of delays and cancellations is a cause of concern to several stakeholders:

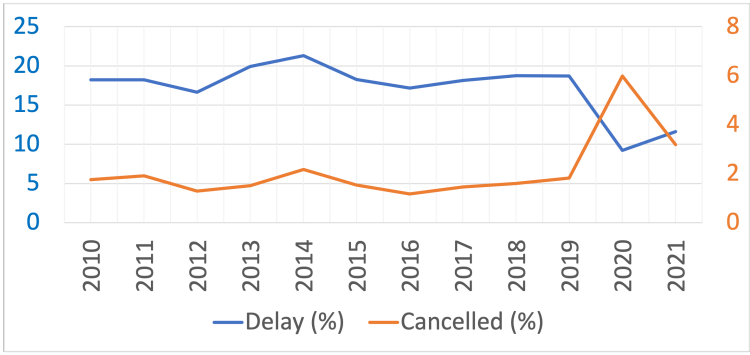


Figure 1-1: Percentage of delayed and canceled flights in the US. The high rate of cancellation and reduced delays in 2020 are due to the impact of the COVID-19 pandemic. The system is expected to return back to normal by 2022.

Cost component	Cost ($ billions)
Cost to airlines	8.3
Cost to passengers	16.7
Costs from lost demand	3.9
Total direct cost	28.9
Impact on GDP	4.0
Total cost	32.9

Table 1.1: Breakdown of the estimated annual cost of flight delays [2].

- **Passengers**: Significant amounts of time and money are lost when passengers are not able to reach their destination as per the original schedule. While delays of a few minutes do not have a significant impact, certain disruptive events that last for several hours or days can have very negative consequences [10].

- **Airlines**: Delays and cancellations often require airlines to pay their crew for longer hours, burn more fuel than anticipated, and spend significant money to re-schedule itineraries for their passengers (or cargo). This not only erodes short-term profitability, but can also lead to poorer customer satisfaction rates that hurts long-term competitiveness [11, 12].

- **Air Traffic Managers**: The goal of system administrators, such as the Federal Aviation Administration (FAA) in the US, or EUROCONTROL in Europe, is to ensure safe, efficient, robust, and environmentally sustainable operations. Delays and cancellations indicate poor resource utilization and planning. Excess delays can lead to higher emissions, and in worse cases, low-fuel scenarios which may compromise safety [13].

- **Advanced Air Mobility Operators**: Excessive delays and the inability to operate a well-

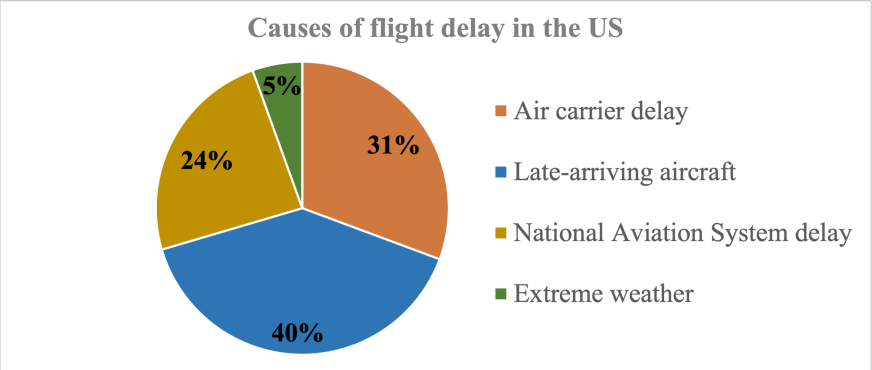


Figure 1-2: Causes of flight delay in the US in 2019 [1].

orchestrated schedule can render several emerging services such as on-demand air taxi and drone-deliveries economically infeasible, or even unsafe [14].

A deeper analysis of the main causes of flight delays is presented in Figure 1-2. The different categories used in the figure are listed below:

- **Late-arriving aircraft:** A single aircraft completes multiple flight legs each day to increase its utilization. Aircraft in the US typically operate 4-6 flight legs a day; some aircraft fly as many as 10 flights in a single day [15]. Although airline schedules are planned with buffers between flight legs to mitigate delay propagation, the delay of the incoming aircraft may exceed the buffer, resulting in reactionary delays.

- **Air carrier delays:** This category includes all causes of delay that are within the control of an airline, such as maintenance or flight crew issues, passenger boarding, fueling, baggage loading, aircraft cleaning, etc.

- **National Airspace System (NAS) delays:** These are delays due to air traffic control and traffic management initiatives, for example, to address reduced capacity due to inclement weather or heavy traffic volume at an airport or in the airspace.

- **Extreme weather:** This category includes severe meteorological conditions such as hurricanes, tornadoes, or blizzards, that prevent the safe operation of flights.

Figure 1-2 presents data on the causes of flight delays. The figure is generated as follows: whenever a flight is delayed, the airline operating that flight partitions the total delay minutes into

these four mutually exclusive categories. For example, if a flight is delayed by 50 minutes, the airline can report that 40 minutes were because the inbound flight leg was late, and the other 10 minutes were because of maintenance issues. These results are then aggregated over all flights and presented in Figure 1-2. This figure indicates that the largest share of delays (i.e., 40%) is due to the late arrival of incoming aircraft. This means that (i) any local disruption that causes flight delays can easily spread to other airports, and (ii) airports with perfect local operating conditions can have delayed flights. Since propagated delays (also referred to as reactionary delays) are the biggest contributors to flight delays, they are the primary target of our efforts to reduce delays. The first objective of the dissertation is to improve system efficiency by reducing flight delays and cancellations.

Improving efficiency alone does not suffice. Our second objective is to improve the robustness (or resiliency) of the system. This means that:

- The system can handle disruptions without a significant decrease in performance or safety. In other words, minor flight delays should not magnify and lead to more extreme outcomes such as flight cancellations.

- The system isolates faults and disturbances without letting it cascade to other nominal components. This means that a thunderstorm over the New York City airports (or any localized region), although detrimental to the operations at those airports should ideally not result in significant delays at other geographically distant airports like Chicago or Atlanta.

- The system can recover quickly after disruptions and restore nominal performance. We desire that the delays and cancellations resulting from disruptions (e.g., due to poor weather) decay quickly (order of a few hours).

A look at aviation system performance data suggests that there is significant scope for improvement along these three dimensions. For instance, Figure 1-1 and 1-2 show that (i) extreme outcomes (i.e., cancellations) are relatively frequent, and (ii) a large fraction of delays (approximately 40%) cascade and spread through the system. Furthermore, several examples show that the effect of disruptive weather can persist even days after the disruption [16, 17, 18]. Recent advances in operational strategies have involved proactive management, which tends to shift some of the re-

planning burdens on the period before the disruption to minimize overall impacts [19, 20]. These observations highlight the need to improve the robustness of the aviation infrastructure.

Some of the requirements for improving system efficiency and robustness are as follows:

1. **Estimation of performance metrics**: We cannot improve what we cannot measure. Hence, any improvements in system efficiency or robustness require us to rigorously estimate the current performance levels of the air transportation system. These estimates should rely on operational data and should explicitly incorporate network-based approaches that can quantify the effect of propagated delays.

2. **Identification of critical elements**: Prioritization can help us allocate limited resources more effectively. Thus, we should use system measurements to quantify the relative (or absolute, if possible) roles played by different components of the infrastructure in managing disruptions. For example, identifying airports, or airspace sectors, or times-of-day that are most susceptible to generating cascading delays will help in developing a strategic infrastructure modernization plan.

3. **Prediction of future system behavior**: Knowing the future helps us prepare for it. Thus, an accurate model for the spread of delays, or cascading effects from local disturbances, is required to proactively mitigate the propagation of delays.

4. **Design of interventions**: Every disruption requires customized interventions for optimal recovery. The identification of appropriate interventions in turn needs the three tools mentioned above: measuring the current state of the system, predicting the future spread of delays, and prioritizing limited resources to restore system performance in minimal time. Potential interventions include a combination of several actions such as flight re-routes, airport slot swaps, or even tactical cancellation of certain flights or closure of selected routes.

The high monetary and environmental costs of disruptions have spurred research on developing solutions for several decades. The classical approach involves *large-scale discrete optimization* techniques that leverage models of microscopic interactions to control and predict the system behavior [21, 22, 23, 24, 25]. The key idea is to use all available information (e.g., delay state of each aircraft, the capacity of airport runways, weather-related airspace closures, flight schedules,

etc.) to study the performance of the larger system. Examples of this approach include works on queuing theory models [26, 27], stochastic resource allocation mechanisms [28, 29], and discrete-event simulations [30, 31]. The major drawback of these techniques is that they have been only applied to systems involving a few airports, or a few thousand flights [32, 25, 33]. Thus, they are not computationally tractable for realistic problem sizes seen today (see [34] for a recent advance), leaving limited scope for analyzing larger systems (with drones, air taxis, and more commercial aircraft) we are anticipating over the coming years.

In this dissertation, we look beyond optimization-based approaches, which have high data and computational requirements, and merge (and leverage) two recent developments: (i) *data-driven techniques* to build predictive models for aviation disruptions, and (ii) analysis of large-scale systems using graph-theoretic and *network models*.

Data-driven approaches, while being accurate in their ability to predict delay dynamics, often lack interpretability and analytic tractability [35, 36, 37]. This limits their usefulness in developing interventions to improve system performance. On the other hand, network analysis and models are typically interpretable, analytically tractable, and scalable. However, the analysis of air traffic networks presents its own set of challenges:

1. **Coupling of airport states and network topology**: Air traffic networks exhibit a strong coupling between the airport delay state and the network topology. The network topology changes with time (e.g., due to flight schedules) and is also a function of the state of the airports of the network (e.g., airport delays). Such complex intertwining of the airport states and the underlying topology poses technical challenges to the modeling and analysis of these systems.

2. **Multilayer networks**: Connectivity in the air transportation system has been traditionally modeled only in terms of operations, that is, flight service between an origin and a destination [38, 39, 40, 41]. However, the air transportation system is *multilayered*: the nodes (airports) may interact along multiple dimensions, each of which can be treated as a layer of the network [42]. For example, one can represent traffic flows on one layer of the network, airport capacities on another layer, and flight delays on a third layer of the network (see Figure 1-3). These layers would in turn be connected; for instance, bad weather can

lead to a reduction in airport capacities, which can, in turn, increase the delays on the network. High delays can lead to network-wide cancellations, thereby modifying the traffic on the network. In other words, the air transportation system is a multilayer network over which delays and other disruptions emerge and spread. Limited methodological tools exist for developing analytical, or even data-driven, models for such real-world networks.

3. **Complex and time-varying dynamics**: Air traffic networks exhibit significant variability (e.g., in weather patterns and airline schedules), inducing seasonal, weekly, and daily trends; furthermore, weather phenomena are uncertain. Consequently, it is difficult to analyze and model the dynamics of air traffic delays even when operational data is available, since one needs to be able to control for the changes in capacity and demand.

4. **Airline competition**: Constrained air traffic network resources are shared by aircraft operated by multiple competing airlines, each with its own utility function. These utility functions govern their scheduling and operating practices and affect the dynamics of the overall system.

5. **Data quality and privacy**: Information and data are a key part of modeling, analysis, and optimization. In data-driven analysis, it is important to note that some data (e.g., the reasons for delay) can be subjective, resulting in potential biases in the conclusions. Some data involving airline operating practices and cost functions are proprietary, and the flight plans of certain aircraft may be confidential, resulting in only partial information on some system states.

6. **Utility of coarse models**: A concern with network models is that they are often too coarse for any practical use. For example, network models for flight delays, such as those studied in [31, 43, 44, 45], (i) consider average delays at airports or routes rather than flight-specific delays, and (ii) use continuum flow approximations instead of actual flight scheduled. Although recent works [46, 47, 48, 49] have started bridging the gap between coarse, macroscopic analysis and microscopic decisions, much work remains to be done.

This thesis attempts to combine the network modeling approach with data-driven methods to develop techniques that can (i) scale for large traffic volumes, (ii) represent system dynamics

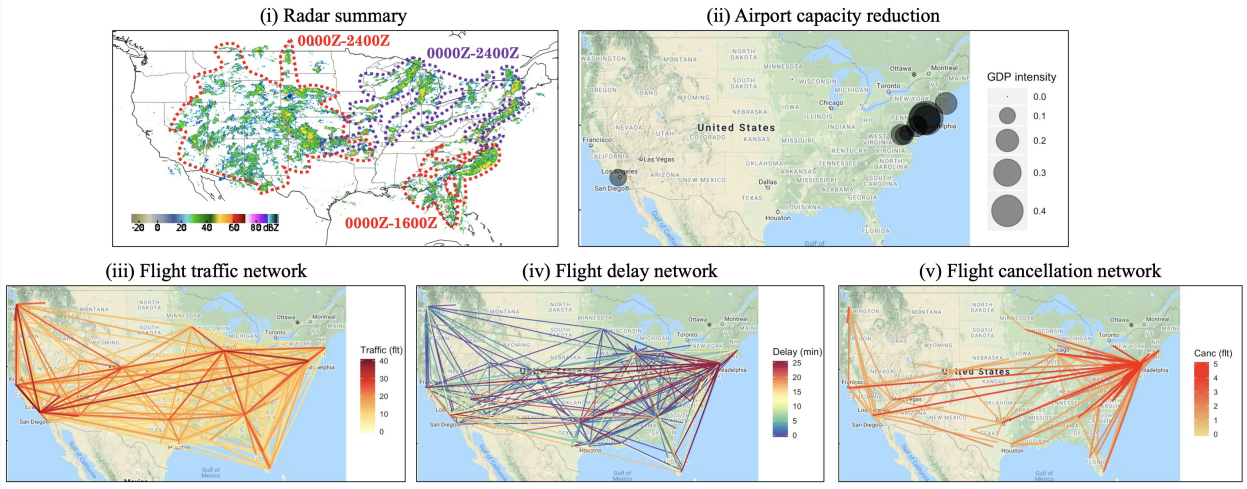


Figure 1-3: Different representations of the state of the air transportation system on July 1, 2016. Although networks in (iii)-(v) are directed, an average undirected network is shown for ease of visualization.

accurately, and (iii) identify interventions to improve efficiency and robustness.

In the next section, we will summarize our work on overcoming these limitations and highlight our major contributions.

1.2 Contributions of this Dissertation

The primary motivation of this dissertation is to design an efficient and robust air transportation infrastructure. This requires an accurate model for the evolution and propagation of airport delays. Current models for delay propagation are not scalable, accurate, and suitable for identifying control actions that improve efficiency and robustness. In this dissertation, we develop a Markov jump linear system (MJLS) model that is scalable, accurate, and can be used for developing controllers that improve efficiency and fairness.

Our dissertation is organized into four segments. First, we develop a few individual components of the model. In particular, we identify representative topologies for delay interactions (Chapter 2) and propose delay propagation models for fixed topologies (Chapter 3). Second, we piece together the components of the model to develop a MJLS model for the dynamics of delays under switching network topology (Chapter 4). Third, we validate the MJLS model that was developed using actual delay data. This involves studying the asymptotic behavior of the model using stability analysis techniques (Chapter 4), as well as comparing its predictive performance against other

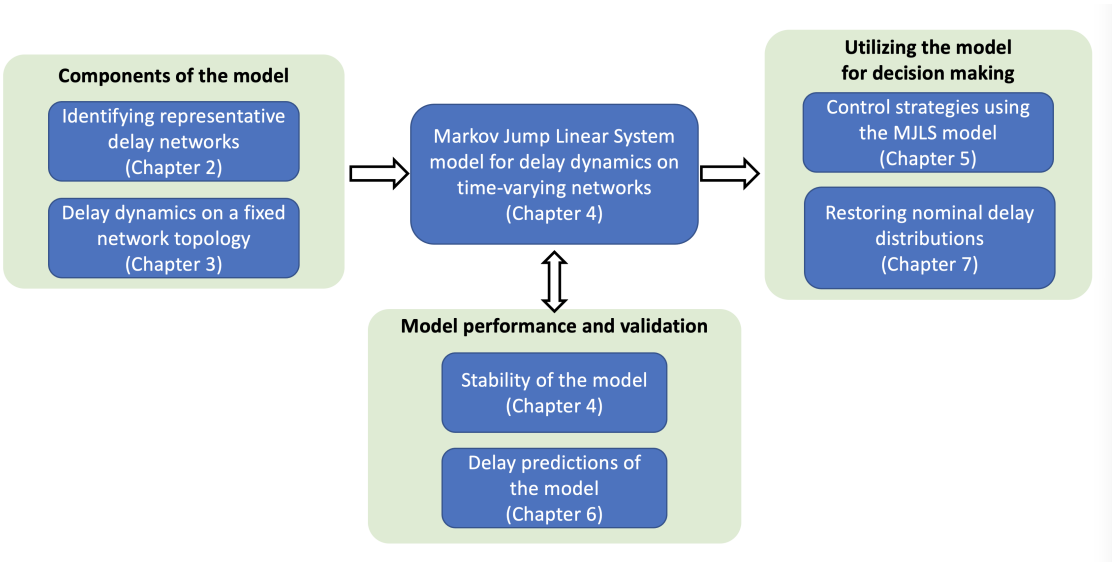


Figure 1-4: Organization of the dissertation and our main contributions.

standard benchmarks (Chapter 6). Finally, we use the MJLS model to propose interventions that can improve the system. In this regard, we identify optimal interventions that can reduce the total system delays (Chapter 5), and explore the distribution of delays across airports resulting from such interventions (Chapter 6). Finally, we study the distribution of delays across airports and propose that we not only develop controllers that minimize the total delay but also restore the nominal distributions of delays across airports (Chapter 7). An overview of the organization and main contributions of this dissertation is presented in Figure 1-4.

1. **Efficient identification of representative delay networks.** To develop models for delay evolution on time-varying networks, we simplify the model and identify a set of representative network topologies. This involves the clustering of a time-series of weighted-directed graphs. Prior works have constructed feature vectors for clustering graphs by concatenate edge attributes. However, these feature vectors ignore the network structure of the data and have a length that is quadratic in the number of nodes. We develop a new feature vector to cluster a time series of weighted-directed graphs. Our feature vector uses the hub and authority scores of the underlying graph to incorporate network information and has a size only linear in the number of nodes, improving computational tractability.

2. **Markov jump linear system model for delay dynamics on time-varying networks.** Typically, network models assume that the underlying network topology is constant. We use

switch systems theory to introduce a new class of models for time-varying networks. Our model (called the Markov jump linear system) considers Markov switching between representative network topologies, with a distinct linear delay dynamics for for each topology. This model is analytically tractable and its parameters can be learned from real-world data. We show that our model predicts airport delays with lower error than standard benchmark techniques.

3. **Stability of the Markov jump linear system model.** Air traffic delays decay with time, and we would like our MJLS model to reflect this. Unfortunately, stochastic switching makes the analysis of the asymptotic behavior of the model challenging. Recently, the stability (i.e., tendency of system-state to decay with time) of positive Markov jump linear systems (positive since the delays are always non-negative) with (i) continuous-time representation and (ii) time-independent transition probabilities have been studied in [50]. To perform a sanity check for our delay dynamics model, we extend the prior work on stability analysis to discrete-time positive MJLS models with time-varying transition probabilities. By using tools from linear algebra and probability theory, we derive conditions for the mean-stability and almost-sure-stability of our delay dynamics model.

4. **Controllers for Markov jump linear systems.** The state of a MJLS network model consists of continuous variables denoting the airport delays, and the discrete network topology denoting the interconnections in the system. Prior works on controlling a positive MJLS have focused on actuating the continuous state and ignored the discrete mode transitions. We use the positivity property of the system to develop a new class of controllers that can actuate the continuous state as well as the discrete mode to minimize total system delay costs. The ability to modify both the airport delay and the airport connections jointly provides significant improvements in our ability to minimize the spread of delays in the system.

5. **Identifying unexpected distributions of airport delays.** Typically, improving the system efficiency means reducing the total delays across all airports. In the last part of our dissertation, we propose that in addition to the total delays, the distribution of delays across airports is also an important indicator of system performance. Thus, we formalize this concept by using tools from graph signal processing to develop a total variation metric. We show that this

metric can capture unexpected spatial delay distributions, and develop analytical and computationally tractable techniques to identify such delay patterns. Finally, we discuss some recent work towards incorporating the total variation metric in designing control actions that enable "smooth" recovery from disruptions.

1.3 Applications of our model

The techniques we develop in this dissertation can be used to enable more efficient and robust air traffic operations. We list some of these applications below.

1. **Benchmarking airline performance.** Performance analysis of the air traffic operations is challenging because of the need to account for weather impacts and network effects. We propose a framework that uses network clustering to identify baselines for benchmarking airline on-time performance, and highlight its utility with several case studies.

2. **Estimating airport resilience.** There has been a lack of quantifiable metrics to characterize the resilience of airports to delays. Such metrics are useful for understanding the robustness of the system, prioritizing infrastructure improvements, and improving system performance. We propose data-driven models to define and characterize airport resilience using two measures: (a) the influence of an airport on the network, and (b) the influence of the network on an airport. We compute and interpret these measures using flight delay data in the US.

3. **Predicting airport and route delays.** We compare the performance of different approaches (MJLS, CARTs, and neural networks) in predicting air traffic network delays. We show that prediction performance can vary significantly depending on the choice of model/algorithm, and identify the feature vectors that give the best predictive performance.

4. **Identifying anomalous airport delay distributions.** Aviation disruptions have been analyzed by looking at the magnitude and duration of flight delays. We show that unexpected differences in the magnitude of delays across airports can also be a useful measure of system state. We identify days with such anomalous delay distributions from 10 years of data (2008-2017) and discuss techniques for their further analysis and interpretation.

1.3.1 Applications to other networked systems

Networks are a powerful abstraction that have been successfully used to model the dynamics of a wide range of systems [51], from disease epidemics [52] and social networks [53], to engineered systems such as power grids [54, 55], the Internet [56], roads [57], public transport [58] and railroads [59]. Similar to the air transportation system, there is a desire to improve the operational efficiency and robustness of these networked systems, as we briefly describe for two examples below.

1. **The financial system.** Financial institutions not only lend and borrow money from customers but also from each other creating a web of transactions that interlinks their operations. However, when one bank issues a loan that does not get repaid, it may lead to ripple effects if the money was crucial for other payments. Hence, the financial institutions, and the regulators overseeing them would like to (i) identify critical institutions that are "too big to fail", and (2) develop risk mitigation strategies to ensure robust performance even under disruptions or shocks to the economy.

2. **The power grid.** The electricity grid connects power generators to consumers via an extensive network of physical wires, transformers, substations, etc. The grid performs the complex task of load balancing to match the dynamic demand and supply of electricity. An ideal power grid should (i) ensure that the dynamic demands are satisfied most efficiently, i.e., with minimal transmission loss, and (ii) ensure that the system can reconfigure quickly in the event of failures.

The study of the efficiency and robustness of complex systems such as the power grid [60, 61, 62] and financial institutions [63, 64] has been an active area of research. Several works have identified critical nodes in the network [65, 66] and control strategies to improve robustness [67, 68]. One of the major challenges in these works is effectively using limited data in modeling the dynamics of time-varying network topologies. We believe that the methods developed in this dissertation can potentially be applied to provide insights into the modeling and operation of these networked systems.

1.4 Organization of the Dissertation

- **Chapter 2:** In this chapter, we study delay network topologies, with an aim to identify representative toplologies that govern delay dynamics. This chapter can also be read independently as it uses representative topologies to tackle the problem of delay performance benchmarking, i.e., comparing whether the delays were good or bad relative to other representative historical scenarios. To solve this problem, we cluster graphs to develop a representative set of benchmarks. For clustering the graphs, we introduce a new feature vector based on the hub and authority scores of the nodes. Using these clusters for computing benchmarks, we show how we can analyze the delay and cancellation performance of the entire system (as well as individual airlines) on any given day.

- **Chapter 3:** We present a simple model for the spread of delays for a fixed network topology. This model incorporates traffic-dependant diffusion of delays at airports, variable time-scales of interaction among airports, absorption of delays due to schedule buffers, and impact of external disruptions. We show how this simple model can be used to quantify the resilience of the system to impulse and sustained disruptions. Finally, we discuss a case study, where we apply our technique to estimate the system-wide benefits of delay-reduction efforts at Charlotte.

- **Chapter 4:** In this chapter, we present our main contribution of the dissertation – a model for delay dynamics on time-varying networks. Here, we build and combine the work from Chapter 2 on identifying representative toplogies and Chapter 3 on constructing topology-dependent models. Our contribution is to develop a Markov jump linear system (MJLS) framework to represent dynamics on networks with varying topologies. We demonstrate the validity of the model by examining and interpreting its parameters as well as its ability to accurately capture temporal trends in delay dynamics. Finally, we also provide a theoretical guarantee that the MJLS model is reasonable and would never allow the delays to grow in an unbounded manner. In particular, we identify conditions for the Mean Stability and Almost Sure Stability of the MJLS model.

- **Chapter 5:** In this chapter, we use the MJLS model to identify interventions that can re-

duce the spread of delays. In other words, we develops algorithms to optimally control MJLS models. Using analytical methods, we identify the optimal combination of airport delay reductions and changes in network connectivity that can improve system recovery after disruptions. Finally, we show how we can leverage these results to develop practical delay-reduction strategies by identifying the most promising airports and time-of-day for these interventions.

- **Chapter 6:** We benchmark the accuracy of our MJLS model and compare its performance against several standard delay prediction approaches. In particular, we set up classification as well as regression problems for predicting OD pair and airport delays. We compare the performance of our MJLS model with other standard approaches such as neural networks, classification and regression trees, and linear regression. Through extensive evaluations, we find that the neural networks show superior performance for the delay classification problems, whereas the specialized MJLS model performs better for the airport delay regression problem. Our results should serve as a valuable baseline for future delay prediction research.

- **Chapter 7:** In this chapter, we consider the distribution of airport delays as a key system feature rather than just consider the total system delay. We introduce the notion of outliers-in-distribution identifying off-nominal delay distributions and use graph signal processing ideas to interpret the cause of such distributions. We end with a discussion on how such requirements on the distribution of delays can be incorporated into decision-making and controller design.

Finally, we conclude the dissertation with a discussion on directions for future research in Chapter 8.

Chapter 2

Graph Clustering for Performance Benchmarking

Large-scale, complex infrastructures, such as air transportation, are prone to operational inefficiencies which result in delays [69, 70]. It is therefore important to analyze system performance, identify operational inefficiencies, and reduce flight delays. The analysis of aviation system performance is confounded by several factors. Weather reduces airport and airspace capacities, resulting in flight cancellations and delays. Since no two days are identical in terms of the locations, timings and intensities of capacity impacts, it is difficult to distinguish between poor performance due to operational inefficiency and delays due to reduced capacity. Similarly, airport closures, security breaches, and equipment failures also impact capacity and cause delays. The complex dynamics of delay propagation in the air transportation network also poses a challenge to performance analysis.

Benchmarking is a standard technique used to evaluate system performance and identify potential operational deficiencies. It involves the comparison of an observed operational metric with a pre-computed reference value, or *baseline* [71, 72]. The baseline values may be different for different operational scenarios or contexts. Airlines have applied benchmarking principles in many aspects of their operations, including the analysis of financial performance, environmental impact [73], quality-of-service metrics [74], and airport performance [75, 76]. For example, by benchmarking their refueling and flight rotation practices with Formula 1 racing, Southwest Airlines reportedly reduced their refueling times by 70% [71].

This chapter addresses the need for performance baselines that are context-dependent, i.e., are

conditioned on the locations, timings, and intensities of the weather impacts. Such baselines allow us to control for extraneous weather effects when analyzing the operational efficiency of different stakeholders. Network models present a natural way to represent the air transportation system. The major portion of previous work in this area has considered models of traffic, either in terms of aircraft or passengers [38, 39, 40, 8, 41]. By contrast, we develop network models of air traffic delays, and describe how one might use operational data to build these models. A key difference between the problems is that network models of operations or passengers tend to be undirected when aggregated over several hours or a day (in other words, traffic levels are similar in both directions between two nodes), while delay network models tend to be directed.

This chapter models the US National Airspace System (NAS) at any time as a directed network, with airports as nodes and the current delays between origin and destination airports as weights on the edges. It then uses algorithms for graph clustering to identify characteristic delay states of the NAS. We use network clustering to develop context-dependant benchmarks to evaluate the operational performance of airlines and airports. The proposed framework has been deployed at a major US airline for conducting post-analysis of its daily on-time performance.

We will now present some of the prior works in this context.

Clustering approaches to computing baselines: Clustering has been previously used to develop appropriate case-specific baseline metrics from non-homogeneous observations [77, 78, 79, 80]. In air transportation networks, there is significant non-homogeneity due to the strong dependence of operational performance on the spatio-temporal occurrences of capacity reductions. However, prior works on clustering for identifying similar days in the National Airspace System (NAS) have typically focused on weather impacts and capacity constraints at a small subset of airports, or have only considered aggregate metrics such as total delays, cancellations, or traffic volume [81, 82]. Furthermore, these prior studies have not considered applying their identification of similar NAS days to stakeholder-specific on-time performance baselines.

An on-time performance benchmarking framework for air transportation using network-level clustering has not previously been considered. One concern with clustering approaches for benchmarking is that there may not be perfect homogeneity between days within the same cluster [83, 84]. We address this concern by incorporating the within-cluster variance to obtain statistically significant confidence bounds for our baselines.

Clustering similar Traffic Management Initiatives: Efforts to identify similar Traffic Management Initiatives (TMIs), in particular Ground Delay Programs (GDPs), have either focused exclusively on individual airports (e.g. Newark EWR [85, 86, 87], San Francisco SFO [87]), or a small subset of airports [84, 88]. Although these methods could be extended to the scale of the entire system, they do not address two key factors: the temporal parameters of the GDP (i.e. when it was issued), and the relative magnitudes of capacity reduction at different airports (i.e. how do we quantify the relative "intensity" of GDPs). For instance, in [89], the authors present a method that can be scaled to the system-level, but they do not consider temporal variation in airport capacity or its intensity. Weather has been the focus of several clustering efforts [90, 91], but convective conditions in and of themselves do not represent the actual reduction in airport or airspace capacity. Since GDPs are a more direct measure of the disruption that is induced by convective weather and other irregular operations, we will use these TMIs within our benchmarking framework.

In our work, we propose clustering network-centric *feature vectors* (that is, vectors of real numbers that represent the characteristics of similarity and dissimilarity between data observations) that scale for the entire system. In addition, these feature vectors also incorporate a normalized metric quantifying the intensity of the GDP, along with temporal information regarding GDP impact.

Clustering delay networks: Networks have been used to model several aspects of air transportation, including connectivity and delay dynamics [43, 92, 93]. The increased prevalence of data from networks has led to the development of techniques for clustering graphs [94, 95, 96, 97, 98, 99, 100, 101]. We emphasize the distinction between the *clustering of graphs* (finding groups of similar graphs), and *community detection* (finding clusters of nodes within a graph [102]). The choice of an appropriate feature vector for clustering strongly influences the quality of clusters. Prior work on clustering graphs has used feature vectors that include network-theoretic measures of centrality, such as degree centrality, between-ness centrality, and eigencentrality [95, 96, 97]; although the lengths of such features scale linearly with the size of the graph, they are best-suited only for unweighted, undirected graphs. By contrast, the clustering of days in the NAS requires feature vectors that can compare time-series of weighted, directed graphs; we therefore develop a suitable network-theoretic feature vector whose length scales linearly with the size of the graph.

Airline performance monitoring: Prior work on airline performance analysis has compared the aggregate statistics [1, 103], economic measures [104], or indicators of consumer satisfaction

and level-of-service of different airlines [105]. Methods for airlines to perform self- and peer-comparison of operational parameters such as on-time performance have not been considered. To the best of our knowledge, the explicit consideration of network state in benchmarking has also remained unexplored. While earlier studies have explored the role of vertical integration in the economic sense [106], there has been limited work on the network [107] and operational benefits [108] of regional airline subsidiary partnerships. Our analysis enables the comparison of the operational performance of regional carriers and their mainline counterparts.

Key contributions: The main contributions of this chapter are as follows.

1. We construct $O(N)$ size network-theoretic feature vectors for clustering a time-series of weighted and directed graphs. We highlight the value of these network-centric feature vectors in identifying characteristic delay and cancellation patterns in air traffic networks.

2. We cluster airport capacity impacts based on their timings, locations, and intensities, and use these clusters to develop performance baselines for airline on-time performance. We demonstrate the value of these baselines for monitoring and improving airline, and system-wide performance.

Chapter Outline: We start with a discussion on centrality measures in graphs in Section 2.1. We leverage these measures to engineer feature vectors for clustering in Section 2.2. The methodology developed in Section 2.2 is used to cluster delay, cancellation, and GDP networks in Section 2.3. Finally, the applications of these clusters for delay prediction, airline performance benchmarking, and system performance benchmarking are discussed in Sections 2.4, 2.5 and 2.6 respectively.

2.1 Measures of Centrality in Graphs

To cluster "similar" networks, we need to identify properties or features that can be used to evaluate their similarity to one another. Many candidate scores can be used to compare two networks; their suitability can vary depending on the application being considered [95, 96]. For example, one can compare the edge weights of two graphs, that is, use the Euclidean distance between the vectors of edge weights as a measure of graph similarity [101]. However, such a representation requires $O(n^2)$ parameters to represent a graph, where n is the number of nodes. In this work, we consider

promising features by which to compare networks that represent the delays in the air traffic system, and then use them to cluster similar networks, that is, identify characteristic delay states of the air traffic network. By considering features that represent the nodes and not the edges of the network, we obtain models with $O(n)$ parameters.

Degrees of nodes: The in- and out-degrees of a node in the air traffic network correspond to the total inbound and outbound delay (summed over all the incoming and outgoing edges from that node). In addition, one can also determine the total delay associated with a node as the sum of its inbound and outbound delays. For an airport i, these quantities can be determined from the adjacency matrix as:

$$d^{\text{out}}(i) = \sum_j A_{ij} \tag{2.1}$$

$$d^{\text{in}}(i) = \sum_j A_{ji} \tag{2.2}$$

$$d^{\text{tot}}(i) = d^{\text{out}}(i) + d^{\text{in}}(i) \tag{2.3}$$

In other words, the air traffic delay network at any time can be represented by the $n \times 1$ vectors, $\mathbf{d}^{\text{out}}$, $\mathbf{d}^{\text{in}}$ and $\mathbf{d}^{\text{tot}}$.

Eigenvector centrality: A measure of the importance of a node in a network is given by the *eigenvector centrality* [97, 98]. The eigenvector centrality of a node increases if that node is strongly connected to other nodes with high eigenvector centrality. The eigenvector centrality vector, $\mathbf{e}$, can be calculated from the adjacency matrix as the eigenvector corresponding to the largest eigenvalue of A. However, the eigenvalues of A are guaranteed to be real only if it is symmetric, which leads to the eigenvector centrality being a useful metric for undirected graphs. By contrast, the air traffic delay networks are directed weighted graphs.

One approach to applying the concept of eigenvector centrality to directed graphs is by "symmetrizing" the adjacency matrix, for example, by considering the graph with adjacency matrix $\tilde{A}$:

$$\tilde{A} = (A + A^{\text{T}})/2. \tag{2.4}$$

In the context of the air traffic delay network, the weight on each edge of the undirected graph is the average of the delays on the two corresponding edges of the directed graph.

Hub and authority scores: Hubs and authorities were introduced for directed networks in the context of the Hypertext Induced Topics Search (HITS) algorithm [109, 110]. The key idea is that there are two types of important nodes in directed networks: *hubs* and *authorities*. An important hub points strongly to many important authorities, while important authorities are nodes that are pointed to strongly by many important hubs. In the context of air traffic delay networks, a strong hub refers to an airport that has significant outbound delays to strong authorities (i.e., airports that have significant inbound delays from strong hubs). The hub scores (**h**) and authority scores (**a**) can be determined iteratively [109], or by considering the dominant eigenvectors of AA^{T} and $A^{\mathrm{T}}A$ respectively. Equivalently, they can be determined from the dominant eigenvector of $\mathscr{A}$ [111], given by

$$\mathscr{A} = \begin{pmatrix} 0 & A \\ A^{\mathrm{T}} & 0 \end{pmatrix}. \tag{2.5}$$

Since $\mathscr{A}$ is symmetric, the largest eigenvalue and corresponding eigenvector are guaranteed to be real. If A is an $n \times n$ matrix, then $\mathscr{A}$ is a $2n \times 2n$ matrix. The first n terms of its dominant (nonnegative) eigenvector correspond to the hub scores of the nodes of the directed graph, while the second n terms correspond to its authority scores [111].

Example 1. *Let us consider the example of the network shown in Figure 2-1.*

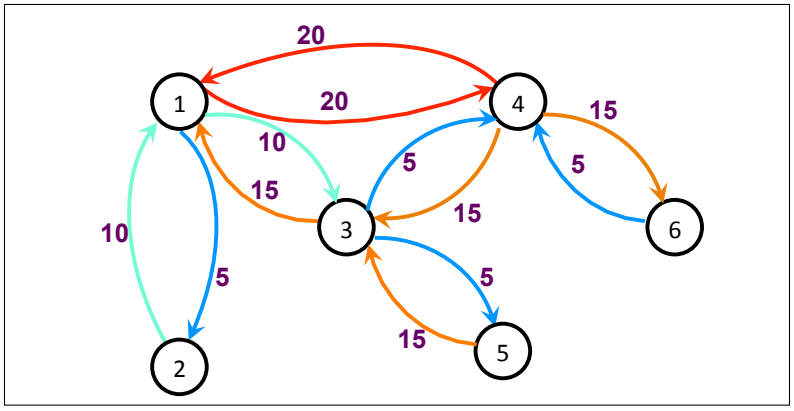


Figure 2-1: Example of a weighted directed network, where the weights on the edges is the departure delay between the airports (which are represented as nodes)

The vectors of incoming delays, outgoing delays and total delays at each airport are given by:

$$\mathbf{d}^{\text{in}} = \begin{bmatrix} 45 & 5 & 40 & 30 & 5 & 15 \end{bmatrix}^{\text{T}} \tag{2.6}$$

$$\mathbf{d}^{\text{out}} = \begin{bmatrix} 35 & 10 & 25 & 50 & 15 & 5 \end{bmatrix}^{\text{T}} \tag{2.7}$$

$$\mathbf{d}^{\text{tot}} = \begin{bmatrix} 80 & 15 & 65 & 80 & 20 & 20 \end{bmatrix}^{\text{T}} \tag{2.8}$$

While the quantities $\mathbf{d}^{\text{in}}$, $\mathbf{d}^{\text{out}}$ *and* $\mathbf{d}^{\text{tot}}$ *reflect the delays associated with each node in the network, they do not reflect the edges, that is, the connectivity of those nodes. The eigenvector centrality can be determined from the undirected graph derived using* (2.4)*, which is shown in Figure 2-2.*

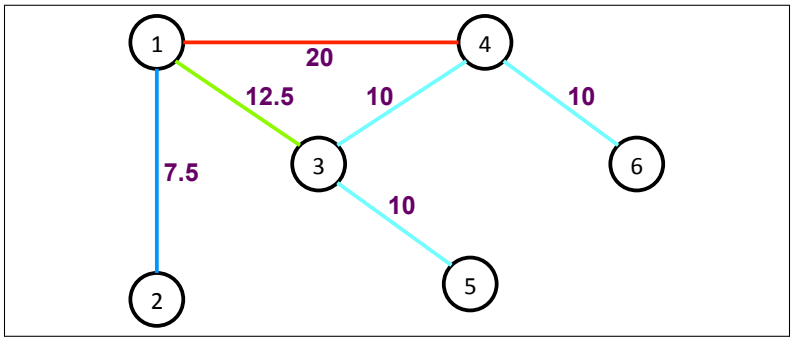


Figure 2-2: Symmetrized version of the weighted directed network shown in Figure 2-1.

The eigenvector centrality (i.e., the dominant eigenvector of $\tilde{A}$*) in this case is given by*

$$\mathbf{e} = \begin{bmatrix} 0.60 & 0.14 & 0.47 & 0.59 & 0.15 & 0.19 \end{bmatrix}^{\text{T}} \tag{2.9}$$

Comparing the values of $\mathbf{e}$ *and* $\mathbf{d}^{\text{tot}}$ *in this example, we see that they are not identical. For example, nodes 1 and 4 have the same total delay (80 units), but their eigenvector centralities differ, with node 4 being slightly more central. Similarly, nodes 5 and 6 have the same total delays, but different centralities. In both cases, we see that this is due to connectivity: Node 4 and node 1 are both similarly connected to each other and to node 3, but node 4 is more strongly connected to node 6 than node 1 is to node 2. Similarly, node 6 is more central than node 5 due to its connectivity to node 4 (which has high centrality) rather than to node 3 (which has lower centrality).*

One can also compare the hub ($\mathbf{h}$*) and authority scores (*$\mathbf{a}$*) of various nodes to their outbound*

(d^{out}) and inbound (d^{in}) delays, respectively. For this directed network,

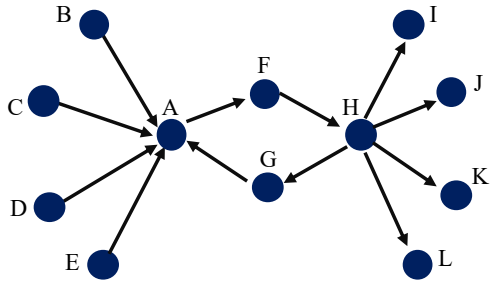
$$\mathbf{a} = \begin{bmatrix} 0.70 & 0.05 & 0.56 & 0.24 & 0.05 & 0.36 \end{bmatrix}^{\text{T}} \tag{2.10}$$

$$\mathbf{h} = \begin{bmatrix} 0.32 & 0.21 & 0.35 & 0.82 & 0.25 & 0.04 \end{bmatrix}^{\text{T}} \tag{2.11}$$

Although node 4 has a higher total inbound delay than node 6, it has a lower authority score since none of the three nodes it is connected to (1, 3 and 6) has a high hub score (and the weights on links (3,4) and (6,4) have low weights), while node 6 is strongly connected to node 4, which is a strong hub.

As the previous example suggests, the different features discussed above reflect different properties of the network. While the delay features ($\mathbf{d}^{\text{in}}$ and $\mathbf{d}^{\text{out}}$) reflect the total inbound and outbound delays at a node, the hub and authority scores reflect the connectivity and the propensity for delay to propagate into and out of a node in a network.

Example 2. *Another example of a network with its hub and authority scores is shown in Figure 2-3. Here, we can see that Nodes F and G are indistinguishable from the perspective of the in-degree an out-degrees. However, the hub and authority scores clearly distinguish between nodes F and G. Similarly, A and H are also clearly distinguished in terms of the hub and authority score.*



Node(s)	In-degree	Out-degree	Hub score	Authority score
A	4	2	0	0.71
B-E	0	1	0.32	0
F	1	1	0	0
G	1	1	0.32	0.32
H	2	4	0.71	0
I-L	1	0	0	0.32

Figure 2-3: Example of a weighted directed network, where the weights on the edges is the departure delay between the origin and the destination.

2.1.1 Clustering Nodes versus Clustering Graphs

We want to emphasize that the focus of this chapter is to cluster graphs themselves into similar groups. This is distinct from the more popular problem of clustering nodes in a graph. Practically,

the clustering of nodes in a graph corresponds to identify most similar airports, that is, once that tend to have similar delays (or propensity for delays) [31, 30, 51]. Here, we describe a simple community detection algorithm for clustering nodes in a graph.

As a first step, we consider the symmetrized version of the air traffic delay network, aggregated over a day. Community detection algorithms using the notion of *modularity*, as proposed by Girvan and Newman [112], determine groups of nodes (communities) such that there is stronger connectivity between nodes within a particular community than between nodes in different ones.

Given a partitioning of the network nodes into C communities, the modularity or quality of the partitioning is evaluated using the function, Q such that

$$Q = \sum_{i=1}^{C} (e_{ii} - a_i^2), \tag{2.12}$$

where e_{ij} is the fraction of total network edge weight that is on edges that connect nodes in community i to those in community j, and $a_i = \sum_j e_{ij}$. The term e_{ii} is the fraction of edge weight originating in cluster i and terminating in the same cluster. The partitioning is optimized over all values of C and possible community structures in order to maximize Q [112]. For example, community detection for the undirected graph shown in Figure 2-2 results in two communities, one containing nodes 1, 2, 4 and 6, and the other containing nodes 3 and 5 (Figure 2-4).

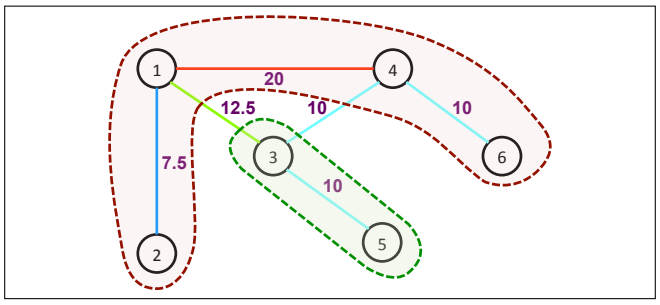


Figure 2-4: Community detection for the undirected graph shown in Figure 2-2.

2.2 Benchmarking framework and clustering methodology

Figure 2-5 shows a schematic of our benchmarking framework. The left side of the figure represents the computation of *baselines*, which are representative metrics. The *reference data set*

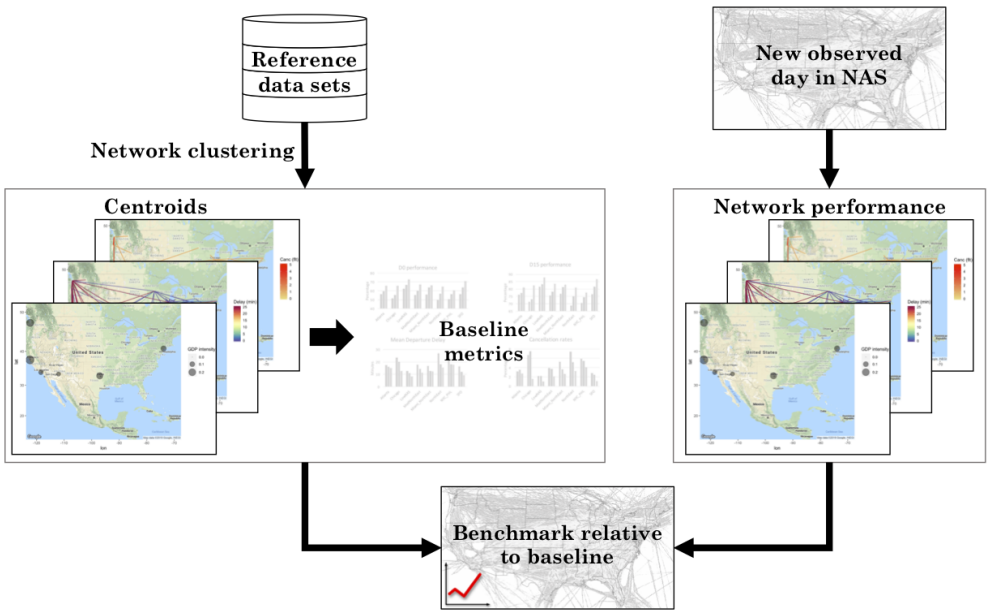


Figure 2-5: Flowchart representation of our benchmarking methodology framework.

contains information about all flights that operated within a certain time interval, e.g., a 2-year period. Clustering is used to categorize data points based on some parameter of interest; e.g., we cluster based on the GDP impacts in the system on that day. In this chapter, each data point represents a day, and we identify representative days. Days that belong to a certain cluster will be used to compute baseline statistics for that cluster. This means that we can define any metric of importance, say mean departure delays, or the number of canceled flights, or average diversions from Atlanta, or even be as specific as cancellations on the Boston-San Francisco route for United Airlines. In other words, we can define any metric of importance, and compute baselines for each of the representative groups. On the right-hand side of Figure 2-5, we describe how we would perform *benchmarking* for a new day. *Benchmarking* is the process by which we compare the new day, with baselines corresponding to an appropriate cluster (or representative group) that the day maps to. Depending on the metrics chosen during the clustering and baseline computation, the benchmarking can be airline-, airport-, or route-dependent.

2.2.1 Clustering methodology

We approach the identification of similar GDP, delay, and cancellation networks for a given day as a classical clustering problem. The k-means algorithm [113] is a widely-used approach in unsupervised learning that splits the data points into k disjoint clusters such that the within-cluster sum-of-squares, i.e. the within-cluster error, is minimized. Our primary motivation for choosing k-means clustering is its simplicity, and the ability to directly control the number of clusters. Other clustering approaches are not as well-suited for our application: e.g., DBSCAN requires a hyperparameter that sets the minimum number of points in a cluster [114], k-medoids was found to give similar results as k-means clustering [93], spectral clustering is not applicable for a time-series of weighted directed graphs [115], and hierarchical clustering methods identify relations between clusters [116], a different objective from ours.

The k-means algorithm considers two data points to be similar if the 2-norm of the difference between their corresponding feature vectors is small, and proceeds to group these two data points within the same cluster. The mean of all data points that belong to a cluster, called the *centroid*, is representative of the entire cluster that it belongs to.

Feature selection, or the choice of vectors that best characterize the similarities and differences between groups of data points – in our case, capacity constraints, delays, and cancellations within the air transportation network – is a critical aspect of clustering. We differentiate between the network-centric feature vectors we construct for airport capacity reductions versus delays and cancellations. Specifically, GDP intensities are represented as node weights, whereas origin-destination delays and cancellations are represented as edge weights in our network. Next, we describe the construction of these feature vectors.

2.2.2 Features for clustering by GDP impacts

A GDP limits the rate at which aircraft can arrive at the impacted airport during a specified period of time, by delaying flights at their departure airports. Several factors can lead to a demand-capacity imbalance at an airport, potentially resulting in a GDP; causes include inclement weather events such as low ceilings, reduced visibility, and convective activities, construction activities, high demand volume, equipment outages, and even security and safety incidents. Each GDP is-

suance at an airport specifies the duration and a reduced capacity profile for that time interval.

In our benchmarking framework, we quantify the severity of the airport capacity reduction as the *intensity* of the GDP. For example, consider San Francisco International Airport (SFO), where the nominal airport arrival rate (AAR) is 60 aircraft per hour. At 2 pm, weather forecasts indicate thunderstorms in the region from 4 to 7 pm. As a result of this potential capacity reduction, a GDP is issued, reducing the AAR at SFO to 20, 30, and 45 aircraft per hour from 4 to 7 pm, respectively. Our proposed GDP intensity metric is defined for each hour as:

$$\text{GDP intensity} = \frac{\text{Nominal rate} - \text{Reduced rate}}{\text{Nominal rate}}. \tag{2.13}$$

In the SFO example, the GDP intensities for the hours starting at 4 pm, 5 pm, and 6 pm are $\frac{60-20}{60} = 0.67$, $\frac{60-30}{60} = 0.5$, and $\frac{60-45}{60} = 0.25$, respectively. By computing the fractional reduction in capacity rather than the absolute value, the metric allows for a fair comparison of GDP impacts at different airports, accounting for airports with differing nominal AARs. A capacity reduction of 5 aircraft per hour will result in a higher GDP intensity at an airport with a nominal AAR of 30 aircraft per hour (the GDP intensity would be $(30-25)/30 \approx 0.17$), than at an airport with a nominal AAR of 80 aircraft per hour (the GDP intensity in this case would be $(80-75)/80 \approx 0.06$). The GDP intensity, as we have defined it, is well-defined at the limits: when no GDP is issued, the GDP intensity is 0; on the other hand, the most severe GDP, with an assigned AAR of zero, has an intensity of 1.

Airports that are geographically proximate to each other oftentimes experience similar weather conditions. However, varying traffic demand at these proximate airports may result in GDPs being issued at one airport but not at the other. Furthermore, even when two proximate airports experience a GDP of similar intensity, their network impacts may be completely different depending on the network connectivity of the airlines servicing those airports. For example, the three major commercial airports in New York (EWR, LGA, and JFK) frequently illustrate these network effects. Hence, we consider the GDP impacts at each airport independently from GDP impacts at other airports when constructing the feature vector for a given day. Every day of operation has an associated time series of GDP intensities for each airport in the NAS. These values are stacked together to form a feature vector that describes the capacity reduction at all the airports across each

time step during that day. For each day, the feature vector is a $(24 \times N)$-dimensional vector of real numbers in the interval $[0,1]$, where N is the number of airports in the network. Specifically, let $\mathbf{g}_{d,h} \in \mathbb{R}^{N\times 1}$ be a vector of the GDP intensities at all the airports on day d at hour h. Then, to cluster based on airport capacity impacts, we construct the feature vector for day d as:

$$\mathbf{f}_d^{\text{GDP}} = \begin{pmatrix} \mathbf{g}_{d,1} \\ \vdots \\ \mathbf{g}_{d,24} \end{pmatrix} \in \mathbb{R}^{24N\times 1}. \tag{2.14}$$

2.2.3 Features for clustering delay and cancellation networks

We now detail the construction of feature vectors to perform network-centric clustering of delay and cancellation networks. We utilize a network representation of delays for constructing the feature vectors. A delay network is defined for each hour h of the day d, with nodes corresponding to airports, and the weight on each directed edge corresponding to the origin-destination delay. The edge weight for a network, a_{ij}, is the median departure delay of flights traveling from the origin airport i to the destination airport j within hour h. We collect all such a_{ij} weights into the adjacency matrix $A = [a_{ij}]$ for the delay network. Since the delays change with time, the edge weights of the delay network change as well. Each day d is represented by a time series of 24 weighted and directed networks $(G_{d,1}, G_{d,2}, \ldots, G_{d,24})$, where each of the delay network graphs $G_{d,h}$ has its unique adjacency matrix $A_{d,h}$. See Figure 2-6 for a simple example.

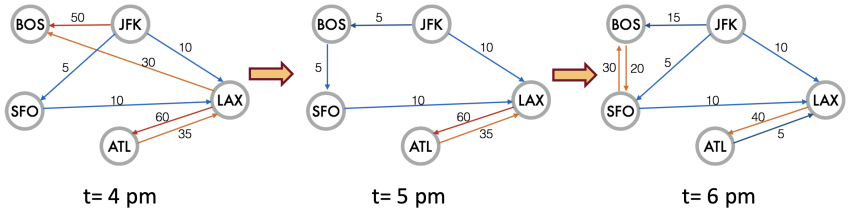


Figure 2-6: Example of a time series of weighted-directed graphs. In this example, nodes represent airports, and edge weights represent the median delay of flights travelling on that route.

We then construct a feature vector $\mathbf{f}_d$ of network delays for a given day of operations d. One approach is to stack all the edge weights sequentially for each of the 24 networks that correspond to a day [101]. This approach has two limitations. First, stacking all the edge weights completely

disregards the network structure, ignoring any possible airport-to-airport interactions embodied by the existence of an edge within the graph. This renders the feature vectors as well as the identified clusters to not be truly representative of the system-wide network impacts that we wish to capture. Second, the feature vector may be very high dimensional (N^2, for a graph with N nodes), causing the clustering algorithms to be computationally expensive. By contrast, our approach of using *weighted hub and authority scores* to incorporate network effects, and is scalable.

We denote $\mathbf{h}_i$ and $\mathbf{a}_i$ as the hub and authority score vectors, respectively for graph $G_{d,i}$ on a day d. We would like to construct $\begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix}$ as a feature vector for each delay networks, thereby incorporating network effects in a compact representation using only $2N$ entries. However, since $\mathbf{h}_i$ and $\mathbf{a}_i$ are arbitrarily normalized eigenvectors, two networks whose weights differ by a scalar constant would have the same feature vector. In other words, delay networks during high congestion periods and periods of low traffic movements (e.g. late at night) could have the same feature vector if the relative proportion between the flight delays on different routes is the same. To address this issue, we use a *weighted* hub and authority feature vector, where the total system delay of each hour is used to re-scale $\mathbf{h}_i$ and $\mathbf{a}_i$.

Thus, for a day d defined by $(G_{d,1}, G_{d,2}, \ldots, G_{d,24})$, the feature vector f_d^{delay} is given by:

$$\mathbf{f}_d^{\text{delay}} = \begin{bmatrix} \alpha_1 \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{a}_1 \end{pmatrix} \\ \vdots \\ \alpha_{24} \begin{pmatrix} \mathbf{h}_{24} \\ \mathbf{a}_{24} \end{pmatrix} \end{bmatrix}, \tag{2.15}$$

where $\alpha_h = \sum_{ij} a_{ij}^h$ is the total delay at the h^{th} hour and $A^h = [a_{ij}^h]$ is the corresponding adjacency matrix.

We define cancellation networks in a similar manner as delay networks, with the only difference being that the edge weights are the number of canceled flights. The number of flight cancellations is typically small compared to the number of total scheduled flights. Furthermore, we note that the reasons for cancellations range from operational (e.g. aircraft maintenance issues, crew scheduling mishaps) to system-wide irregular operations resulting in excessive delays. Not all cancellations

are reflective of system-wide capacity imbalances, and the small number of cancellations every hour would result in undesired sensitivity of the clusters to exogenous factors. For these reasons, to achieve a more robust clustering of cancellation networks, we ignore hourly variations and construct only one cancellation network to represent an entire day of operations. Thus, we construct the cancellation feature vector for each day d as:

$$\mathbf{f}_d^{\text{canc}} = \sum_{ij} a_{ij} \times \begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix}. \tag{2.16}$$

2.2.4 Discussions

The NAS impacts in terms of GDPs, delays, and cancellations for a given day of operations d are reflected in the feature vectors $\mathbf{f}_d^{\text{GDP}}$, $\mathbf{f}_d^{\text{delay}}$ and $\mathbf{f}_d^{\text{canc}}$, respectively. We use these feature vectors as inputs into the k-means clustering to identify days with similar GDP, delay, and cancellation networks. Consequently, each day used for the training would have three labels associated with it: One describing the type of capacity impacts in terms of system-wide GDP issuance; one describing the propagation of delays within the network; and one for the distribution of flight cancellations within the network. Any new day in a testing set can now be mapped to the closest cluster centroid with regards to its GDP, delay, and cancellation networks.

The clusters of pertinent GDP impacts that we identify are primarily used as performance baselines for our benchmarking framework, particularly from the perspective of airlines. Since the effects of issued GDPs are targeted towards airports, this sets up the system constraints within which the airlines operate. We present case studies and discussions of using our benchmarking framework from both the airline and the system operator (e.g., the Federal Aviation Administration or FAA) perspective in Sections 2.5 and 2.6, respectively.

The reference data sets that are used to generate clusters and baseline metrics can correspond to any time period, or could be restricted to an airline. They may also be scoped to include only flights operating through a particular subset of airports (e.g., only hubs). In practice, all baselines could be re-computed periodically to include more recent data; such re-computations can be performed by appending new days of operations to the reference data sets. Furthermore, one could examine hourly (as opposed to daily) clusters of GDP, delay, and cancellation networks as well.

We reiterate that our focus is not the clustering of nodes within a graph, i.e., we are not performing node-based community detection; we are instead taking entire graphs to be the object that we are trying to cluster with other graphs based on their topology (e.g., connectivity patterns, edge weights, etc.) Furthermore, different airspace monitoring applications may require different performance measures and feature vectors (e.g., [101, 89]).

2.3 Identifying Clusters from Data

In this section, we apply k-means clustering algorithm using the feature vectors described in Section 2.2 to identify representative, or characteristic type-of-days.

2.3.1 Description of data sets

Data on airport capacity impacts was obtained from the FAA Advisory Database, which publishes information on the reduction in airport capacity due to GDPs [117]. We merge multiple notifications for the same GDP, including revisions, ground stops, extensions, and early cancellations, to obtain an hourly time series of the reduced capacity at a specific airport. We restricted our analysis to the FAA's Core 30 airports. We obtained nominal capacities from the FAA Airport Capacity Profiles report to compute fractional reductions in AAR [118]. GDP data used in this analysis was available for the period spanning 2014 through 2018. Flight schedules, delays, and cancellation information were obtained for 2014-2016 from public data sources such as the FAA ASPM [103], the Department of Transportation's BTS database [1], and other third-party providers.

The data set included 20,012 GDP advisory notifications specified for the FAA Core 30 airports, as well as records of 22.9 million domestic US flights operated by 55 airlines, through 426 airports, and on 6,600 unique origin-destination (OD) pairs. Approximately 500,000 (2.1% of the total) of the scheduled flights were canceled. We removed 5,000 diverted flights (0.2% of the total) from our baseline calculations and subsequent benchmarking. To improve robustness and eliminate outliers, the clustering was restricted to the top 1,000 OD pairs based on the average traffic. All of these OD pairs had at least 6 flights per day (defined as between 0900Z and 0859Z) on average per day. Data from 2014-2015 was used as the reference.

2.3.2 Selecting the number of clusters

An input parameter for k-means clustering is the number of clusters k that the user wants the data observations to be partitioned into. The choice of this k parameter depends on the application, and can be evaluated using several criteria [119]. We use the within-cluster sum of squares (WCSS), the silhouette score, and the Davies-Bouldin index for this purpose, and the values of these metrics as a function of k is shown in Figure 2-7. However, we note that the data does not naturally form clusters, and there is no clear choice for k. Nevertheless, the underlying delay data is non-uniform [120], and clustering is still useful to identify interpretable and representative baselines.

The choice of k is now guided by two competing factors: lower k improves interpretability, while higher k captures more variability and is more accurate. We select values of k such that we achieve a balance between the number of clusters and the within-cluster population, while taking into account input from airline subject matter experts. For our analysis, we chose $k = 8$, 6, and 5, for the number of GDP, delay, and cancellation type-of-day clusters, respectively.

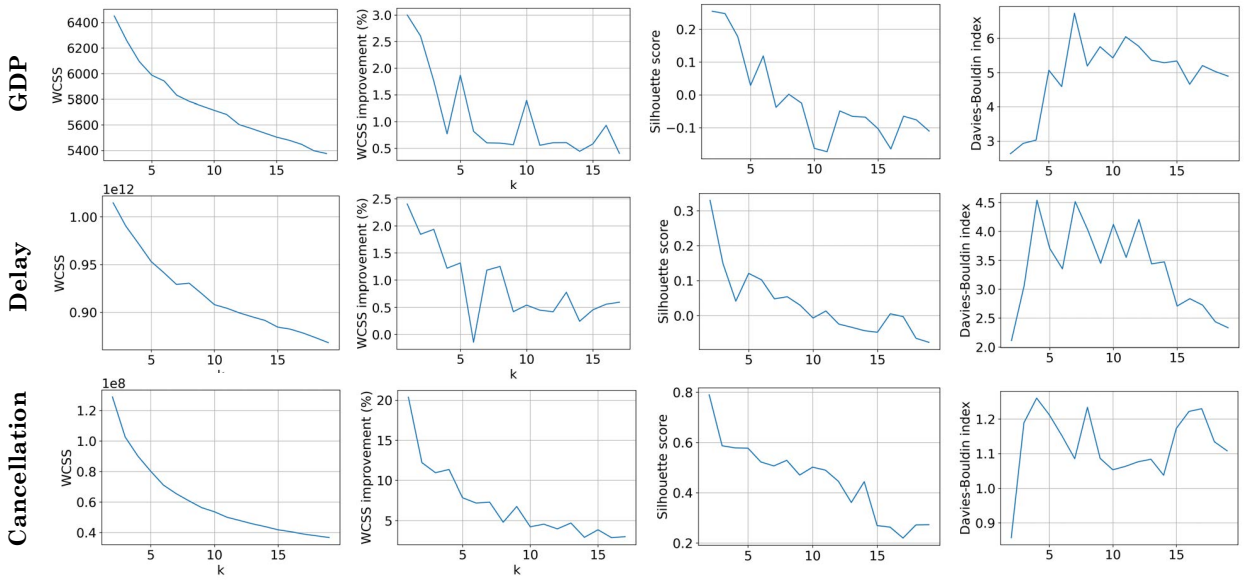


Figure 2-7: Four measures to guide the selection of k (WCSS, change in WCSS, the silhouette value, and the Davies-Bouldin index, all as a function of k), for k-means clustering of GDP, delay, and cancellation networks.

2.3.3 GDP type-of-day

Eight GDP type-of-days are identified through k-means clustering using the feature vector $\mathbf{f}^{\text{GDP}}$. Qualitative descriptions and frequencies of occurrence of these clusters are presented in Table 2.1. We visualize the centroid, or the representative data point, of each GDP type-of-day cluster in Figure 2-8.

Day type	**Qualitative description**	**Frequency (2014-15)**
Atlanta (`ATL`)	High GDP intensity at ATL; moderate to low GDP activity elsewhere.	6.3%
Chicago (`CHI`)	Very high GDP intensity at ORD and MDW; moderate to low GDP activity elsewhere.	7.8%
Medium Northeast (`MedNE`)	Medium-high GDP intensities in the Northeast (NYC airports, PHL and BOS). Moderate to low GDP activity elsewhere.	15.3%
Low NAS (`LowNAS`)	Low GDP intensities nationwide.	48.1%
Miami, Northeast (`Miami_NE`)	High GDP intensity in MIA, and medium-high GDP intensities in the NYC/PHL area.	5.1%
Northeast (`NE`)	Very high GDP intensities in the NYC, PHL, and Washington DC airports; medium-high at BOS and Chicago. Moderate to low GDPs elsewhere.	2.2%
New York City, Philadelphia (`NYC_PHL`)	Very high GDP intensities at the NYC airports (LGA, JFK and EWR) and PHL; medium-high GDP activity at BOS. Moderate to low GDPs elsewhere.	5.1%
San Francisco (`SFO`)	High GDP intensity in SFO; moderate to low GDP activity elsewhere.	10.1%

Table 2.1: Characteristic GDP type-of-day identified through clustering.

Each centroid in Figure 2-8 for the GDP type-of-days is visualized as the average of the corresponding time-series of GDP intensities at the FAA Core 30 airports; the full time-series is a sequence of 24 node-weighted networks. A larger circle indicates a higher average GDP impact at the airport. For ease of visualization, we use one image to represent the entire time series, although the clustering considers temporal variations. We note that peak GDP intensities and impacts may occur at different times for different centroids.

As expected, `LowNAS` GDP type-of-days occur most frequently: 48% of days in 2014-2015 are classified as having low GDP intensities throughout the NAS. However, the GDP types affecting the Northeast, i.e. `NE`, `MedNE`, `Miami_NE`, and `NYC_PHL`, together account for a significant fraction of days (28%), and have higher GDP intensities and impacts. We note the multitude of distinct GDP type-of-days that impact the Northeast and Mid-Atlantic region of the US, reflecting both the frequency and diversity of GDP activity at airports in the East Coast.

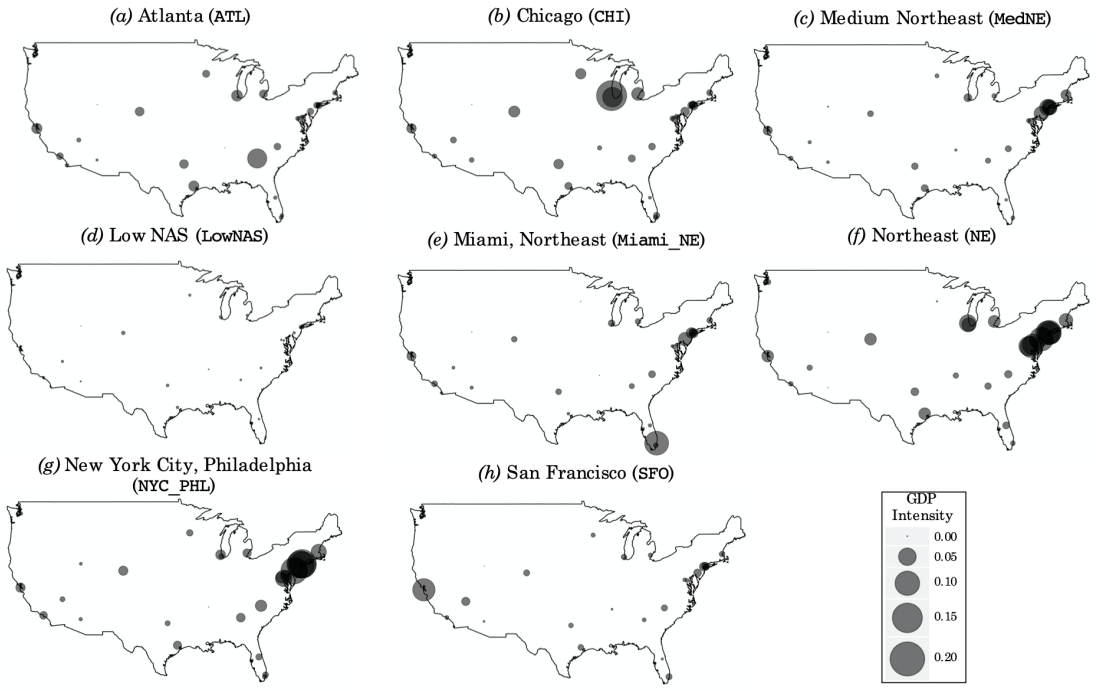


Figure 2-8: Visualization of the centroids for all GDP type-of-days. The size of the circle reflects the GDP intensity at an airport.

2.3.4 Delay type-of-day

We present the six different delay type-of-days identified via network-centric clustering. We list each delay type-of-day, along with its qualitative description and the frequency of occurrence, in Table 2.2. We visualize the centroid for each cluster (Figure 2-9) as a network, where the edge weight is the median OD pair delay for all flights on that OD pair during the days of operations belonging to that particular cluster. We note that even though the edges are directed, for ease of visualization, the edges are depicted as undirected and weighted by the maximum of the two directions. Although our centroid visualizations show a single delay network, recall that it is actually representative of a 24-hour time series, reflecting the temporal evolution of delays. Analogous to the case with the GDP centroid visualizations, to maintain simplicity and visual interpretability, we project out the temporal dimensions, even though they are still relevant in the actual clustering process.

Similar to the case with GDPs, the most prevalent type-of-day for delays is `LowNAS`, with average departure delays of 9 minutes, and with over 85% of flights arriving within fifteen minutes of

Day type	Qualitative description	Frequency (2014-15)
Atlanta (ATL)	Delays centered around ATL, with a mean departure delay of 19 min. 71% of flights arrive within 15 min of their scheduled arrival times. Largest avg. arrival delays: ATL, MEM, ORD, IAD, EWR.	3.6%
Chicago (CHI)	Delays centered around Chicago, with a mean departure delay of 17 min. 74% of flights arrive within 15 min of their scheduled arrival times. Largest avg. arrival delays: ORD, MDW, MEM, EWR, IAD.	8.1%
High NAS (HighNAS)	Delays widespread and high, with a mean departure delay of 29 min. Only 61% of flights arrive within 15 min of their scheduled arrival times. Largest avg. arrival delays: EWR, DFW, IAH, LGA, ORD.	2.9%
Low NAS (LowNAS)	Delays low nationwide, with a mean departure delay of only 9 min. Over 85% of flights arrive within 15 min of their scheduled arrival times. Largest avg. arrival delays: SFO, EWR, JFK, ORD, IAD.	49.3%
Northeast (NE)	Delays centered in the Northeast, with a mean departure delay of 18 min. 72% of flights arrive within 15 min of their scheduled arrival times. Largest avg. arrival delays: LGA, EWR, JFK, PHL, BOS.	11.0%
West Coast, Medium NAS (WC_MedNAS)	Delays centered in the West coast & moderate elsewhere; mean departure delay of 14 min. 77% of flights arrive within 15 min of their scheduled arrival times. Largest avg. arrival delays: SFO, DFW, DEN, LAX, LAS.	25.2%

Table 2.2: Characteristic delay type-of-day identified through clustering.

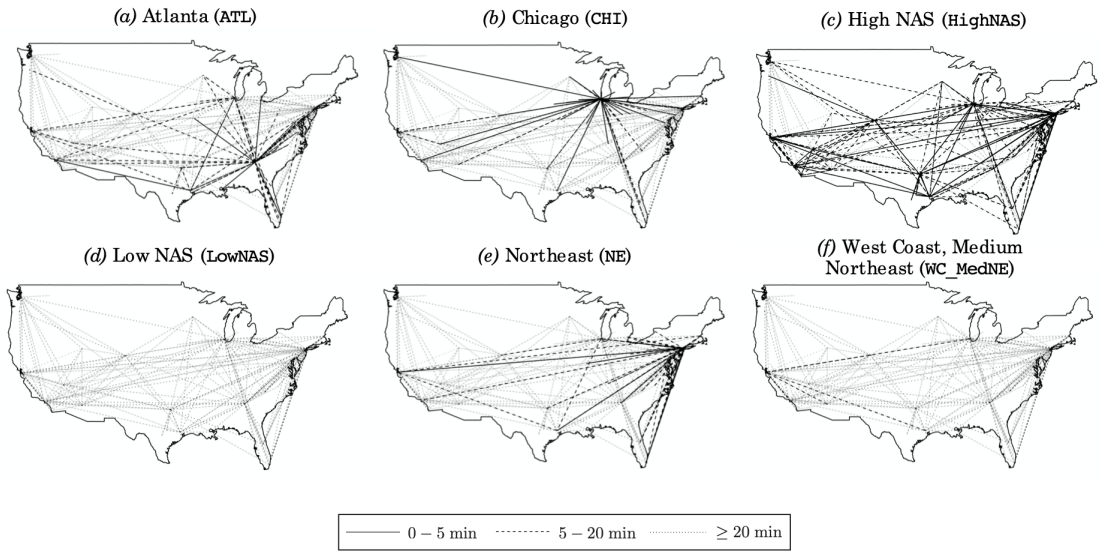


Figure 2-9: Visualization of the centroids for all delay type-of-days.

their scheduled arrival times (i.e. the A15 percentage metric). We can also isolate the airports with the largest arrival delays for each delay type-of-day to identify airports that are typically delayed when a particular delay type-of-day occurs. Certain airport nodes begin to stand out in terms of their participation in many delay type-of-days. For example, Chicago O'Hare International Airport (ORD), a major dual hub that also shares its terminal airspace with the proximate airport Chicago-

Midway, has significant delays in ATL, CHI, and most prominently, HighNAS delay type-of-days.

2.3.5 Cancellation type-of-day

To complete our presentation and analysis of resultant cluster centroids, we qualitatively discuss the five type-of-days for cancellations in Table 2.3, and present the visualizations of the centroids in Figure 2-10. We constructed these simplified visualizations analogous to those for delay type-of-days, with the key difference being that the edge weights now correspond to the number of cancellations observed on an OD pair.

Day type	**Qualitative description**	**Frequency (2014-15)**
Atlanta (ATL)	Very heavy cancellations on routes to/from ATL (75% of all scheduled arrivals at ATL canceled). Systemwide cancellation rate of 18%.	0.4%
Chicago, Northeast (CHI_NE)	Heavy cancellations in Chicago and parts of the Northeast (37% of all scheduled arrivals at LGA and ORD, 35% of scheduled arrivals at EWR, and 32% of scheduled arrivals at MDW, canceled). Systemwide cancellation rate of 14%.	2.3%
Medium Chicago, Medium Northeast (Med_CHI_NE)	Moderate cancellations of flights to/from Chicago and the Northeast (14% of all scheduled arrivals at LGA, 13% at EWR, 12% at ORD, and 10% of scheduled arrivals at PHL, canceled). Systemwide cancellation rate of 6%.	13.2%
Low NAS (LowNAS)	Most frequently-occurring day, with a systemwide 1.3% cancellation rate. EWR has the largest (2.5%) rate of canceled arrivals.	83.5%
High Northeast (HighNE)	Very high cancellations of flights to/from the Northeast (83% of all scheduled arrivals at LGA, 79% at EWR, 66% at PHL, 64% at BOS, 56% at DCA, and 51% of scheduled arrivals at IAD and JFK, canceled). Systemwide cancellation rate of 25%.	0.5%

Table 2.3: Cancellation type-of-day centroids

Due to their operational rarity, cancellation type-of-days are more difficult to distinguish compared to delay and GDP type-of-days. This rarity is exemplified in the fact that 83.5% of operational days within the 2014 to 2015 time frame are classified as LowNAS cancellation type-of-day, with an overall system-wide cancellation rate of 1.3%. Interestingly, in the LowNAS cluster, EWR had the highest rate of canceled arrivals, at 2.5%. Some of the cancellation clusters match those seen in delay and GDP type-of-days in terms of geographic distribution, centering on major hub airports in Atlanta, Chicago, and the Northeastern region of the US.

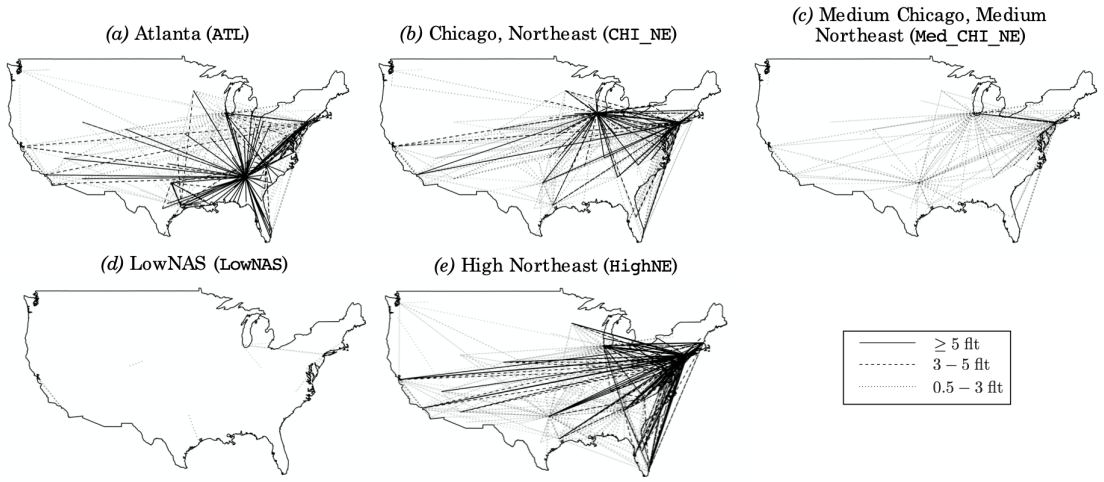


Figure 2-10: Visualization of the centroid for all cancellation type-of-days.

2.4 Types-of-Day as Features for Delay Prediction

The type-of-day classification of a day results in three categorical labels for each day that concisely represent the complex, time-dependent network dynamics that governs the spatiotemporal patterns of GDPs, delays and cancellations. It is therefore reasonable to expect that these labels can be used as features to predict delays (on individual OD pairs, for example) a few hours in advance.

We consider the problem of predicting, 2 hours in advance, whether or not the median delay experienced by flights on a given OD pair will exceed 60 min. Prior work suggests that random forest classifiers are a good approach to this prediction problem [101]. To evaluate the benefit of including GDP, delay and cancellation type-of-day labels as features, we consider two variants of the random forest classifier: One that considers only the current delay levels on that OD pair and the time-of-day as features (predictors), and another that considers the three type-of-day labels *in addition to* the current delay levels on that OD pair and the time-of-day.

In other words, the two feature vectors ($\mathbf{x}_1$ and $\mathbf{x}_2$) considered for the random forest classifier are:

$$\begin{aligned}
\mathbf{x}_1 &= [d^{\mathrm{OD}}(t),\, t]^{\mathrm{T}}, \\
\mathbf{x}_2 &= \Big[d^{\mathrm{OD}}(t),\, t,\, \text{Delay type-of-day},\, \text{GDP type-of-day},\, \text{Canc. type-of-day}\Big]^{\mathrm{T}}
\end{aligned}$$

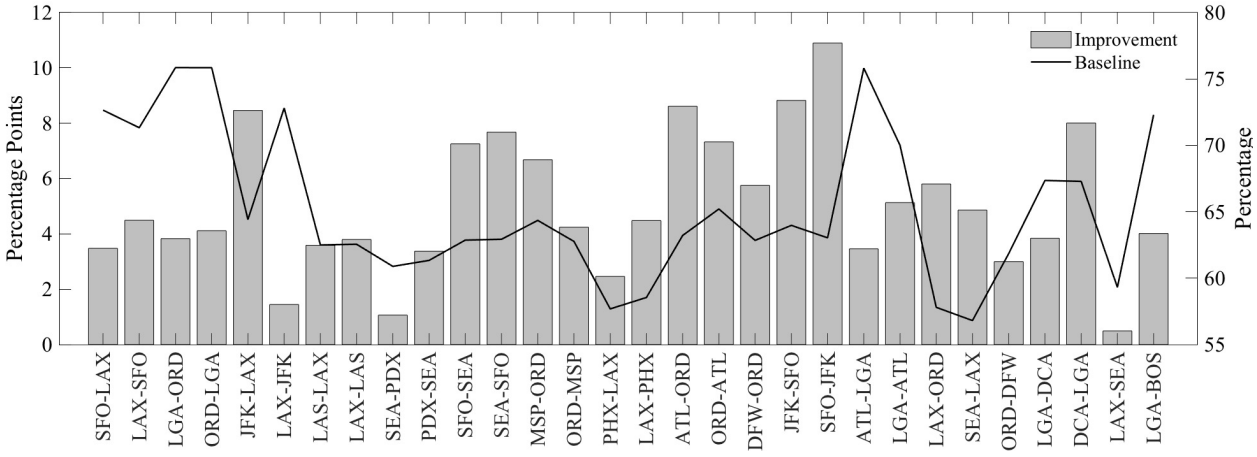


Figure 2-11: Improvement in prediction accuracy from adding type-of-day labels as features, for the test dataset. The baseline (solid black line) corresponds to the accuracy of the random forest classifier that does not include the type-of-day labels.

where $t \in \{1,...,24\}$ is the hour of the day, and $d^{\mathrm{OD}}(t)$ is the median delay of flights departing from origin O towards destination D during the t^{th} hour of a day. The output of the random forest classifier is a binary variable $\mathbf{y}$ that predicts whether (or not) $d^{\mathrm{OD}}(t+2)$ will exceed 60 min. We restrict the datasets to the period 9 AM to midnight Eastern Time, where we expect traffic to be nonzero on the OD pair. A random forest classifier is trained for each of the top 30 OD pairs (based on traffic volume in mainland US) in 2014-2015. Since the datasets are imbalanced (that is, the number of data points with delay level exceeding 60 min is much lower than the number with delay less than 60 min), we randomly sample low ($<$ 60 min) delay points to get balanced training and test datasets for each OD pair. The balancing and model training steps are repeated 100 times. It is worth noting that the type-of-day used in feature $\mathbf{x}_2$ is only known with certainty after the entire day is over; however prior work has shown that it can be predicted with high accuracy by 12 pm Eastern Time [121].

Figure 2-11 shows the improvement in prediction accuracy (measured as the percentage of correctly predicted data points) on the test dataset, when the type-of-day labels are included as a feature. The results are promising, and show 0.5–10.9 percentage point improvement depending on the OD pair, with an average value of 5.0 percentage points. The average improvement in accuracy over the baseline (for the top 30 OD pairs) from including the type-of-day features is 7.8%.

The variability by OD pair, both in the baseline performance and in the improvement in accuracy, is interesting. We see that some OD pairs (notably, LGA-ORD, ORD-LGA and ATL-LGA)

have reasonably good baseline accuracy, suggesting that just the current delays on that OD pair and the time-of-day are reasonably good predictors of future delay levels. By contrast, other OD pairs (for example, JFK-LAX, ATL-ORD and SFO-JFK) see considerable improvements from including the type-of-day labels, with increases in prediction accuracy of as much as 17%.

2.5 Airline performance benchmarking

We present an on-time performance benchmarking application from the perspective of an airline. This use case was based on the implementation of these methods at a major US airline for post-hoc analysis of on-time performance. As illustrated in Figure 2-12, an airline can select from a variety of on-time performance metrics such as average departure and arrival delays, number of cancellations, D0/D15/A0/A14 percentages, block time adherence, and ground turn times, each at the airline-, airport-, or system-level. We consider the average departure delay (henceforth simply referred to as *delays*) per flight and the cancellation percentage for illustrative purposes.

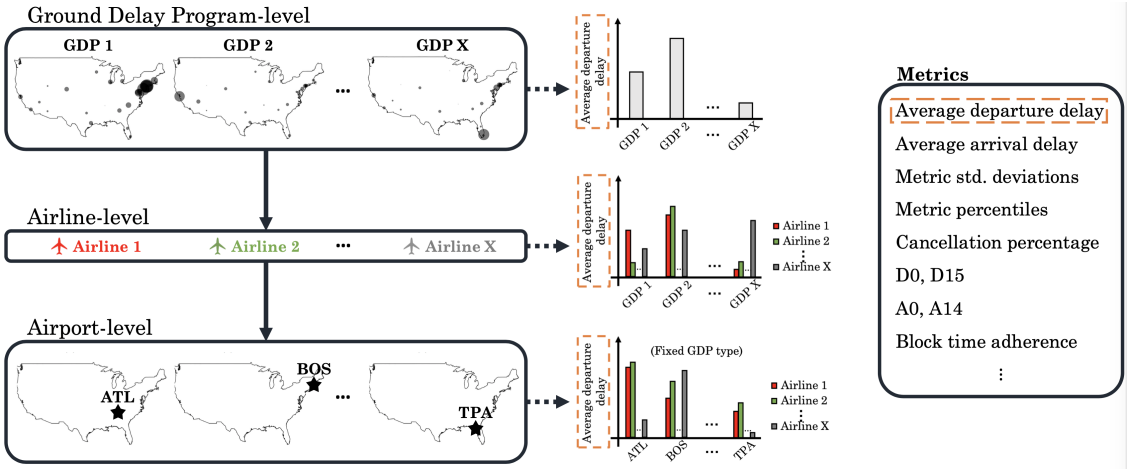


Figure 2-12: Performance baselines can be customized by airline, airport, or system-wide.Mainline and regional carrier breakdowns are not shown here for simplicity, but are considered in Section 2.5.3.

2.5.1 Confidence intervals for performance metrics

Let $\{x_d^{(i)}\}_{i=1}^{i=n(\mathscr{P})}$ represent the set of observations of performance metric d. Every observation $x_d^{(i)} \in \mathbb{R}$ is recorded by aircraft $i = 1, \ldots, n(\mathscr{P})$, where $n(\mathscr{P})$ is the $\mathscr{P}$-conditioned sample population.

$\mathscr{P}$ could represent any particular combination of airline, mainline or regional, airports, and cluster identity. We denote the sample mean and standard deviation of $\{x_d^{(i)}\}_{i=1}^{i=n(\mathscr{P})}$ by $\bar{x}_d^{n(\mathscr{P})}$ and $s_d^{n(\mathscr{P})}$, respectively. The two-sided confidence interval (CI) of level α is denoted by

$$CI\,(d, n(\mathscr{P})) = \bar{x}_d^{n(\mathscr{P})} \pm z_{\alpha/2} \frac{s_d^{n(\mathscr{P})}}{\sqrt{n(\mathscr{P})}} \tag{2.17}$$

For $\alpha = 0.01$, $z_{\alpha/2} \approx 2.576$. We only compute the CI for delays, and not for cancellations. The percentage of cancellations is a *discrete* performance metric with a single value (and no CI), since a flight is either canceled or it is not.

The CI can be computed for the mean performance of the baselines, as well as for any particular day. When benchmarking an individual day (Section 2.5.3), we can assert that the average delay on an individual day deviates in a significant way from the mean cluster delay if the two CIs do not overlap. Lastly, we note that (2.17) is only meaningful when $n(\mathscr{P})$ is large; for our data, we find that this is a reasonable assumption.

2.5.2 On-time performance baselines

Figure 2-13 shows the baseline metrics computed using the 2014-15 data set. The baseline metrics are evaluated as an average over all flights on days classified as belonging to a particular GDP type. From the system-wide baseline values (Figure 2-13*(a)*), we see that `NE` and `NYC_PHL` GDP type-of-days result in the highest departure delays and cancellations, while `LowNAS` experiences the lowest delays and cancellations.

A1, A2, and A3 correspond to three major US airlines. We observe from Figure 2-13*(b)* that the baseline delays and cancellations for airline A2 are lower than those for A1 and A3, across all GDP types-of-day. Furthermore, the GDP type-of-days that most impact system performance (Figure 2-13(a)) may not necessarily be the most severe for a particular airline. For example, `NE` GDP types-of-day affect the performance of A2 and A3 less severely than A1. Not surprisingly, airlines differ in their baseline performance even in the absence of airport capacity reductions (`LowNAS` GDP type-of-day), due to differences in operational practice.

Although baseline performance differs by airline for most GDP types-of-day, the `ATL` GDP type results in similar impacts for all three major airlines, even though it is the prominent hub

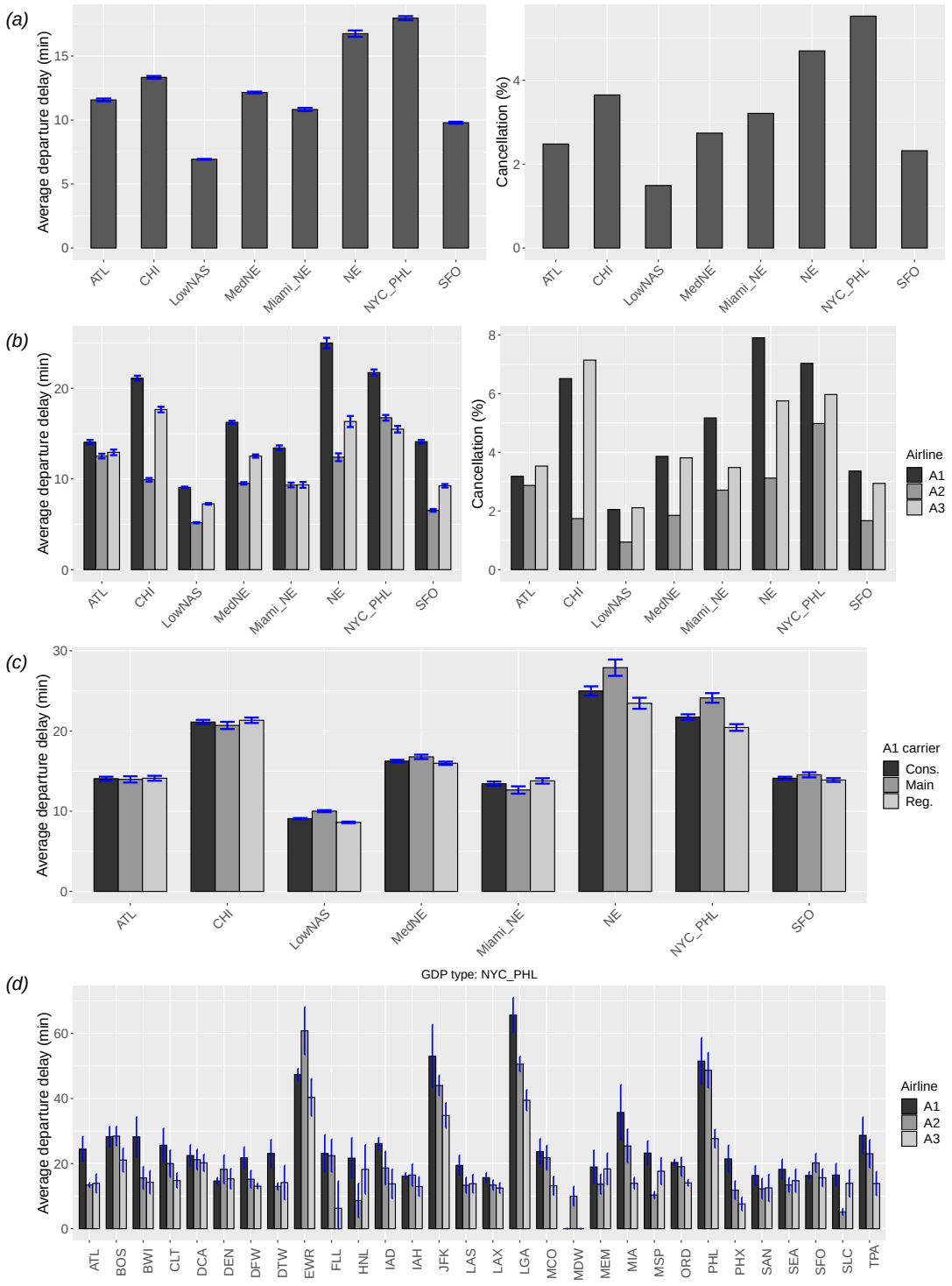


Figure 2-13: Performance baseline plots generated with average departure delays in minutes; baseline performance across *(a)* GDP types; *(b)* three major US airlines for each GDP type; *(c)* consolidated fleet, mainline, and regional carriers of airline A1 for each GDP type; *(d)* three major US airlines at the top 30 US airports, given a specific GDP type. The error bars indicate a 99% two-sided confidence interval (see Section 2.5.1 for details)

airport of one of them.

We analyze the mainline and regional carrier performance for airline A1 in Figure 2-13*(c)*, and consider specific airports in Figure 2-13*(d)*. Although all airlines have high delays at EWR, JFK, LGA, and PHL during NYC_PHL GDP type-of-days, airline-specific impacts are also evident: the delays are the highest for A1 at LGA, and A2 at EWR. The propagation effects of GDPs also vary by airline. For example, by controlling for the NYC_PHL GDP type-of-day (Figure 2-13*(d)*), the propagation of departure delays for airline A1 from the NYC-PHL airports to ATL becomes much more apparent.

2.5.3 Benchmarking case study from the airline perspective

We present two case studies of benchmarking operational performance from an airline's perspective. For the first case, we consider two days, July 1, 2016, and February 21, 2015, which were both classified as being NYC_PHL GDP type-of-days. We then benchmark the on-time performance of the three airlines A1, A2, and A3 with respect to the appropriate baselines for that GDP type-of-day. The left-hand panel of Figure 2-14 summarizes July 1, 2016, a day when three distinct convective systems moved through the western, Midwestern, and East Coast regions of the US. The weather patterns in the West and Midwest persisted through the entire day, whereas the East Coast system moved offshore at around 1600Z. These weather systems resulted in intense GDPs in the Mid-Atlantic region, and less intense GDP activity in the Los Angeles area. By contrast, the weather patterns on February 21, 2015 (Figure 2-14, right panel) consisted primarily of a large nor'easter-type storm that moved steadily from west to east, persisting through the entire day, resulting in GDPs in the Northeast. Our clustering framework helps compare these two complex weather phenomenon and their aviation impacts precisely: if the two days belong to the same GDP cluster, we can assume the weather impacts to be similar. However, the resultant delay networks for the two days are significantly different: July 1, 2016, was a NE delay type-of-day, while February 21, 2015, was a LowNAS delay type-of-day.

The average delay (and associated CI) for all airlines combined was lower on February 21, 2015, than the NYC_PHL GDP type-of-day baseline values. This trend was reversed for July 1, 2016. On July 1, 2016, A2 and A3 performed poorly compared to their baselines, whereas A1 saw

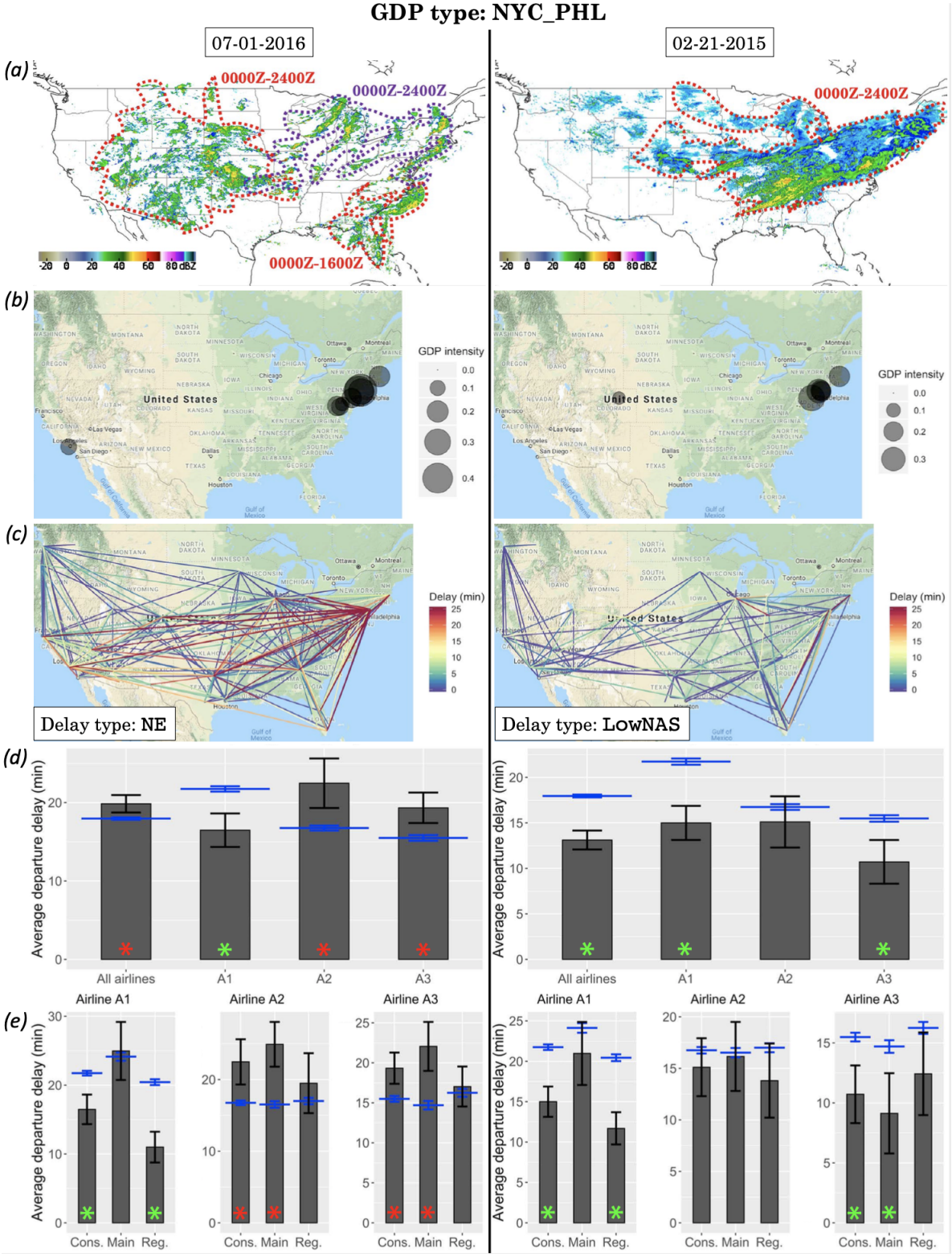


Figure 2-14: Example of two NYC_PHL GDP type-of-days. *(a)* Composite weather radar returns from 0000Z to 2400Z. *(b)* Average GDP intensities by airport. *(c)* Delay networks. *(d)* Airline-level benchmarks. *(e)* Mainline and regional fleet benchmarks. Note: horizontal blue lines in *(d)* and *(e)* represent baseline values, a 99% two-sided confidence interval for the mean is plotted in black, the asterisk indicates a statistically significant difference between the observed mean and the benchmark, the color of the asterisk indicates if the observed delays on the particular day were higher or lower than the mean.

a statistically significant improvement over its baseline. This analysis illustrates the importance of conditioning on capacity impacts while benchmarking: without doing so, one would rate A1 and A2's delay performance to be similar. However, A1 performed better than its NYC_PHL GDP type-of-day baseline, while A2 performed worse. In other words, A1 not only outperformed its typical performance for NYC_PHL GDP type-of-days, but it also outperformed A2, even though A2's delay performance is typically better on similar GDP type-of-days.

Figure 2-14*(e)* compares the performance of mainline and regional carriers. On July 1, 2016, even though A1 saw a reduction in delays, most of the reduction could be attributed to their regional carrier. On February 21, 2015, both A1 and A3 saw reduced delays for their consolidated fleet; the improvement for A1 was its regional operations, while it was the mainline carrier for A3. The

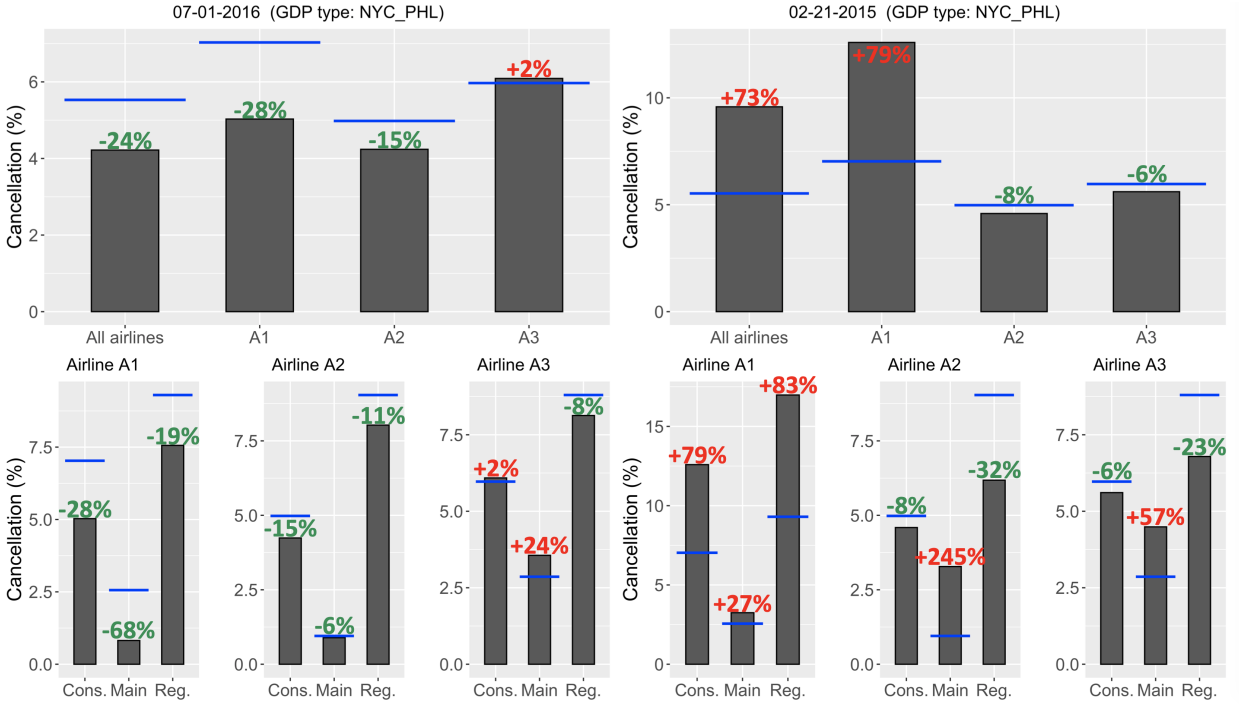


Figure 2-15: Cancellation percentage benchmarks for the two NYC_PHL GDP type-of-days considered in Figure 2-14. The horizontal blue lines represent baseline values.

cancellation percentages (Figure 2-15) provide a more complete picture of the two days described in Figure 2-14.

On February 21, 2015, A1 and A3 saw a statistically significant decrease in delays compared to their baselines. However, A1 canceled significantly more flights (79 percentage points higher) compared to A3. We see that A1 canceled heavily within its regional and mainline operations,

whereas A3 selectively canceled mainline operations. In fact, the cancellation rate in A3's regional operations was below average. Of course, different external factors (e.g., airline networks, schedules, etc.) could have been more favorable for one airline, even if they both experienced the same GDP impacts. Our analysis helps identify such days for closer inspection. The relative benefits and costs of canceling mainline vs. regional operations for such a GDP type-of-day is a question for future research.

The second case study is presented in Figure 2-16. Both of the days within this benchmarking case study – January 8, 2017, and February 6, 2015 – were classified as SFO GDP days. The weather radars between January 8, 2017, and February 6, 2015, look remarkably similar, with the latter-day having a secondary weather pattern with minor GDPs affecting the East Coast during the early morning hours.

Despite the similarities in terms of GDP issuance, the delay networks for both days are significantly different (Figure 2-16*(c)*). While severe West Coast and transcontinental route delays can be seen in the delay networks for both days, there are also heavy delays on many East Coast and Mid-Atlantic routes for January 8, 2017. In fact, unlike the previous case study where proactive cancellations explain a large portion of the decrease in average departure delays, in this SFO GDP case study, cancellations across the network were not significantly higher than the baseline values. In fact, on January 8, 2017, cancellations across all airlines were higher than the baseline (6.61% compared to 2.32%), yet the departure delays were significantly worse than baseline values. This insight highlights the caveats that must be taken into account when assuming that in general delays can always be mitigated via cancellations. It also highlights the significant network impacts of minor East Coast disruptions.

The benchmark summary of January 8, 2017, indicates severe average departure delays across all airlines, with some average delay values more than 400% above the baseline. Airline A2 had the worst on-time performance, both in an absolute sense as well as relative to its SFO GDP baseline. Even though airline A3 had the lowest average departure delays among the three major US airlines, it performed worse (144% increase in average departure delays) than airline A1 (77% increase in average departure delays) when benchmarked against its own baseline. We also note that on January 8, 2017 the mainline carrier for airlines A1 and A2 performed worse (123% and 612% increase, respectively) than its regional counterpart (44% and 263% increase, respectively),

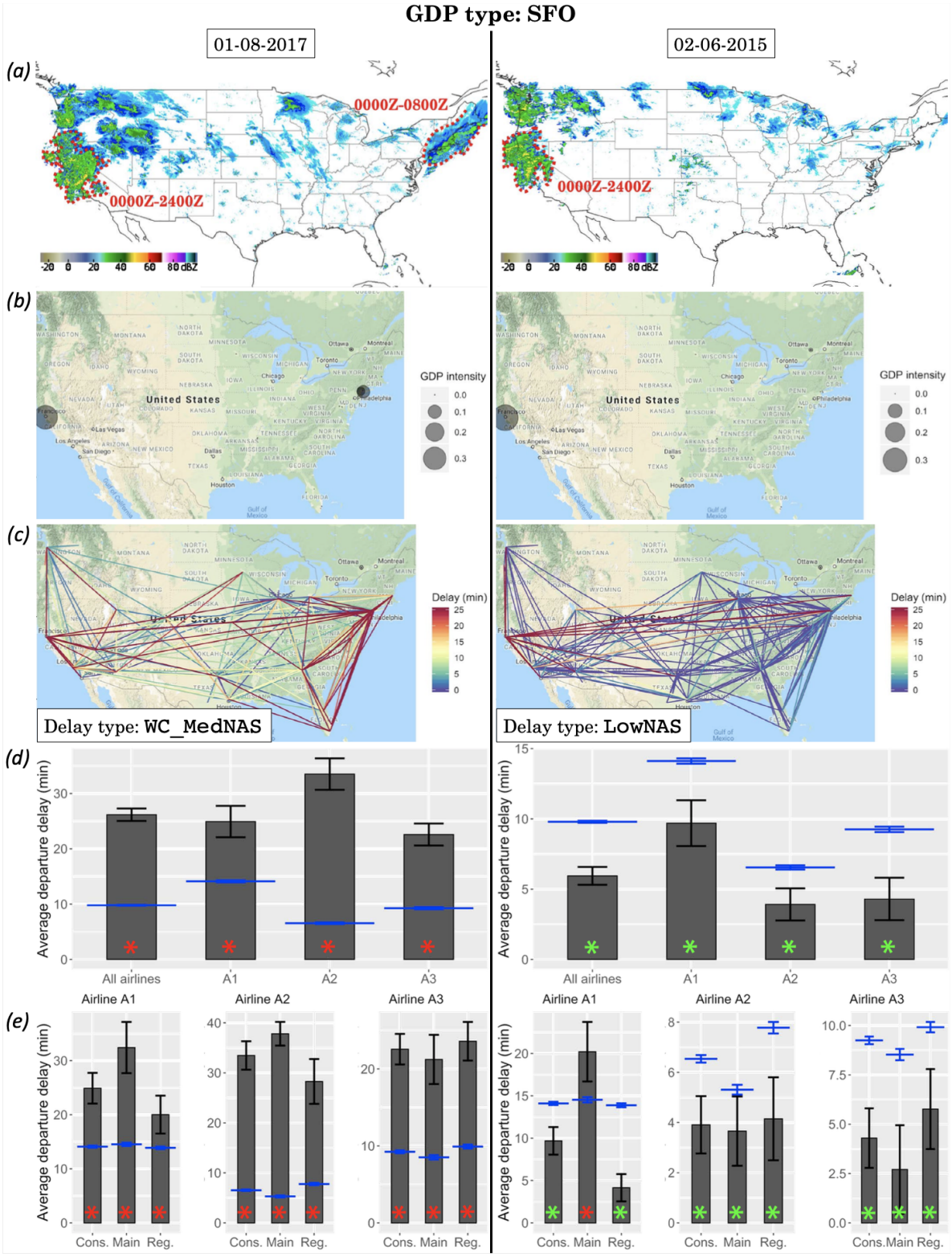


Figure 2-16: Benchmarking for two days classified as SFO-type GDP; *(a)* composite weather radar returns from 0000Z to 2400Z; *(b)* GDP intensity plots; *(c)* delay networks; *(d)* on-time performance of airlines; *(e)* on-time performance by mainline or regional fleet. Note that the horizontal magenta lines in *(d)* and *(e)* represent baseline values. See Figure 2-14 caption for details

whereas airline A3 saw similar performance between its mainline and regional carriers.

Overall, we could conclude that the system on-time performance on February 5, 2016, was better than January 8, 2017, with the delay and cancellation benchmarks reinforcing the conclusion. Airline stakeholders may opt to re-run baseline computations and benchmarking exercises with different metrics such as average arrival delays or A0 and A14 percentages to ascertain if February 5, 2016, under-performed on the arrivals end. Furthermore, to examine specific cases such as the under-performance of airline A1's mainline carrier on February 5, 2016, the user may also choose to isolate airline A1 at an airport-specific and carrier-specific level for further insights.

2.6 Benchmarking for system operators

We present two applications of our clustering analysis for a system operator (the FAA, in the US): the analysis of seasonal and yearly trends in the occurrences of GDPs, and an evaluation of the correlation between GDPs and system impacts (delays and cancellations).

2.6.1 Frequency of occurrence of GDP type-of-days

Figure 2-17 shows the frequency of occurrence of different GDP type-of-days for each year in 2014-2018. First, we observe that nearly half the days in a year exhibit a significant amount of GDP activity. We also note that the `Miami_NE` GDP type-of-day was seen only in 2014, and very rarely thereafter. The increase in the occurrence of `LowNAS` days approximately compensates for the decrease in `Miami_NE` GDP type-of-days, indicating a change in terms of the timing, location, and intensity of GDPs.

The year 2017 saw the highest frequencies of occurrence of `MedNE`, `NYC_PHL`, and `SFO` GDP types-of-days. A possible explanation for the observed increase in `Med_NE` and `NYC_PHL` GDP type-of-days in 2017 is the removal of slot controls at EWR in October 2016, and the subsequent congestion that resulted due to the increase in demand over the next year [122].

GDP occurrences also show seasonal trends (Figure 2-18), since the weather is a major cause of capacity reductions. The months of January through March experience snowstorms in the Mid-west and the Northeast, whereas summers exhibit increased thunderstorm activity near Atlanta and Chicago. As a result, we observe more frequent `ATL` and `CHI` GDP type-of-days in the months of

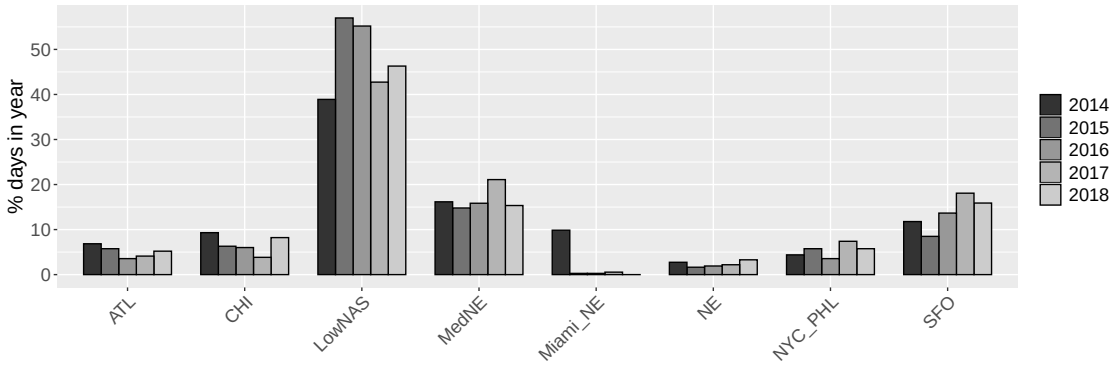


Figure 2-17: Frequency of occurrence of each GDP type-of-day in 2014-2018.

May and June. Somewhat surprisingly, we find that LowNAS GDP type-of-days are more common in winter than summer. A possible explanation is that proactive flight cancellations in advance of snowstorms reduce demand sufficiently, and eliminate the need for GDPs.

2.6.2 Impact of GDPs on system-wide delays and cancellations

GDP occurrences are correlated with flight delays and cancellations. To analyze this correlation at the system-wide level, we consider the triplet (G,D,C), where G, D, and C represent the GDP, delay, and cancellation type-of-day, respectively. This triplet takes one of $6 \times 8 \times 5 = 240$ values. An example of a valid triplet would be (LowNAS, LowNAS, LowNAS), indicative of a day with little GDP activity, and low delays and cancellations.

Frequency of occurrence of (G,D,C) triplets

Even though the clustering and subsequent classification of GDP, delay, and cancellation type-of-days are conducted independently, it is reasonable to expect that they are correlated. Table 2.4 lists the ten most common triplet combinations, which together account for 75% of the days in 2014-2016. As expected from earlier results in Section 2.3, the most frequently-occurring triplet is (LowNAS, LowNAS, LowNAS), which accounts for 37% of the days. Furthermore, five out of the top ten most frequent triplets involve GDPs, delays, or both, on the West Coast, in conjunction with a LowNAS cancellation type-of-day. The only triplet combination in the top ten that is not of LowNAS cancellation type is the triplet (NYC_PHL, NE, Med_CHI_NE), which corresponded to

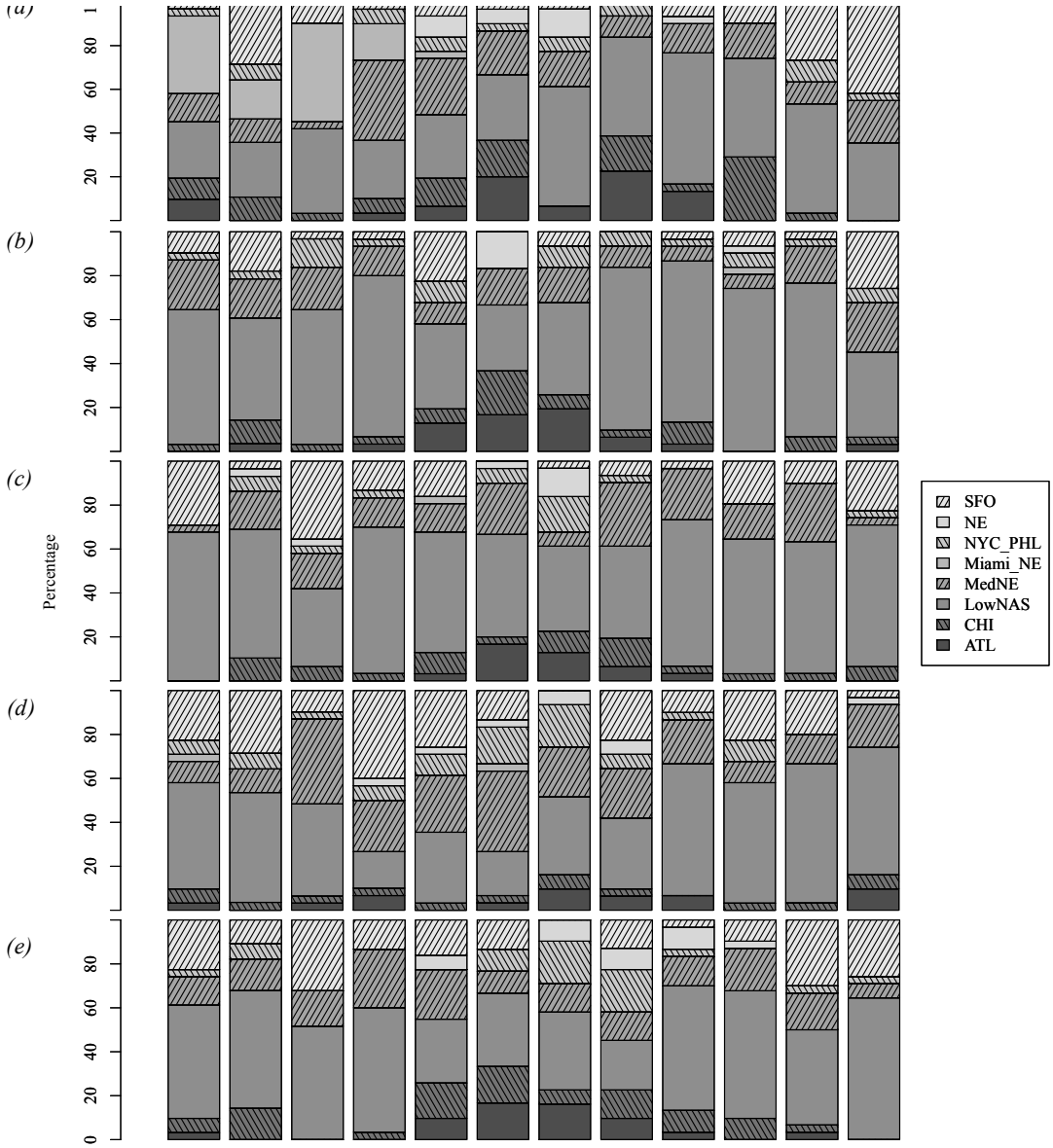


Figure 2-18: Frequency of monthly occurrence of different GDP type-of-days from 2014 through 2018.

approximately 2% of all days.

Rare (G,D,C) triplets present a way to identify outlying days in terms of operational performance. We present three such examples:

- *January 8, 2014*: This was classified as a (`SFO`, `CHI`, `Med_CHI_NE`) type-of-day; in fact, this was the only day in 2014-2016 that was classified as such. Closer investigation revealed that while this day in itself saw little weather and GDP activity outside of SFO, the days preceding it had experienced bad weather and significant snow accumulation in Chicago and

GDP type	Delay type	Cancellation type	# of days (frequency)
LowNAS	LowNAS	LowNAS	402 (36.7%)
LowNAS	WC_MedNAS	LowNAS	101 (9.2%)
SFO	LowNAS	LowNAS	62 (5.7%)
MedNE	LowNAS	LowNAS	60 (5.5%)
SFO	WC_MedNAS	LowNAS	51 (4.6%)
MedNE	WC_MedNAS	LowNAS	42 (3.9%)
MedNE	NE	LowNAS	32 (2.9%)
CHI	CHI	LowNAS	26 (2.4%)
ATL	WC_MedNAS	LowNAS	23 (2.1%)
NYC_PHL	NE	Med_CHI_NE	21 (1.9%)

Table 2.4: Ten most frequently-occurring triplet combinations in 2014-2016.

the Northeast. The triplet corresponding to this day reflects the slow recovery process after a major disruption.

- *September 6, 2014*: In terms of GDP, this day was a NE type-of-day, with significant impacts in the Washington, DC area, Philadelphia, and New York City. Despite the intense GDP activity in the Northeast, it was a LowNAS type-of-day both in terms of delay and cancellations. This type of triplet combination only occurred twice in 2014 through 2016. We hypothesize that the low demand in terms of the number of scheduled flights on this particular day – almost 27% lower than what is typically seen during NE GDP type-of-days, possibly because it was the Saturday following Labor Day weekend – mitigated any congestion-related delays or cancellations.

- *March 2, 2014*: This day was classified as the triplet combination (SFO, WC_MedNAS, Med_CHI_NE). At first, it seems counterintuitive that capacity reductions and delays on the West Coast are associated with significant cancellations in the Northeast. A deeper analysis reveals the presence of a snowstorm over Chicago and the Northeast regions of the US. We hypothesize that this particular winter storm was well-predicted, resulting in extensive proactive cancellations. Since the demand at airports were reduced significantly due to the cancellations, the demand-capacity imbalance was less severe, and required little GDP issuances for the Northeast region.

It is worth noting that only 87 out of 240 possible (G,D,C) triplet labels were assigned at least once in 2014-2016. Most combinations, e.g., (CHI, WC_MedNAS, NE) and (LowNAS, CHI, ATL),

were never observed.

Scenario tree analysis using (G,D,C) triplets

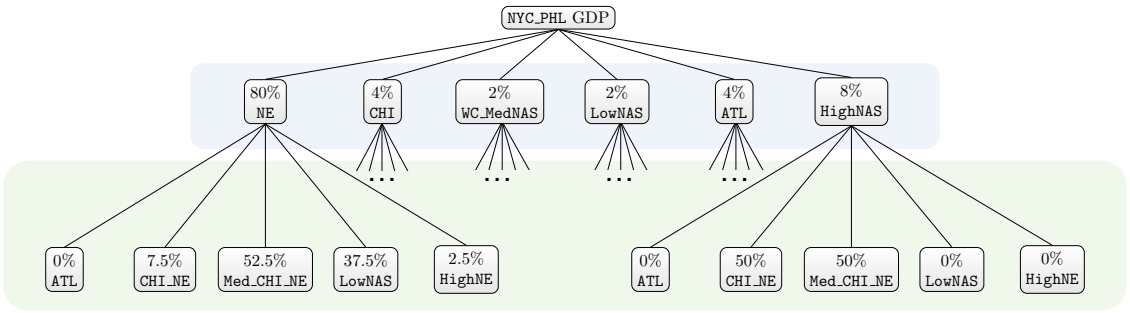


Figure 2-19: Scenario tree with NYC_PHL GDP type-of-day root; delay type-of-day stems (blue background) and cancellation type-of-day leaves (green background). Not all stems and leaves are shown.

Scenario trees present a natural way to interpret the historical correlations between the GDP, delay, and cancellation types-of-days (i.e., the (G,D,C) triplets). For a given triplet, the root node is the GDP type-of-day; the next row presents the probability for delay type-of-days conditioned on the GDP type, and the leaves present the GDP and delay type-of-day conditioned probability of occurrence of a cancellation type-of-day. The scenario tree for a NYC_PHL GDP type-of-day is illustrated in Figure 2-19. We see that most (80%) NYC_PHL GDP type-of-days are NE delay type-of-days, a smaller fraction (8%) are HighNAS delay type-of-days, and it is rare (2%) for a NYC_PHL GDP type-of-day to be classified as a LowNAS delay type-of-day. Furthermore, 37.5% of days classified as NYC_PHL GDP and NE delay type-of-days map to LowNAS cancellation days. By contrast, days associated with the NYC_PHL GDP and HighNAS delay type-of-days will also see flight cancellations, as either CHI_NE or Med_CHI_NE cancellation type-of-days. In other words, similar GDP patterns can result in different cancellation patterns, depending on the magnitude and spatial distribution of delays in the system.

Finally, we note that the scenario tree corresponding to a (G,D,C) triplet is not unique. For example, we can reverse the order of delay and cancellation types-of-day; the resulting scenario tree would help understand the correlations between GDP and cancellation patterns.

2.6.3 Extensions

Several extensions beyond what has been presented in this thesis have been studied, and are summarized below. We refer the readers to [93] for more details.

- Clustering of hourly networks: This chapter focused on clustering time-series of networks to identify characteristic type-of-days. A special case would thus be the clustering of each hour, to assign a label for each hour of the day. This idea is explored in greater detail when developing the MJLS model in Chapter 4.

- Comparison of feature vectors: In this chapter, we argued that the weighted hub-authority feature vector is the most suitable for clustering weighted directed graphs. However, for completeness, we have also clustered individual networks, as well as time-series of networks based on other feature vectors, such as the eigencentrality.

- Clustering of airline-specific networks: Clustering networks that are constructed solely with flights from a specific airline can also be useful. This helps identify typical delay and cancellation patterns for each airline and may be more useful to identify typical congestion and delay hot-spots for an airline.

- Community detection and its relation with network clusters: An extensive analysis of the community structure among the airports has been performed in [93]. There, we show that the community structure remains largely stable for all networks that belong to the same cluster. However, we also find and discuss interesting differences in the community structure across networks from different clusters.

Chapter 3

Diffusion Model for Delay Dynamics

The air transportation system has evolved into a large-scale, interconnected network with many interacting elements. As a result of a large number of shared airport and airspace resources, disruptions in one part of the system can propagate to many others. A significant portion of this propagation occurs at airports (that is, the nodes of the air transportation network), where incoming aircraft continue on the subsequent legs of their planned itineraries, crew members may connect to other flights, and passengers also connect to other flights. Aircraft connectivity is known to be a key driver of flight delays; nearly one-third of all delayed domestic departures (and 40% of departure delays in minutes) in the United States are due to the late arrival of the aircraft on its previous leg. Air carrier delay (a category which includes crew connections) is the associated cause for another 28% of delayed departures (and 32% of delays in minutes) [1]. The above statistics suggest that flows of aircraft and crew through airports are the dominant mechanism by which delays propagate through the system. This chapter therefore proposes a model that relates traffic flow in the air traffic network to the flow of delays, and uses the model to analyze the dynamics of delays on the network.

Prior work: Network models have been previously proposed for a vast range of problems, from disease epidemics [52] and rumor propagation [53], to engineered systems such as power grids [54, 55], the Internet [56], roads [57], public transport [58], railroads [59] and even air transportation [123]. They have also been used to evaluate connectivity between airports in terms of operations [38, 39, 40].

Queuing theory offers a powerful approach to model delay propagation in networks. These

models involve simulations that track the evolution of each aircraft in the system. The airspace and airport bottlenecks are modeled as queues. Flight schedules are inputs to these simulations, and the position and delay of every flight in the system is tracked [124, 125, 31, 30, 126]. These methods require complete information regarding the flight schedules and the capacity of the airspace and airport. Most of the input parameters, like crew schedule, connecting passenger information, schedule slack, and the future airport capacity are not known. They are either information private to airlines, or challenging to estimate. Estimating the airport and airspace capacity are research problems in themselves and robust models are limited to a handful of the busy airports [127, 128, 129, 130]. Scaling up such first principles models to represent the dynamics for multiple airports with thousands of Origin-Destination pairs (OD pairs) is a challenge. Our approach involves simplifying the queuing-theoretic models to improve analytical tractability while still retaining enough expressivity to capture the complex dynamics of air traffic delays.

Numerous prior studies on modeling air traffic delay propagation [131, 132, 133, 123, 31, 134, 135, 30] have revealed the underlying complexities of the process, and the inherent challenges in predicting system behavior [136, 137, 138, 139, 101]. These complexities include the characteristic that while traffic flows through airports result in a spread of delays, the build-up of queues can result in a persistence of delays even after aircraft depart from the airport (or a weather disturbance subsides), and the buffers or slacks contained within flight schedules can help mitigate delays (that is, remove delays from the system). Another challenge is that the interactions between different pairs of airports occur at different time-scales due to differences in flight times between them. For example, it may only take an hour for delays to propagate from Boston to an airport in New York City, while it may take several hours for delays at Boston to propagate to San Francisco. This chapter proposes a model that allows for these phenomena, while accounting for the fact that delay propagation is primarily driven by traffic flows between airports.

While the proposed model can be adapted to any given structure of traffic flows between airports (or even delay flows [140]), an illustration using traffic demand data from the Bureau of Transportation Statistics [1] for 2011 – in particular, a network whose edges are weighted by the average number of daily flights between two airports – is presented as a proof-of-concept. Different system performance characteristics, including the resilience of different airports (as measured by the length of time before delays mitigate after a disruption), the delays induced by disruptions

at different airports, and the number of airports that are impacted when a given airport experiences disruptions, can all be evaluated and compared using the proposed modeling approach.

Key Contributions: We make two key contributions in this chapter.

1. We develop an intuitive model for the evolution of flight delays that incorporates traffic-dependant diffusion of delays at airports, variable time-scales of interaction among airports, absorption of delays due to schedule buffers, and impact of external disruptions.

2. We demonstrate applications of this model to a) quantify the resilience of the air transportation system to impulse and sustained perturbations, and b) evaluate system-wide benefits of delay reduction at airports.

Outline: Section 3.1 presents the delay diffusion model for airport delay dynamics. The setup for the subsequent experiments and the associated evaluation metrics are described in Section 3.2. We study the case of an impulse external disturbance in Section 3.3 and the case of a sustained external disturbance in Section 3.4. Finally, we present a case study on evaluating system-wide impact of airport delays in Section 3.5, and conclude with some extensions in Section 3.6.

3.1 Developing the Delay Diffusion Model

We model the air traffic network as a weighted directed network with N *vertices* (nodes or airports). Each *edge* is represented as an ordered pair, (v_1, v_2), denoting a link from v_1 to v_2. Edge (i, j) is assumed to have a non-negative weight associated with it, representing the flow of traffic from one node to another. The adjacency matrix is given by $\Theta = [\theta_{ij}]$, where each element θ_{ij} denotes the number of flights from airport i to airport j. Figure 3-1 shows the network of average daily traffic in 2011.

The state of airport i at time-step t, denoted $x_i(t)$, is defined as the average delay per flight at airport i at time t. The state vector of the system at time-step t is therefore given by

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots x_N(t) \end{bmatrix}.$$

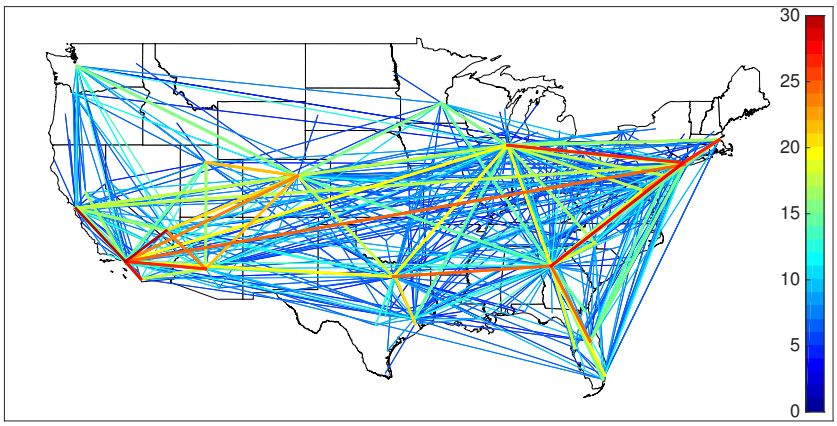


Figure 3-1: Average daily traffic in 2011 [1]. The color denotes the average number of flights on each link

The proposed model of delay dynamics must reflect the following characteristics exhibited by air traffic delays:

- **Redistribution of delays**: The basic premise of the proposed model is that delays at an airport tend to be redistributed amongst the traffic traversing through it. For example, consider a particular airport where 10 flights are arriving with a delay of 25 min/flight and 5 flights arriving with a delay of 10 min/flight. In the absence of any other effects (such as slack in the schedules), if the outbound delay equals the inbound delay, then the average delay of a flight leaving the airport will be $\frac{10\times25+5\times10}{10+5} = 20$ min/flight. If there are two outbound links (going to airports k and l, respectively, with 5 flights on each route), then the network will tend to redistribute 10 min/flight to each of the airports k and l.

- **Persistence of delays**: Another attribute of air traffic delays is their tendency to persist at an airport, even after the disruption has ended. One reason for this phenomenon is the build-up of queues or workload that then takes time to subside. The proposed model reflects this behavior by assuming that a fraction $\alpha \in [0,1]$ of the delay level at an airport at any time persists through the next time-step. In the current model, we set $\alpha_i = \alpha \; \forall i$. As α decreases from 1 to 0, the inertia at airports (persistence of delays) increases.

- **Slack in the system**: Flight schedules are known to contain some amounts of slack or buffering that can mitigate delay propagation to a certain extent [132]. This slack takes the form of longer-than-necessary block times (that is, schedule padding) or long turnaround times between two consecutive legs of an aircraft itinerary. In either form, the available slack in the

schedule prevents the propagation of delays that are within the amount that can be handled by the buffer. This attribute is modeled by a slack term of β min/flight on each link of the network. This term serves to decrease the propagation of delays through a link.

- **Multiple time scales**: Because distances (and therefore flight times) over different links vary, airport delays interact over multiple time scales. In other words, while it may take 1 hour for delays to propagate from Boston (BOS) to New York's LaGuardia (LGA) airport, while it may take several hours for them to propagate from BOS to Dallas/Fort Worth (DFW) airport. The network model is augmented with pseudo-nodes to model these variations in time-scales. In our discrete-time model, each time-step corresponds to 1 hour; it should take multiple time-steps for delays to propagate between two airports that are more than an hour's flight time apart. The augmented network is created by inserting pseudo-nodes between airports that are more than 1-hour apart. If the transit time along the edge from node i to j is h hours, then we introduce a chain of $h-1$ pseudo-nodes between them, each 1-hour from the next. The traffic on each of these edges will be θ_{ij}. If P pseudo-nodes are added, the augmented network contains $V = N + P$ nodes. The adjacency matrix of the augmented matrix is denoted $A = [a_{ij}] \geq 0, A \in \mathbb{R}^{V \times V}$.

Figure 3-2 shows the original and augmented networks for the case when one of the edges has a transit time of 2 hours and another has a transit time of 3 hours.

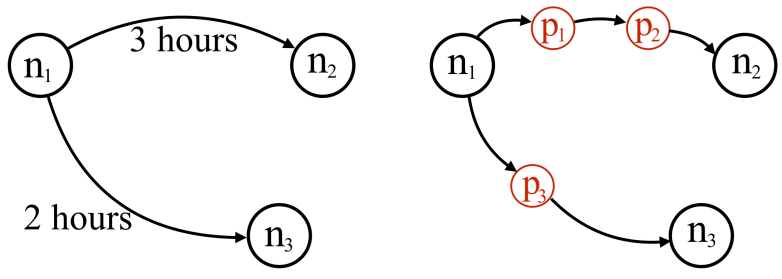


Figure 3-2: (Left) Original network. The flight time from node n_1 to n_2 is 3 hours, and from node n_1 to n_3 is 2 hours. (Right) The augmented network has three pseudo-nodes p_1, p_2 and p_3

Suppose the traffic matrix is given by

$$\Theta = \begin{bmatrix} 0 & \theta_{12} & \theta_{13} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The augmented network has 6 vertices, $V = \{n_1, n_2, n_3, p_1, p_2, p_3\}$ and its adjacency matrix is given by

$$A = \begin{bmatrix} 0 & 0 & 0 & \theta_{12} & 0 & \theta_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \theta_{12} & 0 \\ 0 & \theta_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \theta_{13} \end{bmatrix}.$$

- **Exogenous input**: The injection of delay into the system at an airport i is assumed to take the form of an exogenous input, $u_i(t)$. This delay could be caused by bad weather or other disruptions, or because of traffic management initiatives (such as ground delays or ground stops) that are issued in response to such disturbances. We assume that this input is non negative.

3.1.1 Model Description

The model features described earlier can be combined to determine the equations that govern the evolution of the state vector, $\mathbf{x}(t)$. First, we note the presence of two factors: the persistence of delays at an airport, and the redistribution of delays due to network effects. We assume that the delay at any time-step is a convex combination of these two factors at the previous time-step. As α increases and delays persist from one time-step to the next, the influence of network effects decreases. Second, we note that the network effects term represents the average delay level of the airport and depends on the incoming delay, which in turn depends on the incoming traffic flows. However, some of the incoming delay is mitigated by of the slack on each link leading into that node. The pseudo-nodes are assumed to just transfer the delay along incident edges. The resultant

equations for the evolution of delays are described by (3.1)-(3.2):

$$x_i(t+1) = \alpha x_i(t) + (1-\alpha)\frac{\sum_j a_{ji}\ (x_j(t)-\beta)^+}{\sum_j a_{ji}} + u_i(t), \forall i \in N; \tag{3.1}$$

$$x_i(t+1) = \sum_j \mathbb{I}_{a_{ij}>0}\, x_j(t), \forall i \in P, \tag{3.2}$$

where $\left(x_j(t)-\beta\right)^+ = \max\{x_j(t)-\beta, 0\}$, and $\mathbb{I}_{a_{ij}>0}$ is an indicator variable which is 1 when $a_{ij} > 0$ and 0 otherwise.

The term $\frac{\sum_{j\in V} a_{ji}\ \max\{x_j(t)-\beta,0\}}{\sum_{j\in V} a_{ji}}$ is the traffic-weighted incoming delay. The max operator ensures that the delays do not become negative; it however results in nonlinear system dynamics. The exogenous input is given by $u_i(t)$.

Although not considered in this work, an extension of the model would involve airborne delays as well. These delays could be injected via an edge specific input $\tilde{u}_{ij}(t)$. The network effect term would then be $\frac{\sum_{j\in V} a_{ji}\left(x_j(t)-\beta+\tilde{u}_{ij}(t)\right)^+}{\sum_{j\in V} a_{ji}}$. $\tilde{u}_{ij}(t)$ can be positive as well as negative and can be used to model delay reductions.

3.2 Setup for Evaluating the Model

The proposed network model can be adapted using operational data from a variety of possible sources, such as the Aviation System Performance Metrics (ASPM) database [103] and the Bureau of Transportation Statistics (BTS) database [1]. The primary difference is that the BTS data only includes records of US carriers which accounted for at least 1% of passenger revenues, and therefore does not include airports that are served by smaller carriers, air taxis, etc. We illustrate our approach through an analysis of the model developed using operational air traffic data from BTS for the year 2011 [1]. We only consider links (Origin-Destination pairs) that have at least 5 flights per day. The rationale for not including the smaller links is that there is likely to be sufficient slack on these routes that delays would tend not to propagate through them, that is, their contribution to delays at other airports would remain negligible. The resultant traffic network contains 158 airports and 1,102 links, as shown in Figure 3-1. The top 10 airports in this network (in terms of

traffic between the nodes) are listed in Table 3.1. We note that these counts of daily departures are smaller than the daily operational counts from other databases (such as OPSNET [141]) because they only account for traffic within this network, and not international flights, smaller/international air carriers, and flight legs with infrequent service.

Table 3.1: Top 10 airports by traffic in the network model

Airport	**Avg. no. of daily departures**
Atlanta (ATL)	912
Chicago (ORD)	662
Los Angeles (LAX)	489
Dallas (DFW)	486
Denver (DEN)	468
Phoenix (PHX)	417
San Francisco (SFO)	300
Las Vegas (LAS)	285
Houston (IAH)	279
Charlotte (CLT)	239

We use distance as a proxy for time to determine the augmented network, which has $2,554$ nodes (and pseudo-nodes). The average traffic flow matrix for each hour, Θ, is symmetric and is used to construct the adjacency matrix of the augmented network, A.

3.2.1 Exogenous Inputs

In the subsequent analysis, we use the exogenous input $u_i(t)$ to simulate the injection of delays into a particular node (termed the *inducing airport*) and evaluate the resulting behavior of the system. In particular, we focus on two types of exogenous input functions:

1. *An impulse input*, where we introduce a certain amount of delay at a particular time-step (in this case $t = 0$), and then maintain $u_i(t) = 0 \; \forall t > 0$.

2. *A constant set-point*, where we vary the control input in order to keep the delay $x_i(t)$ at a particular airport i at a fixed set-point x_i^*.

These two input functions simulate transient and sustained delay respectively. Although a sustained delay will mean that $u_i(t)$ is a constant, in the steady-state it is equivalent to having a

constant set point x_i^*.

3.2.2 Performance Metrics

We use the following performance metrics to evaluate the system behavior in response to the exogenous inputs:

1. *Total delay*, namely, the sum of the delay levels seen at all the airports in the network at time-step t, that is, $\sum_{j \in N} x_j(t)$.

2. *Average induced delay*, namely, the average delay level seen across all airports when an exogenous input is introduced at an inducing airport. It is defined as:

$$\bar{\text{ID}}(t) = \frac{\text{total delay}(t)}{|N|},$$

   and is the expected delay level that an airport will see under that particular exogenous input and inducing airport.

3. *Largest impacted cluster*, or the largest set of connected airports that have a non-zero delay. If we have an exogenous input only at one airport, then the size of the largest impacted cluster (also known as the giant component) is simply the number of airports with non-zero delays. This is because the dynamics are such that the delay spreads only when there is a connected path. Further, airports with near-zero delay $x_i(t)$ are also counted in this metric. Whenever an airport has a non-zero delay, it means that there is at least one incoming link with a delay of greater than β min/flt. Consequently, the largest impacted cluster includes airports that have high incoming delays but propagate very little.

3.3 Impact of an Impulse Disruption

An impulse input at an inducing airport k is of the form $u_k(0) > 0$ and $u_k(t) = 0 \; \forall t > 0$. The exogenous input is assumed to be zero at all other airports. This represents a scenario in which there is a sudden, brief disruption at an airport. The system response depends on the propensity towards the persistence of delays (that is, α) and the schedule slack (that is, β).

- If $\alpha = 1$, the system is completely inertia-driven. The impulse will be isolated, but persist indefinitely.

- If $\alpha \in (0,1)$ and $\beta = 0$, then there is no slack in the system and delays will disperse through the network, and persist indefinitely. Lemma 1 characterizes this scenario:

  Lemma 1. *Consider a system governed by* (3.1)-(3.2) *with* $\alpha \in (0,1)$, $\beta = 0$ *and an associated symmetric traffic matrix. If we introduce an impulse delay* $x_k(0)$ *at an airport k, the system will reach a steady-state where the delays at all the airports will be given by*

$$x^{\text{SS}} = \frac{x_k(0)\deg(k)}{\sum_{i \in N} \deg(i)}.$$

- If $\alpha = 0$ and $\beta = 0$, then the delays can keep oscillating and not converge. A simple example is a two-node network where one airport receives an impulse delay. The delay will keep getting transferred between the two airports.

- If $\beta > 0$, then we have slack in the system and delays will get absorbed. In other words,

  Lemma 2. *Consider a system governed by* (3.1)-(3.2) *with* $\beta > 0$. *For any delay impulse,* $|x_i(t)| \to 0$ *as* $t \to \infty$, $\forall i \in N$.

The dynamics for an intermediate range of α and β are simulated for an impulse input delay of 120 min/flight at Chicago O'Hare (ORD) International Airport. In Figure 3-3, we plot the dynamics of the average induced delay and the size of the largest connected cluster. First, we note that the time for delays to decay to zero increases with α. This increase is nonlinear and grows rapidly as α approaches 1. The parameter α is also related to the response time of the system: When it is low, the system can share the delay with other airports much faster. This distribution enables more flights to use up the slack in their schedule to mitigate the delay. However, when α is low, the peak delay seen is also higher. Thus α determines whether the system will experience a low level of delay for a long period of time or a high level of delay for a short time. The maximum size of the largest impacted cluster is found to be independent of α (and around 70).

With increasing β, the average induced delay and the number of impacted airports both decrease. This is shown in Figure 3-4. Comparing it with the effect of α, we observe the following.

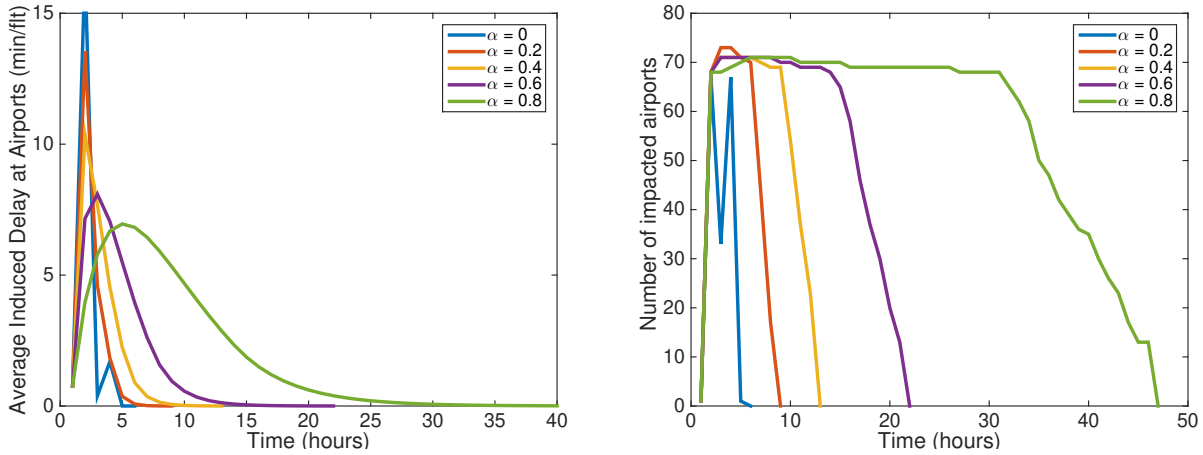


Figure 3-3: (Left) Avg. induced delay, and (right) number of impacted airports for varying α, for an impulse input of 120 min/flight at ORD and $\beta = 10$ min/flight.

When we decrease α, the delays reduce and persist for longer, where as when we increase β, the delays scale down proportionally. This highlights the fundamentally different way in which these parameters impact delay dynamics. Figure 3-5 shows the time needed for delays to subside, for varying values of α and β, for an impulse input of 120 min/flight at ORD.

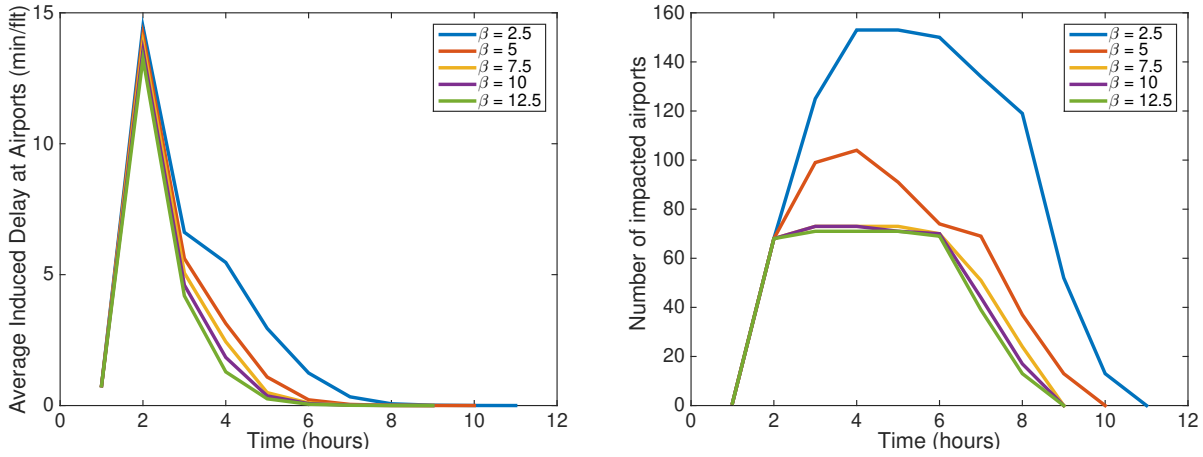


Figure 3-4: (Left) Avg. induced delay, and (right) number of impacted airports for varying β, for an impulse input of 120 min/flight at ORD and $\alpha = 0.2$.

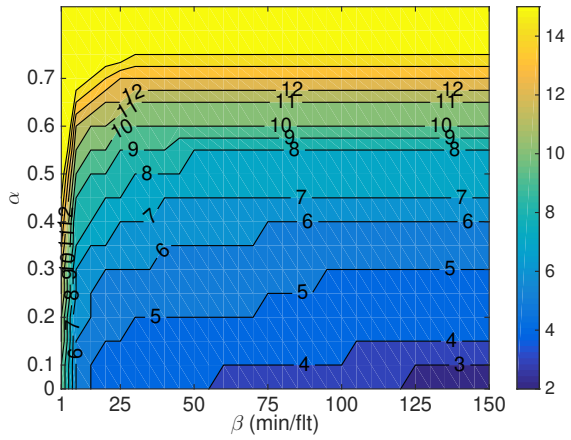


Figure 3-5: Contour plot showing the time needed (in hours) for delays to subside, for varying values of α and β, for an impulse input delay of 120 min/flight at ORD.

3.4 Impact of a Sustained Disturbance

In this section, we study the system behavior under a constant delay input. Since the system is not linear, the superposition principle cannot be used to relate the constant-input response to the impulse-response from Section 3.3. This scenario is analogous to one in which a traffic management initiative is used to maintain delays at an airport at a specified level or set-point over a sustained period of time. The initial condition is assumed to be $x_i(0) = 0 \; \forall i \in N$. An appropriate "exogenous input" can then be engineered to maintain a set point of x^* for a particular airport k.

When $\beta = 0$, there is no mechanism for delays to be absorbed. The steady-state solution will have all airports at a delay of x^*. When $\beta > 0$, then there is some slack in the system it will reduce the exposure of other airports to the delay input. We look at ORD to study the influence of the slack parameter β and the set point x^*. Figure 3-7 shows the time needed for delays to subside, for varying values of α and β, for an impulse input of 120 min/flight at ORD.

For a given set point x^*, the average induced delay as well as the number of airports impacted in the network decrease with increasing β (Figure 3-6). There are two distinct regions of decrease for the average induced delay plot. For low β values, the decrease is primarily due to the smaller number of airports impacted. At higher β values, the number of impacted airports does not change, and the decrease happens only at all those airports that are directly connected to the delay-inducing airport. When $\beta \geq x^*$, no delay gets transmitted. It is interesting to note that for a wide range of

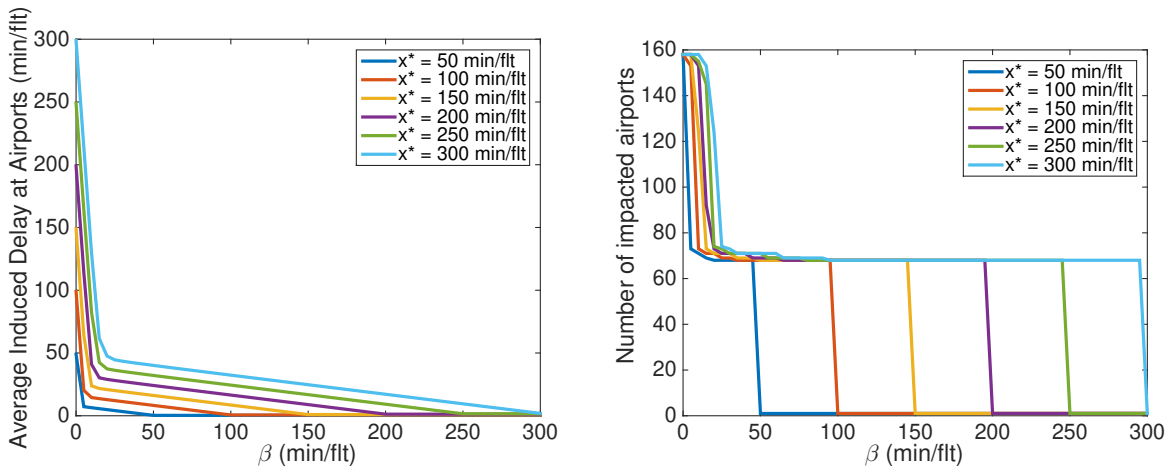


Figure 3-6: System performance for different set-points x^*_{ORD}. Variation of (left) average induced delay and (right) number of impacted airports, with β.

x^*, there is little benefit (in terms of delay) of investing in a β greater than 30 min/flight. The delays are sensitive to the slack in the system when the slack is small (that is, the schedules are very constrained).

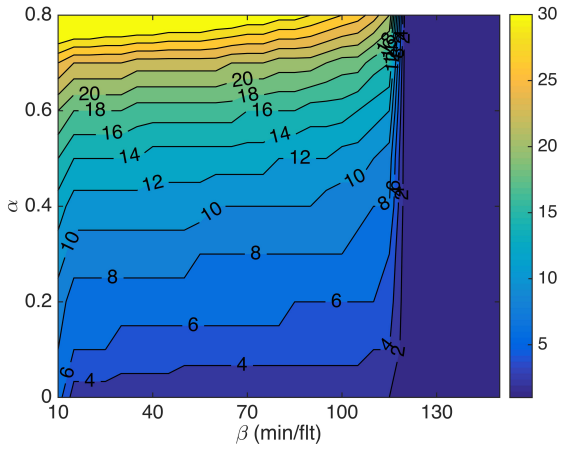


Figure 3-7: Contour plot showing the time needed (in hours) for delays to subside, for varying values of α and β, for a set-point delay of 120 min/flight at ORD.

Variations in β not only change the average induced delay, but also the geographical spread of the delay (Figure 3-8). When $\beta = 5$ min/flight, the exogenous input at ORD induces delays at many airports ranging from Seattle in the west coast to Miami in the south. With $\beta = 10$ min/flight, the delay gets limited to those airports that have a high fraction of their traffic coming directly from ORD.

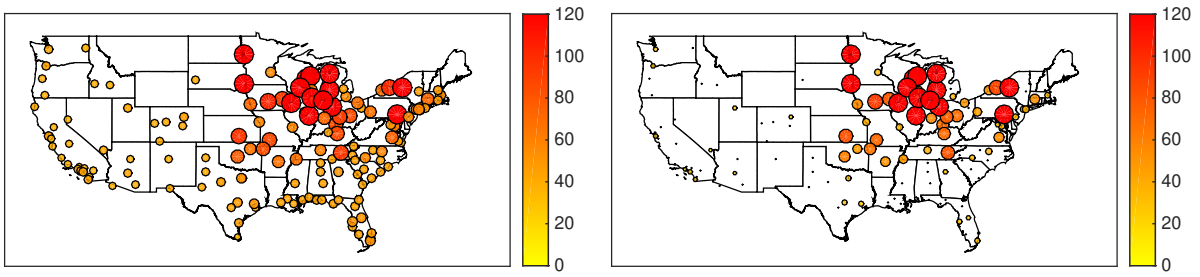


Figure 3-8: System performance with a set-point $x^*_{ORD} = 120$ min/flight, (top) $\beta = 5$ min/flight and (bottom) $\beta = 10$ min/flight. We note the difference in both the number airports impacted, and the induced delays

This analysis of the induced delay is extended to all airports. For each of the 158 airports in the data set, we use the appropriate exogenous input so that the set-point (delay level) is maintained at 120 min/flight. The geographical extent of the spread of delays is shown for some airports in Figure 3-9. The 10 airports which can induce the highest delays throughout the system are listed in Table 3.2.

Table 3.2: Top 10 airports by average induced delay when $\beta = 10$ and set-point $x^* = 120$ min/flight at the inducing airport

Inducing Airport	Average Induced Delay(min/flight)
Atlanta (ATL)	31.75
Chicago (ORD)	17.82
Denver (DEN)	8.30
Dallas (DFW)	7.97
Los Angeles (LAX)	7.28
Phoenix (PHX)	5.42
San Francisco (SFO)	4.73
Baltimore (BWI)	4.37
Houston (IAH)	4.25
Honolulu (HNL)	3.45

It is worth noting that none of the airports in the New York area (EWR, JFK or LGA) appear to be significant in Table 3.2 or Table 3.3. The reasons for these are several: First, both Newark (EWR) and John F. Kennedy (JFK) airports serve large numbers of international flights, which are not included in this analysis. Second, the three airports serve different airline networks, and their connectivity is quite diffused. However, it is worth noting that if the three airports were treated as a single “super-airport", then it would rank 6th in terms of induced delay (just below LAX in Table

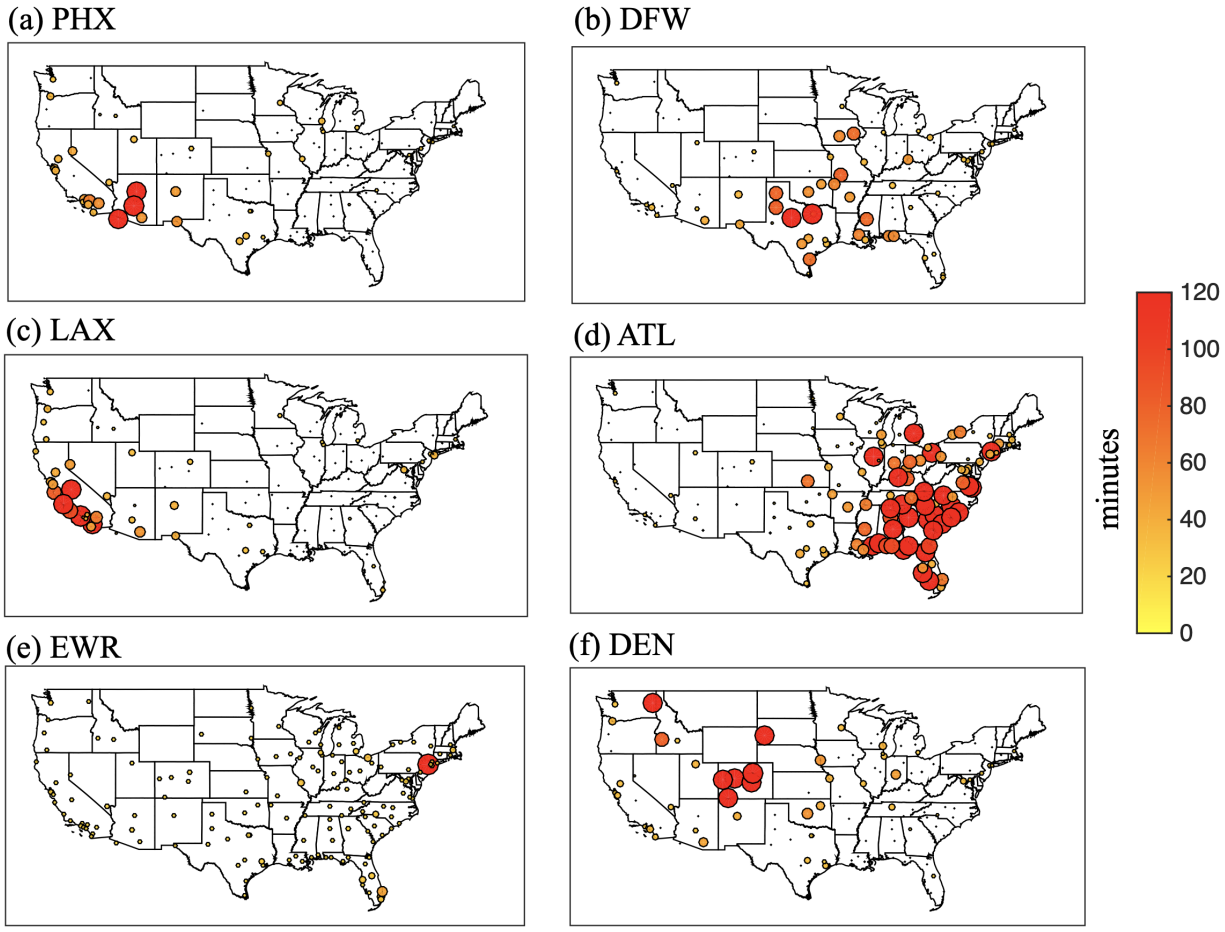


Figure 3-9: Impact on the system when there is a set-point of 120 min/flight at specific airports.

Table 3.3: Top 10 airports by total number of impacted airports when $\beta = 10$ and set-point $x^* = 120$ min/flight at the inducing airport

Inducing Airport	Number of impacted airports
Atlanta (ATL)	125
Chicago (ORD)	74
Los Angeles (LAX)	68
Dallas (DFW)	40
Denver (DEN)	39
San Francisco (SFO)	52
Phoenix (PHX)	45
Houston (IAH)	40
Boston (BOS)	37
Orlando (MCO)	32

3.2) and tie for 8th place in terms of the number of airports impacted (i.e., tied with Houston).

Finally, we note that network effects can cause the induced delays to increase super-linearly with the degree of the node in the network (Figure 3-10). This observation provides further rationale on why merging multiple airports (such as in the New York area) would serve to increase the degree of the network, and thereby significantly increase the induced delays. As expected, we also see that the induced delays decrease as the slack β increases.

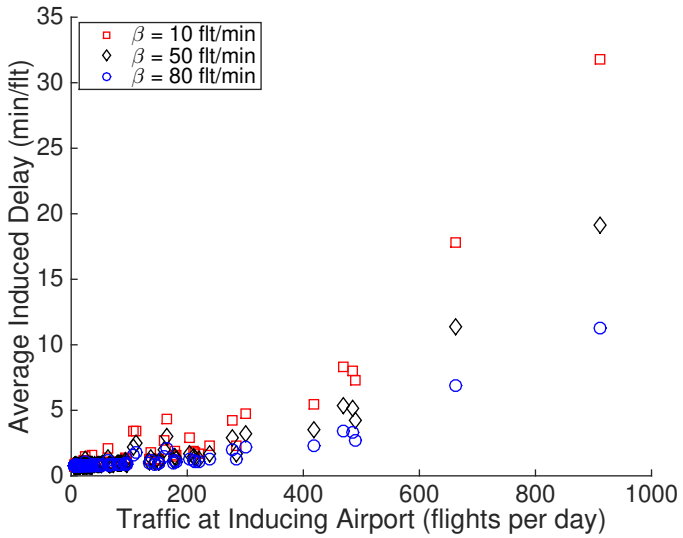


Figure 3-10: Induced delay increases super-linearly with the degree of the node i, where $x_i^* = 120$ min/flight

3.5 Case study: Benefits of Delay Reduction at Individual Airports

The motivation for this case study is the benefits assessment of a proposed departure metering policy at Charlotte (CLT) airport. This metering policy is expected to reduce the delays of outbound flights. We would like to know what the benefits at other connected airports would be. The time period of interest was May-July 2016 and the 30 busiest airports in the US (ASPM Core 30 list) were considered. An estimate of $\beta = 5$ min was obtained by taking the median slack (approximated as the difference between the scheduled turn time and the actual turn time) of all the flights in that period. A sensitivity analysis was performed to find the benefits at different airports. The benefits vary based on the baseline delay levels at CLT. Figure 3-11 shows the average induced delay at all other airports for different levels of a sustained delay at CLT. There are three distinct regions in the figure. When the baseline delays at CLT is less than or equal to 5min/flt, no delay gets propagated, as schedule slack would absorb all the delays. Therefore, if the baseline delay at CLT is already less than 5 min/flight, a reduction in delays at CLT would have no system-wide benefits. When the baseline CLT delays are between 5 min/flt and 150 min/flt, a reduction in delays at CLT will translate into a decrease in delays at other airports. It is important to note that this effect is not linear. A decrease in delay of 1 min/flight when the baseline is 60 min/flt is going to have a different effect from a 1 min//flt decrease with the baseline delay being 100 min/flt. This is because of the network interconnections- as the CLT delay increases, it impacts a non-linearly growing number of connected airports. Lastly, above a baseline delay of 150 min/flt, all the 30 airports start getting affected (the network is saturated), and the relationship becomes linear.

In Figure 3-12, we look at the sensitivity of induced delay at specific airports, due to delays at CLT. The plot shows the slope of the airport-specific induced delay curve, at a CLT baseline delay of 50 min/flight. This highlights that the benefits would not be evenly shared by other airports. Some closely connected airports like Memphis and Dulles would see up to 0.1 min/flight delay reduction per minute decrease at CLT. This means that CLT share almost 10% of its benefit with Dulles and Memphis due to network effects. This analysis is useful for performing a complete benefits assessment of the proposed departure metering procedure at CLT.

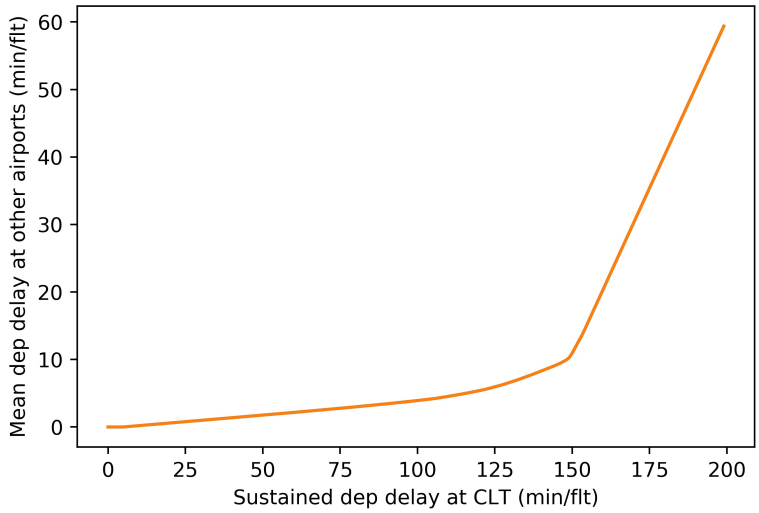


Figure 3-11: Induced delay at other airports due to CLT.

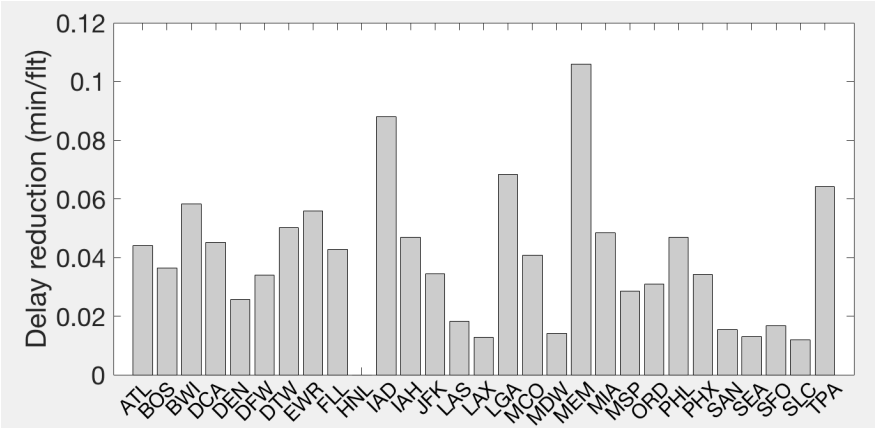


Figure 3-12: Benefit at other airports due to a 1 minute delay reduction at CLT (baseline of 50 min/flt).

3.6 Extensions

This chapter motivated and proposed a new model of delay propagation in air traffic networks. The model accounted for the tendency of delays to propagate through traffic flows at airports, while also accounting for the persistence of delays at an airport, the propagation of delays through the network, and the potential mitigation of delays due to slacks in the schedules. Disruptions were introduced in the form of an exogenous input, and a proof-of-concept illustration with BTS data was provided. The chapter has shown the potential for such models to reflect system behavior, as measured by metrics such as, the amount of system-wide delay induced by disruptions at a particular airport, and the number of airports that are impacted when a given airport experiences disruptions.

We conclude by presenting several directions for future work.

1. Incorporating airport-specific attributes: We can consider the generalization of this approach to account for differences in schedule slack, inertia, etc. between different airports.

2. Learning from data: In the present form, only the traffic flow is learned from data. One extension would be to learn all the model parameters, including the slack β and persistence α from operational data. The main challenge in this approach would be to identify the external input term $u_i(t)$ from real-world data.

3. Varying network topology: Our model assumed that the network topology is constant, i.e., the elements a_{ij} do not change with time. In practice, the flow of air traffic is highly dependant on the time of the day. Thus one could consider a case where $a_{ij} = a_{ij}(t)$. The two ideas of learning model parameters from data, and incorporating time-varying topologies motivates the work we present in Chapter 4.

Chapter 4

Markov Jump Linear System Model for Network Dynamics

This chapter proposes methodologies to model processes on networked systems. In particular, we are interested in modeling the spread of delays in the air transportation network. Three major reasons motivate the need for such models. First, we would like to use improved models to predict future delays accurately. Second, we would like to leverage these models to define, and quantify the resilience of airports (or nodes) to delays and disruptions in the system. Third, we would like to use delay dynamics models to identify control actions that can help minimize the spread of delays in the system. In this chapter, we will focus on developing and analyzing such models. In the next chapter, we will use the model to control the spread of delays.

Prior work: Network models have been proposed and studied for a vast range of systems [51], ranging from disease epidemics [52, 142] and rumor propagation [53], to engineered systems such as power grids [54, 55], Internet [56], roads [57], public transport [58], railroads [59] and air transportation [143]. The application of network science tools to air transportation has been limited to a static analysis of the role of aviation in connecting different cities. Connectivity in the air transportation system has been traditionally modeled in terms of operations, meaning two cities are connected if there exists a direct flight between them [38, 39, 40, 41].

While these models are representative of network interactions in certain settings, they do not encompass some of the behaviors that are important in infrastructure systems. Key among these limitations is the representation of the state of a node. Epidemic models such as Susceptible-

Infected-Susceptible (SIS) or Susceptible-Infected-Recovered (SIR) models assume that a node is in one of a small set of discrete states [142, 144]; by contrast, the nodal state in infrastructure systems are better modeled as continuous variables (for example, delays, traffic volume, capacity, etc.). The type of networks that have been considered in epidemic models tend to be undirected, and unweighted. However, the interactions between nodes in many real systems are not all binary (in that nodes either interact or they do not), but are instead weighted and directed (that is, the interactions are not symmetric). Finally, most infrastructure systems exhibit not just spatial patterns due to network interactions, but temporal patterns such as seasonal or daily trends, resulting in time-varying network topologies (see Figure 4-1). Only very recently has there been an analysis on disease spread over switched networks; however, this work has considered undirected networks for SIS models [145]. We present a switched system model to describe dynamics that involve time-varying network topologies. Prior work on the robustness of networks to external disturbances or perturbations has generally been restricted to undirected (symmetric) networks [146, 147]. These problems are also closely related to those of consensus formation in networked systems [148]. Approaches to analyze robustness have only recently been extended to the case of directed networks, but for fixed topologies [149].

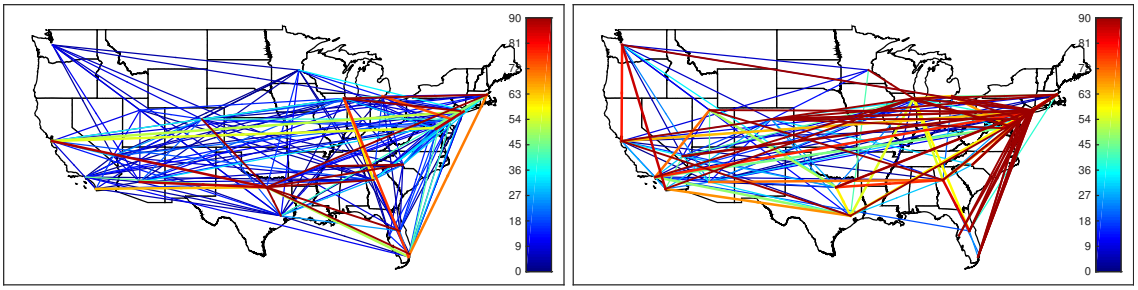


Figure 4-1: Network showing weighted connectivity of delays observed between airports (in minutes) at two different times. Although the networks are directed in nature, the links are colored by the average of the weights in the two directions for ease of visualization.

With respect to predicting and modeling flight delays, queuing theory approaches and data-driven methods have been considered. Chapter 3 presents one example of a diffusion model to approximate the queue dynamics at airports. However, we identified two challenges with the diffusion model - learning model parameters from data, and incorporating time-varying network topologies. In this chapter, we address both these limitations. In contrast to queuing theory approaches, data-driven methods follow a black-box approach, where all factors that could potentially affect the

delay of a particular flight (or a particular route) are fed as input features to make a delay prediction [150, 151, 37, 152]. While the power of these big-data techniques is apparent when looking at the prediction accuracy, they offer little insight into the mechanisms underlying the process. They are not suitable for model development for purposes of control, or to make predictions on scenarios it has never seen in the training data. Our solution involves developing an analytically tractable, and interpretable model for delay dynamics.

Key Contributions: We summarize the key contributions of this chapter.

1. We develop an analytically-tractable, data-driven framework, called a Markov Jump Linear System (MJLS), to model dynamics on networks with time-varying topologies.

2. We validate the performance of the MJLS model in capturing airport delay dynamics with operational data, and show its usefulness in quantifying operational resilience of the airports.

3. We extend prior results on the stability of switched systems to derive conditions for the mean stability, and almost sure stability of a discrete-time positive MJLS. These results indicate that our model for delay dynamics is stable and reflects the evolution of delays in practice.

Outline: We develop the Markov Jump Linear System (MJLS) model in Section 4.1, and present the validation studies in Section 4.2. In Section 4.3, we discuss how the parameters of the MJLS model can be used to quantify the resilience of the system. Finally, in Section 4.4, we identify conditions for the stability of the model and discuss its utility in validating the model.

4.1 Developing the MJLS Model

We present a Markov Jump Linear System (MJLS) model for the hourly progression of airport delays [153]. MJLS are a subset of hybrid systems [154], which have been successfully used to model electromechanical systems [155], epidemic propagation [156], communication networks [157], and multi-agent interactions [158]. The proposed model is constructed using operational flight delay data from the US and incorporates the time-varying network effects that drive delay propagation.

The MJLS model offers three main advantages over the delay-diffusion model presented in Chapter 3. First, the MJLS model incorporates time-varying network topology, which is inherently

a more accurate representation of the underlying air traffic system. Two, the MJLS parameters can be easily learned from data, making it more accurate and reliable. Three, the MJLS model is analytically tractable and lends itself easily to the subsequent development of control strategies.

4.1.1 Data and Setup

An air traffic delay network comprises of nodes corresponding to airports, and weighted, directed edges corresponding to the magnitude of flight delays between the two airports. We define a delay network for each hour of the day, with the edge weight being the median departure delay for all flights that were scheduled to take off on that route during the hour. We use data for domestic air traffic in the US during the years 2011-12 to construct these time-series of weighted-directed networks [1]. We consider only the 30 major airports in the US (FAA Core 30); thus the number of nodes $V = 30$. A network for each hour in the two-year period results in $731 \times 24 = 17,544$ networks, each with a distinct $V \times V$ adjacency matrix, $A(t)$. We denote the total inbound and outbound delays at an an airport i at hour t as $d_i^{\text{in}}(t)$ and $d_i^{\text{out}}(t)$ respectively. These airport delays are related to the elements of the adjacency matrix $A(t) = [a_{ij}(t)]$ as $d_i^{\text{in}}(t) = \sum_j a_{ij}(t)$ and $d_i^{\text{out}}(t) = \sum_j a_{ij}(t)$. Our objective is to develop a model for the evolution of $d_i^{\text{in}}(t)$ and $d_i^{\text{out}}(t)$. With regards to notation, we use boldface to denote vectors (e.g., $\mathbf{x}$(t) for the state vector, $\mathbf{0}$ for a vector of zeros, etc.), $\mathbb{R}$ for the set of reals, $\mathbb{N}$ for the set of natural numbers, $\mathbb{1}_{\text{condition}}$ as an indicator variable, and $\mathrm{I}_{M\times M}$ to denote a $M \times M$ identity matrix. The largest eigenvalue of a matrix is referred as the spectral radius of the matrix. We denote the spectral radius of A as $\rho(A)$.

4.1.2 Topology-dependent Model

The total outbound delay at airport i at time-step $t+1$ depends on two components: the outbound delay at that airport at time t (reflecting delay-persistence) and the delay bound to i from the other airports at time t (modeling the network effect). Similarly, the outbound delay at airport i at $t+1$ depends on its outbound delay at time t as well as the network effect of other connected airports. Combining these two factors linearly using proportionality constants α_i^{out}, α_i^{in}, β_i^{out}, and β_i^{in}, we

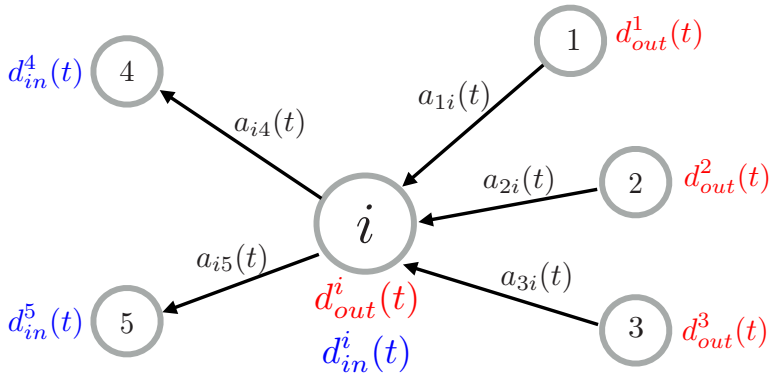


Figure 4-2: Notation for node i and its neighbors

get:

$$d_i^{\text{out}}(t+1) = \alpha_i^{\text{out}} d_i^{\text{out}}(t) + \beta_i^{\text{out}} \sum_{j=1}^{V} \bar{a}_{ij}(t) d_j^{\text{in}}(t) \tag{4.1}$$

$$d_i^{\text{in}}(t+1) = \alpha_i^{\text{in}} d_i^{\text{in}}(t) + \beta_i^{\text{in}} \sum_{j=1}^{V} \bar{a}_{ji}(t) d_j^{\text{out}}(t) \tag{4.2}$$

where $\bar{a}_{ij}(t)$ represent the elements of the row normalized adjacency matrix $\bar{A}(t)$ (i.e., $\bar{a}_{ij}(t)$ is the fraction of the total outbound delay at node i that is destined for node j). The constants α and β, are assumed to be non-negative, and their magnitudes reflect the relative importance of the persistence and the network effect in the delay dynamics. These constants do not necessarily sum up to 1, since delay may not be conserved due to factors such as buffers in flight schedules (which help attenuate delays) or high airport connectivity (which could exacerbate delays). We denote the node-state of the system at any time t as the vector $\mathbf{x}(t)$ which comprises of all the inbound and outbound delays at all airports:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{d}^{\text{out}}(t) \\ \mathbf{d}^{\text{in}}(t) \end{bmatrix}. \tag{4.3}$$

Using $\mathbf{x}(t) \in \mathbb{R}_{\geq 0}^{N \times 1}$, where $N = 2V = 60$, Equations 4.1 and 4.2 are expressed succinctly as

$$\mathbf{x}(t+1) = \Big(\text{diag}([\alpha^{\text{out}}; \alpha^{\text{in}}]) + \text{diag}([\beta^{\text{out}}; \beta^{\text{in}}]) \mathscr{A}(t)\Big) \mathbf{x}(t) = \Gamma(t)\mathbf{x}(t), \tag{4.4}$$

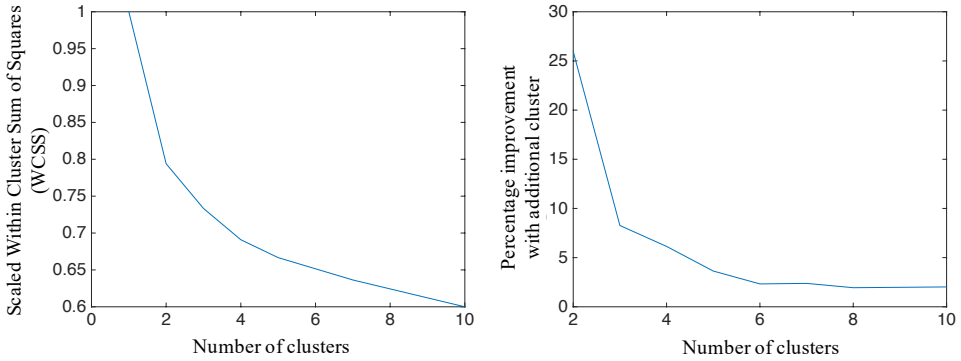


Figure 4-3: Plot of the within-cluster sum of squares (WCSS) distance as a function of the number of clusters (left), and the percent reduction in WCSS with each additional cluster (right). We use $k = 6$.

where $\mathscr{A}(t) = \begin{pmatrix} 0 & \bar{A}(t) \\ \bar{A}^{\mathrm{T}}(t) & 0 \end{pmatrix}$. Thus, Equation 4.4 describes the dynamics of the continuous-state $\mathbf{x}(t)$ as a function of the topology $A(t)$.

4.1.3 Clustering of Network Topologies

Since the evolution of the delays depends on the network topology $A(t)$ which can take many possible values, we aim to find a set of characteristic network topologies. In other words, we want to identify a limited set of matrices that are representative of $\mathscr{A}(t)$. Incidentally, the principal eigenvector of $\mathscr{A}(t)$ is the vector comprising of the hub-and authority scores of a network with adjacency matrix $\bar{A}(t)$ [109]. We emphasize a subtle observation: Equation 4.4 is such that a centrality measure on static networks ends up being relevant in identifying characteristic patterns that govern the temporal dynamics of the system. The feature $\mathbf{f}_{\mathscr{G}}$ is computed for each graph $\mathscr{G}$, as shown in Equation 4.5, is used to perform a k-means clustering of the network topologies. Subsequently, we identify six representative network topologies (based of Figure 4-3 (right)), which are further split based on whether the overall system delay is increasing or decreasing at that time. The $M = 12$ network topologies, $\{A_1, A_2, \ldots, A_M\}$, are shown in Figure 4-4.

$$\mathbf{f}_{\mathscr{G}} = \sum_{i,j} a_{ij} \begin{pmatrix} \mathbf{h} \\ \mathbf{a} \end{pmatrix} \tag{4.5}$$

We will now define two terms that are going to be used for the rest of the thesis.

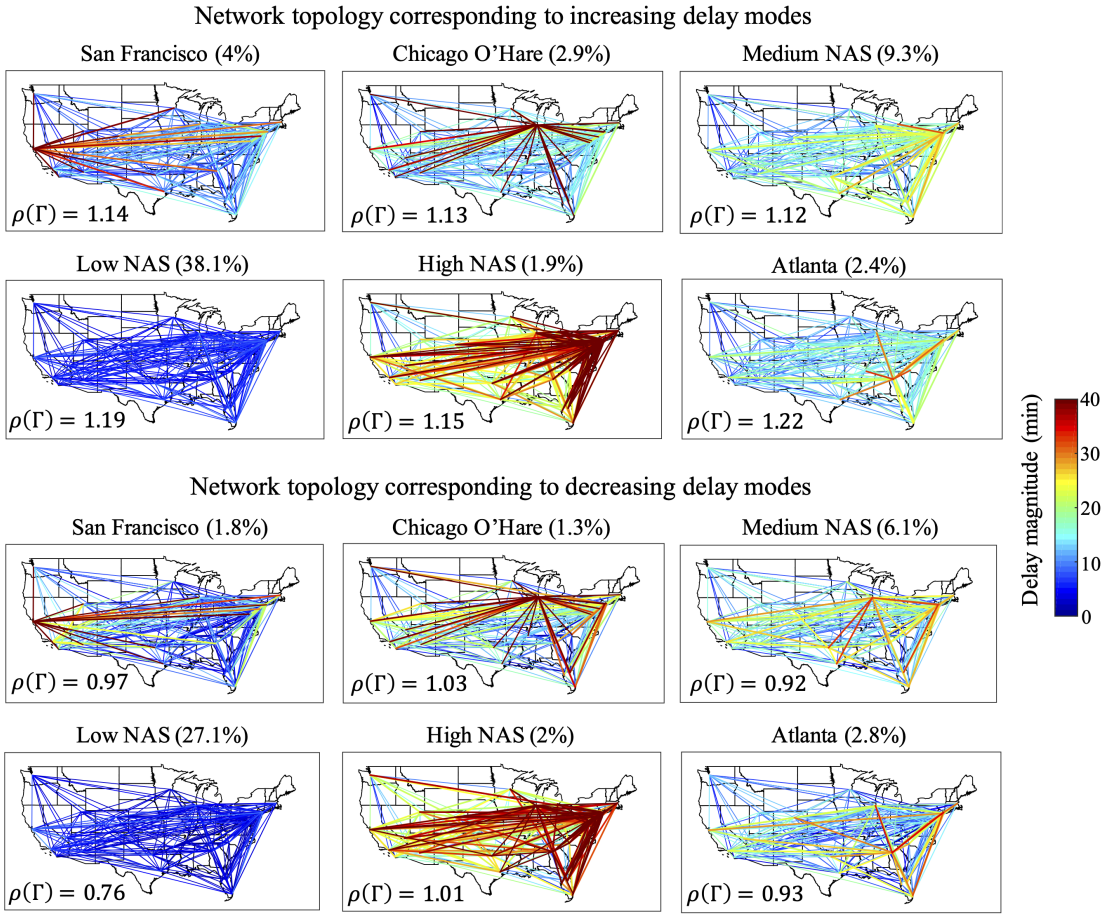


Figure 4-4: Characteristic network topologies (discrete modes) from 2011-12, including delay trends (increasing or decreasing). The modes are named on a qualitative basis, and the frequency of occurrence is shown in parentheses. The spectral radius (magnitude of the largest eigenvalue) of the corresponding system dynamics matrix, $\rho(\Gamma)$ is also presented.

Definition 1 (Node state)**.** *The node state is the vector* $\mathbf{x}(t)$*. It is also referred to as the continuous state.*

Definition 2 (Discrete Mode)**.** *The discrete mode of the system is denoted as* $m(t)$*, and it belonging to one of the M possible options. It is also referred to as the mode or topology.*

4.1.4 Parameter Estimation

The parameters α and β are estimated for each discrete mode (network topology) using a linear regression. These parameters are combined with the topologies $\{A_1,\ldots,A_M\}$ via Equation 4.4 to identify the system dynamics matrices $\{\Gamma_1,\Gamma_2,\ldots,\Gamma_M\}$. The transitions in the network structure are assumed to be a result of Markovian jumps between $M = 12$ possible topologies. In other words, the probability of transitioning from mode $m(t)$ at time t to $m(t+1)$ at time $t+1$, $\mathbb{P}[m(t+1) = j|m(t) = i]$ is denoted $\pi_{ij}(t)$. The transition probability $\pi_{ij}(t)$ (and the corresponding $M \times M$ transition matrix Π_t) is obtained using the maximum likelihood estimator, which in this case is equal to the empirically observed frequency of transitions from mode i to j at time t. The discrete mode transitions may also be associated with resets or jumps in the state of the node. In other words, any transitions from topology i to topology j modify the node state as $\mathbf{x}(t+1) = J_{i,j}\Gamma_i\mathbf{x}(t)$. The state reset matrix $J_{i,j}$ is also estimated by linear regression.

4.1.5 The MJLS Model

The dynamics of the inbound and outbound airport delays is described using a discrete-time switched linear system, where the evolution of the airport delays in each discrete mode is linear, and the discrete mode transitions are governed by a Markov process [159, 160]. An example of a MJLS system with three discrete network topologies is shown in Figure 4-5.

In general, the dynamics of the airport delays $\mathbf{x}(t)$ from mode $m(t)$ at time t is given by:

$$\mathbf{x}(t+1) = J_{m(t),m(t+1)}\Gamma_{m(t)}\mathbf{x}(t) \tag{4.6}$$

$$\pi_{ij}(t) = \mathbb{P}[m(t+1) = j|m(t) = i] \tag{4.7}$$

$$\text{Initial conditions} : \mathbf{x}(0) \geq \mathbf{0} \text{ and } m(0) \tag{4.8}$$

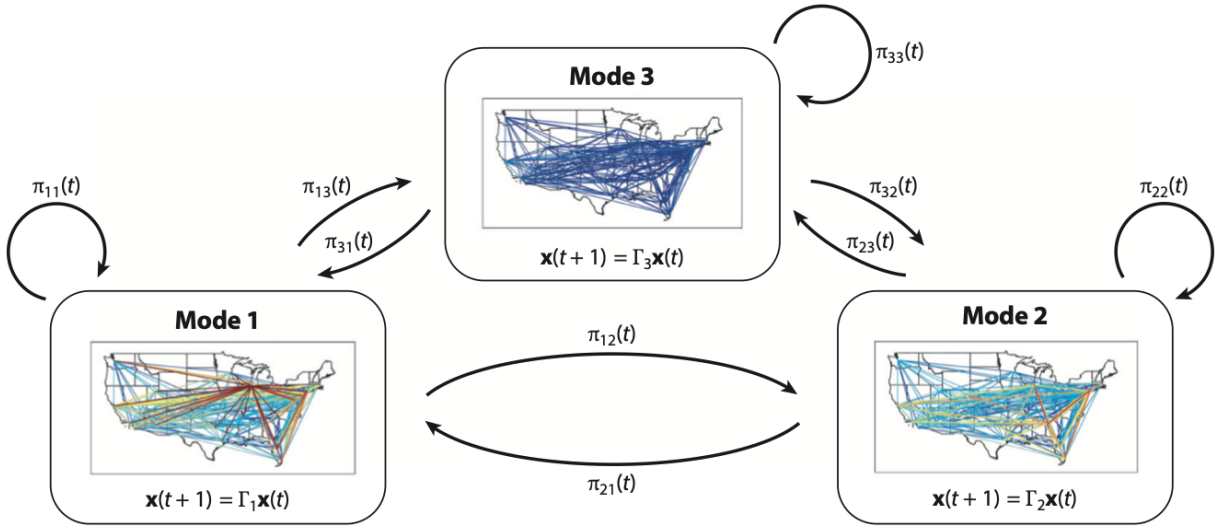


Figure 4-5: A Markov jump linear system with three discrete network topologies

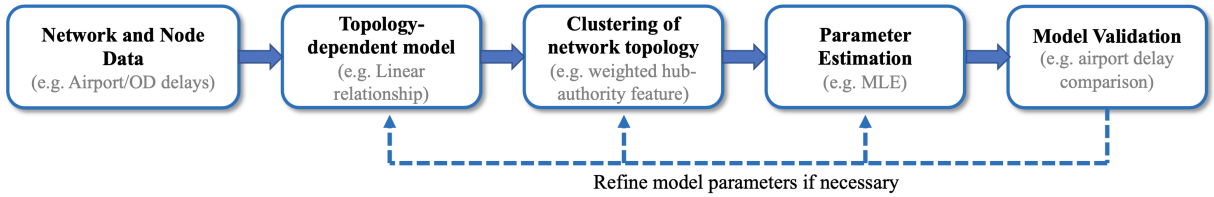


Figure 4-6: Overview of the steps involved in developing a switched-system network model from data. The text in parenthesis describes the adaptations of the steps for the airport delay model.

where the state reset matrix $J_{m(t),m(t+1)} = \mathrm{I}_{N\times N}$ if $m(t) = m(t+1)$.

Definition 3 (Positive MJLS, or PMJLS)**.** *A PMJLS is a MJLS for which the state always remains non-negative for any non-negative initial condition, i.e.,* $\mathbf{x}(0) \geq \mathbf{0} \implies \mathbf{x}(t) \geq \mathbf{0}\ \forall t$.

For airport delay dynamics, all the elements of Γ_i and $J_{i,j}$ are non-negative and thus the system is a PMJLS, consistent with the fact that airport delays are always non-negative.

4.1.6 Broader Applicability of the MJLS Framework

The proposed framework (Figure 4-6) yields a switched-system model for the temporal evolution of a continuous nodal state vector. In our application, the continuous state corresponded to airport delays, but it could be extended to packet latency at routers in a communication network, or the

extent of gossip spread in a social network. In each of these settings, one could express a parametric, topology-dependant model for the nodal state, reduce the model complexity by restricting the topology to a finite set by clustering [161], learn the model parameters by regression, and iteratively refine the parameters if needed. More generally, the model need not be linear or the topology transitions Markovian, presenting great flexibility.

4.2 Validating the MJLS model

We consider the MJLS model whose parameters α, β, and J are estimated using data from 2011. Figure 4-7 highlights the potential for the model to reflect high-level temporal trends in delay dynamics. For this experiment, the MJLS model is initialized with the airport delays at 8 am EST, and the delays are propagated using Equations 4.6 and 4.7. We see that the MJLS model can capture the varying scale and magnitude of delays across different airports, as well as the temporal trend of increasing and decreasing delays (with a 6.4% error over the first 12 hours). There is a 1- to 2-hour lag in the ability of the model to capture decaying delays, possibly because the model, for simplicity, has just $M = 12$ discrete modes. Although not shown in this figure, the model is also found to track the variance in delay reasonably well.

The validity of the MJLS model can also be demonstrated by noting its predictive performance compared to naive baselines. Table 4.1 compares a linear regression and the MJLS models for predicting delays at a 2-, 4-, and 6-hour horizons. The models use different inputs: the MJLS model uses $\mathbf{x}(t)$ and $m(t)$ as the initial condition, while the linear regression models use t, $\mathbf{x}(t)$, and a characteristic network topology label for the previous day (for potential residual delay effects). The results indicate that the MJLS model performs better than linear regression (as well as simple neural network architectures) while modeling airport delay propagation. We will discuss

Method	**Mean Prediction error (min)**		
	2 hour	4 hour	6 hour
Linear Regression	7.59	9.51	10.47
MJLS model	6.75	8.83	10.09

Table 4.1: Mean prediction errors for the total system delay for the MJLS and a linear regression model across different prediction horizons. The model was developed using data from 2011, and tested on data from 2012. See Chapter 6 for more details.

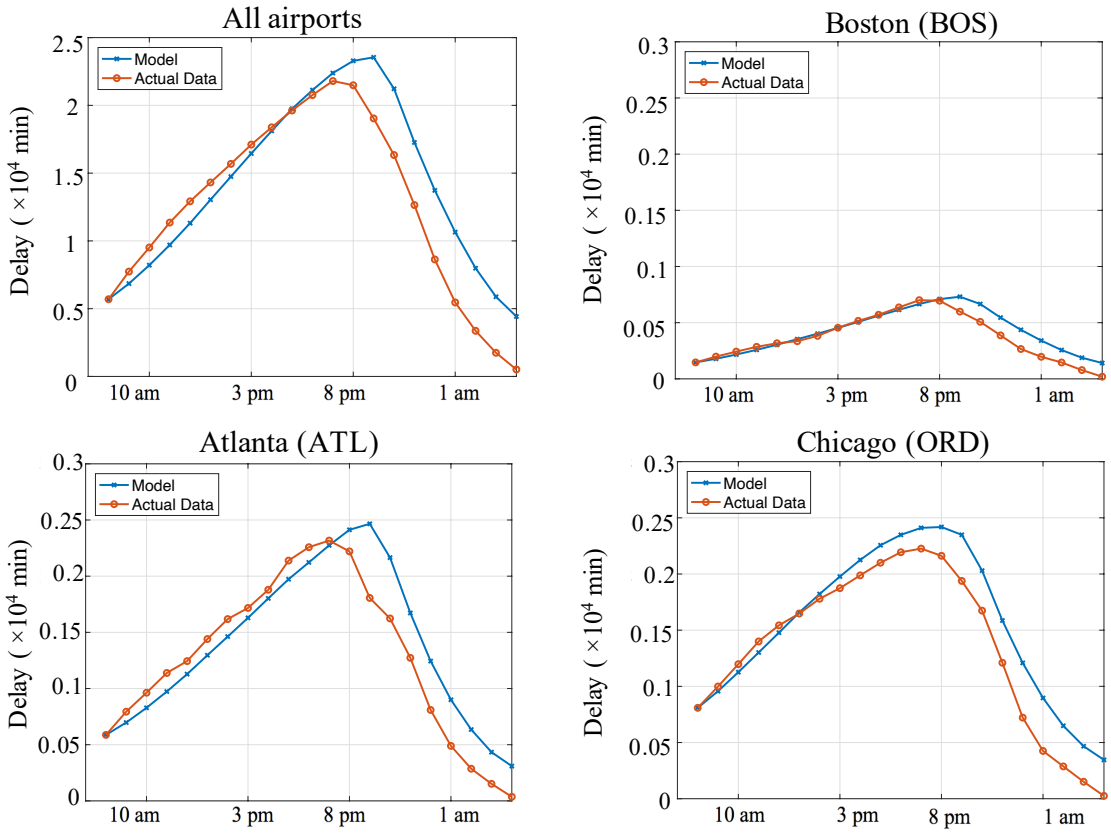


Figure 4-7: MJLS model initialized with the conditions at 8AM EST.

the prediction problem in more detail in the next section.

4.3 Estimating Airport Resilience

We describe how the parameters of the model can be used to quantify the operational resilience of the airports. In particular, we quantify the inertia of an airport to change in its delay state. This metric is complementary to some of the airport-centric measures obtained using the delay-diffusion model (in Chapter 3). The delay-diffusion model was used to quantify the ability of an airport to spread delays under a fixed network topology with respect to pre-determined external inputs. The MJLS model on the other hand attempts to estimate the resilience of airports based on operational data without assuming anything about the network topology or nature of external perturbation.

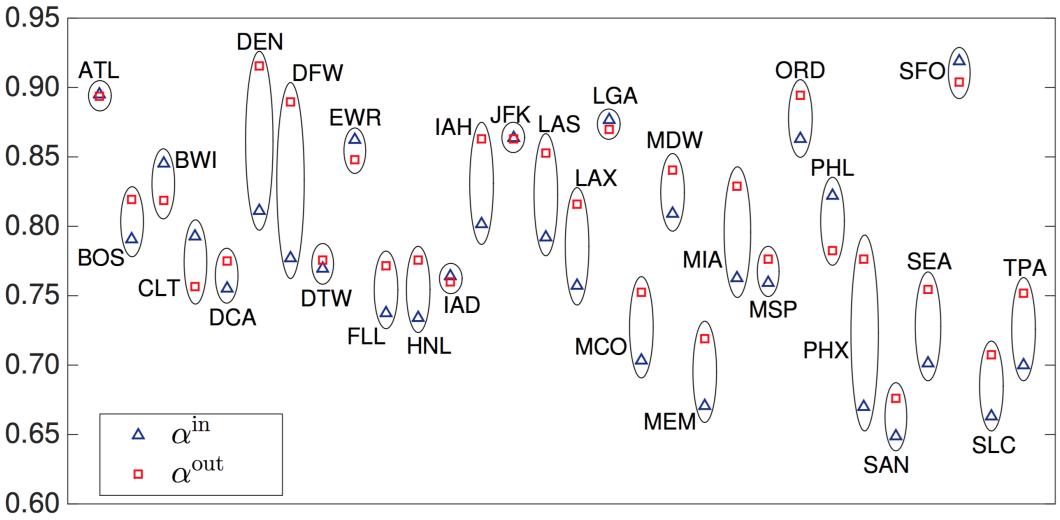


Figure 4-8: Average value of α^{out} and α^{in} across all the discrete modes.

The parameters α^{out} and α^{in} (equations (4.1)-(4.2)) quantify the persistence of out-delays and in-delays at an airport. Figure 4-8 plots the average values of α^{in} and α^{out} for different airports. Several major airports, like Atlanta (ATL), San Francisco (SFO), and Chicago O'Hare (ORD) have among the largest values of these parameters. Denver (DEN) and Dallas/ Fort Worth (DFW) have a high value of α^{out}, but a smaller value of α^{in}, suggesting an asymmetry in the persistence of outbound and inbound delays, with the former being more persistent (and less susceptible to network effects). Smaller airports (e.g., Memphis (MEM), San Diego (SAN) and Salt Lake City (SLC)) have lower values of these parameters, especially with regards to inbound delays. The reason for this phenomenon could be that there is typically sufficient capacity at these airports to

satisfy the levels of demand seen at these airports; as a result, there is little congestion. Higher values of α suggests more persistent delays at the airport, and less susceptibility to delays from other's. On the other hand, low α indicates that the airport is very susceptible to the delay influence of others. Thus, the resilience metric α captures the susceptibility of an airport to network effects.

4.4 Stability of discrete-time positive Markov Jump Linear Systems

Stability analysis provides insights into the temporal and topology-specific patterns that lead to increasing or decreasing delays and also helps further validate the model. For example, one would expect airport delays to decay overnight, as traffic demand decreases. The stability, or instability, of a system may also motivate approaches to effectively control the system, as we will explore further in Chapter 5.

For a discrete-time, linear, time-invariant (LTI) system, stability is defined as the convergence of the sequence of the continuous state norms to 0, and a necessary and sufficient for stability is that the spectral radius of the system matrix is less than 1. The evolution of $\mathbf{x}(t)$ in a MJLS is a stochastic process, and consequently, relevant notions of stability have been proposed for switched systems in general [162, 163], and MJLS in particular [164, 165, 166]. Stability conditions for MJLS are starkly different from those for LTI systems– stability or instability of the individual modes is neither necessary nor sufficient for the stability of the MJLS [165]. In this section, we present conditions for the stability of PMJLS with continuous state resets, and interpret them in the context of air traffic delay networks.

4.4.1 Various Notions of Stability

Definition 4 (Mean Stability (MS))**.** *A system is said to be* mean stable *if the expected value of the state tends to zero as time tends to infinity, i.e,* $\lim_{k\to\infty}\mathbb{E}[\mathbf{x}(k)] = \mathbf{0}$*, for any non-negative initial conditions,* $\mathbf{x}(0)$*, and for any* $m(0)$.

Definition 5 (Exponentially Mean Stability)**.** *A system is said to be exponentially mean stable if*

there exist positive scalars c and $r < 1$, such that $\mathbb{E}[\mathbf{x}(k)] \leq c\, r^k \, \|\mathbf{x}(0)\| \, \mathbf{1}$, for all times k, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.

Definition 6 (Exponential δ-Moment Stability)**.** *A system is said to be exponentially δ-moment stable if there exist positive scalars c and $r < 1$ such that $\mathbb{E}[\|\mathbf{x}(k)\|^\delta] \leq c\, r^k \, \|\mathbf{x}(0)\|^\delta$, for all times k, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.*

Definition 7 (Exponential Mean-Square Stability)**.** *A system is said to be exponentially mean-square stable if there exist positive scalars c and $r < 1$ such that $\mathbb{E}[\|\mathbf{x}(k)\|^2] \leq c\, r^k \, \|\mathbf{x}(0)\|^2$, for all times k, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.*

Definition 8 (Almost Sure Stability (ASS))**.** *A system is said to be almost surely stable if the state tends to zero as time tends to infinity, with probability 1, i.e., $\mathbb{P}[\lim_{k\to\infty} \|\mathbf{x}(k)\| = 0] = 1$, for any non-negative initial conditions, $\mathbf{x}(0)$, and for any $m(0)$.*

Remarks:

1. The notion of MS was specifically introduced to exploit the non-negativity property of PMJLS, to obtain stability criteria [50, 167].

2. The concept of ASS requires that *every* realization of the system state tends to zero, rather than just tending to zero in expectation. This definition is typically more useful in practice [168].

4.4.2 Background

The stability of a MJLS has been extensively studied [165, 166]. As the evolution of the state vector $x(t)$ is stochastic, characterizing stability is non-trivial and several definitions exist. Necessary and sufficient conditions for one definition, Mean Square Stability, have been identified [164, 165]. Mean Square Stability requires the expected 2-norm of the state vector to go to 0 as $t \to \infty$. In [168], the authors argue that defining stability based on every single realization is more useful, and introduce the notion of Almost Sure Stability. Relations between the different definitions are presented in [164] for the continuous-time version of the MJLS model. They are:

- Exponential Mean Square Stability $\iff$ Mean Square Stability

- Mean Square Stability $\Rightarrow$ Almost Sure Stability

- If $\delta_1 \geq \delta_2$, Exponential δ_1-Moment Stability $\Rightarrow$ Exponential δ_2-Moment Stability. Note that $\delta = 2$ gives Exponential Mean Square Stability.

Several sufficient conditions, with varying degrees of conservatism, have been proposed for Almost Sure Stability, but a necessary and sufficient condition has been elusive. The current approach would be to either use a conservative sufficient-condition or perform a Monte Carlo simulation to check stability for a given confidence level [169].

Recently, Markov Jump Linear Systems where the state vector always remains non-negative for a non-negative initial condition (meaning $x(t) \geq 0$ if $x(0) \geq 0$), have received attention. These are called Positive Markov Jump Linear Systems, or PMJLS, and are used to model a wide variety of physical systems (including air traffic delays). For these positive systems, [50] introduces yet another definition of stability, namely the Mean Stability, and exploits the non-negativity to prove its equivalence to 1-Moment stability. The results of [50] are presented for the continuous-time case, and limited work exists on extending them for the discrete case [167]. In the rest of this section, we will extend the results of [50] for the case of discrete-time PMJLS, and consider the special case when we have periodic transition matrices.

4.4.3 Stability of a PMJLS

We begin with the mathematical preliminaries needed for our results on the stability of the PMJLS defined by Equations 4.6-4.8.

Definition 9. *The augmented state vector* $\mathbf{q}(t) \in \mathbb{R}^{NM\times 1}$ *given by*

$$\mathbf{q}(t) = \begin{bmatrix} \mathbf{q}_1(t), & \cdots, & \mathbf{q}_M(t) \end{bmatrix}^\mathsf{T}, \text{ where } \mathbf{q}_i(t) = \mathbb{E}[\mathbf{x}(t)\,\mathbb{1}_{m(t)=i}] \text{ and } \mathbf{q}_i(t) \in \mathbb{R}^{N\times 1}. \tag{4.9}$$

Definition 10. *We define an augmented state transition matrix* $B_t \in \mathbb{R}^{NM \times NM}$ *as*

$$B_t = \begin{bmatrix} \pi_{11}(t)J_{1,1}\Gamma_1 & \dots & \pi_{M1}(t)J_{M,1}\Gamma_M \\ \vdots & \ddots & \vdots \\ \pi_{1M}(t)J_{1,M}\Gamma_1 & \dots & \pi_{MM}(t)J_{M,M}\Gamma_M \end{bmatrix} \tag{4.10}$$

The following propositions describe the evolution of $\mathbf{q}(t)$ and $\mathbb{E}[\mathbf{x}(t)]$:

Proposition 1. *The expected state* $\mathbb{E}[\mathbf{x}(t)]$ *is equal to* $\sum_{i=1}^{M} \mathbf{q}_i(t)$.

Proof.

$$\mathbb{E}[\mathbf{x}(t)] = \sum_{j=1}^{M} \mathbb{E}[\mathbf{x}(t)\,\mathbb{1}_{m(t)=j}] = \sum_{j=1}^{M} \mathbf{q}_j(t) \tag{4.11}$$

□

Proposition 2. *Given* $\mathbf{x}(t)$ *and* $m(t)$, $\mathbf{q}_i(t+1) = \pi_{m(t),i}(t)J_{m(t),i}\Gamma_{m(t)}\mathbf{x}(t)$.

Proof.

$$\begin{aligned} \mathbf{q}_i(t+1) &= \mathbb{E}[\mathbf{x}(t+1)\,\mathbb{1}_{m(t+1)=i}] \\ &= \Pr[m(t+1)=i|m(t)]\;\mathbb{E}[\mathbf{x}(t+1)|m(t)] \\ &= \pi_{m(t),i}(t)J_{m(t),i}\Gamma_{m(t)}\mathbf{x}(t) \end{aligned}$$

□

Proposition 3. *The augmented state vector evolves as* $\mathbf{q}(t+1) = B_t\mathbf{q}(t)$.

Proof. For any $j \in \{1,\dots,M\}$,

$$\begin{aligned} \mathbf{q}_j(t+1) &= \mathbb{E}[\mathbf{x}(t+1)\,\mathbb{1}_{m(t+1)=j}] \\ &= \mathbf{x}(t+1)|_{m(t+1)=j}\Pr[m(t+1)=j] \\ &= \sum_{i=1}^{M} J_{ij}\Gamma_i\,\mathbf{x}(t)|_{m(t)=i}\,\pi_{ij}\Pr[m(t)=i] \\ &= \sum_{i=1}^{M} \pi_{ij}(t)J_{ij}\Gamma_i\mathbf{q}_i(t). \end{aligned}$$

Writing the above equation in vector notation and using Definition 10, we get the desired result.

□

These propositions indicate that knowledge of the current state, either in the form of $(\mathbf{x}(t), m(t))$ or $\mathbf{q}(t)$ is sufficient to propagate the augmented state and the expected state $\mathbb{E}[\mathbf{x}(t)]$ for future time. We now present some results that relate the various notions of stability and provide conditions for stability when applicable. Two cases will be considered for further analysis.

Case 1: Constant Mode Transition Probability

The first case will assume that the mode transition probabilities do not vary with time, i.e., $\pi_{i,j}(t) = \pi_{i,j} \ \forall t$. Thus, $B_t = B \, \forall t$.

Theorem 1. *If $\rho(B) < 1$, then the system is Exponentially Mean Stable*

Proof.

$$\begin{aligned}
\mathbb{E}[\mathbf{x}(t)] &= \sum_{i=1}^{M} \mathbf{q}_i(t) \\
&\leq M \max_i ||\mathbf{q}_i(t)|| \, \mathbf{1} \quad (\mathbf{q}_i(t) \geq \mathbf{0} \text{ for a PMJLS}) \\
&\leq C \, ||\mathbf{q}(t)|| \, \mathbf{1} \qquad \text{(since all elements are non-negative)} \\
&= C \, ||B^t \mathbf{q}(0)|| \, \mathbf{1} \qquad \text{(Special case of Proposition 3)} \\
&\leq C \, ||\rho(B)^t \mathbf{q}(0)|| \, \mathbf{1} \qquad \text{(properties of the spectral radius)} \\
&\leq C \, \rho(B)^t \, ||\mathbf{q}(0)|| \, \mathbf{1} \\
&\leq C \, r^t \, ||\mathbf{q}(0)|| \, \mathbf{1} \qquad (\text{where } r = \rho(B))
\end{aligned}$$

□

Theorem 2. *If a system is Exponentially Mean Stable, then it is Mean Stable*

Proof. By definition of a PMJLS, $\mathbf{x}(t) \geq \mathbf{0} \, \forall t$. This means that

$$\lim_{t \to \infty} \mathbb{E}[\mathbf{x}(t)] \geq \mathbf{0} \tag{4.12}$$

Also, since the system is EMS, we have

$$\begin{aligned}
\mathbb{E}[\mathbf{x}(t)] &\leq C\beta^t ||\mathbf{x}(0)||\mathbf{1} \\
\lim_{t\to\infty} \mathbb{E}[\mathbf{x}(t)] &\leq \lim_{t\to\infty} C\beta^t ||\mathbf{x}(0)||\mathbf{1} \qquad \text{(taking limits)} \\
&= 0 \qquad (\beta < 1 \text{ by definition})
\end{aligned}$$

Combining the last statement with Equation 4.12, we get $\lim_{t\to\infty} \mathbb{E}[\mathbf{x}(t)] = \mathbf{0}$ □

We have shown using Theorem 2 and 1 that $\rho(B) < 1$ is sufficient for EMS and MS. In the subsequent theorem, we show that this is also a necessary condition.

Theorem 3. *The following statements are true:*
(i) If the PMJLS is Mean Stable, then $\rho(B) < 1$
(ii) If the PMJLS is Exponential Mean Stable, then $\rho(B) < 1$

Proof. We prove this by contradiction. Suppose $\rho(B) \geq 1$. We know that $\mathbf{q}(t) = B^t\mathbf{q}(0)$. Since the spectral radius of B is greater than or equal to 1, there exists some vector $\mathbf{q}(0) > \mathbf{0}$ such that all subsequent augmented states will be non-decreasing, i.e., $\mathbf{q}(t_1) \geq \mathbf{q}(t_2)\ \forall t_1 \geq t_2$. Since $\mathbf{q}(t)$ is a non-decreasing vector, atleast one element of $\mathbf{E}[x(t)] = \sum_{i=1}^{M} q_i(t)$ will be positive, and non-decreasing. This implies that the system cannot not satisfy $\lim_{t\to\infty} \mathbb{E}[\mathbf{x}(t)] = \mathbf{0}$ or $\mathbb{E}[\mathbf{x}(t)] \leq C\beta^t ||\mathbf{x}(0)||\mathbf{1}$. This contradicts the statement that the PMJLS is Mean Stable (or Exponentially Mean Stable). Hence proved by contradiction. □

The subsequent remark follows directly from Theorem 2, 3 and 1:

Remark 1. *The following statements are equivalent*
(i) $\rho(B) < 1$
(ii) The system is Mean Stable
(iii) The system is Exponentially Mean Stable

Theorem 4. *The following two statements are equivalent*
(i) The system is Exponentially Mean Stable
(ii) The system is Exponentially 1-Moment Stable

Proof. First we show that $(ii) \Rightarrow (i)$. For any non-negative vector, $\mathbf{x}(t) \leq ||\mathbf{x}(t)||\mathbf{1}$. Hence $\mathbb{E}[\mathbf{x}(t)] \leq \mathbb{E}[||\mathbf{x}(t)||]\mathbf{1}$. Using the fact that the system is Exponentially 1-Moment stable, we have

$$\mathbb{E}[\mathbf{x}(t)] \;\leq\; \mathbb{E}[||\mathbf{x}(t)||]\mathbf{1} \;\leq\; C\beta^t||\mathbf{x}(0)||\mathbf{1} \tag{4.13}$$

Taking limits as $t \to \infty$ and using the non-negativity of $\mathbb{E}[\mathbf{x}(t)]$ (PMJLS definition), we have $\lim_{t\to\infty}\mathbb{E}[\mathbf{x}(t)] = \mathbf{0}$.

Now we show $(i) \Rightarrow (ii)$. For a positive vector $\mathbf{x}(t)$ and any k-norm,

$$||\mathbf{x}(t)|| \;=\; (\sum_{i=1}^{N} x_i(t)^k)^{\frac{1}{k}} \;\leq\; [(\sum x_i(t))^k]^{\frac{1}{k}} \;\leq\; \sum_{i=1}^{N} x_i(t) \tag{4.14}$$

where the first inequality comes from completing the square. Hence,

$$\mathbb{E}[||\mathbf{x}(t)||] \;\leq\; \mathbb{E}[\sum_{i=1}^{N} x_i(t)] \tag{4.15}$$

Since the system is Exponentially Mean Stable, every element of the state vector is exponentially bounded, i.e $\mathbb{E}[x_i(t)] \leq C\beta^t||\mathbf{x}(t)||$. Consequently, $\mathbb{E}[\sum_{i=1}^{N} x_i(t)] \leq NC\beta^t||\mathbf{x}(t)||$. Substituting this in Equation (4.15), we get $\mathbb{E}[||\mathbf{x}(t)||] \leq \tilde{C}\beta^t||\mathbf{x}(0)||$ □

Theorem 5. *If a system is Exponential 1-Moment Stable, it is Almost Surely Stable*

Proof. $||\mathbf{x}(t)||$ is a non negative random variable. Using Markov's inequality, for any $\varepsilon > 0$, we have

$$\mathbb{P}(||\mathbf{x}(t)|| > \varepsilon) \leq \frac{\mathbb{E}[||\mathbf{x}(t)||]}{\varepsilon} \leq \frac{C\beta^t||\mathbf{x}(0)||}{\varepsilon}$$

where the second inequality comes from the definition of Exponential 1-Moment Stability. Using the property that all probabilities are non-negative, we have

$$0 \leq \mathbb{P}(||\mathbf{x}(t)|| > \varepsilon) \leq \frac{C\beta^t||\mathbf{x}(0)||}{\varepsilon} \tag{4.16}$$

$$\implies \; 0 \leq 1 - \mathbb{P}(||\mathbf{x}(t)|| \leq \varepsilon) \leq \frac{C\beta^t||\mathbf{x}(0)||}{\varepsilon} \tag{4.17}$$

$$\implies \; 1 - \frac{C\beta^t||\mathbf{x}(0)||}{\varepsilon} \leq \mathbb{P}(||\mathbf{x}(t)|| \leq \varepsilon) \leq 1 \tag{4.18}$$

Taking limits as $t \to \infty$ and $\varepsilon \to 0$ and using L'Hospitals rule, we have the desired result. $\square$

For the case when $B_t = B$, our results are summarized below:

1. $\rho(B) < 1$ is a necessary and sufficient condition for Mean Stability [Remark 1]

2. $\rho(B) < 1$ is a sufficient condition for Almost Sure Stability [Remark 1, Theorem 4 and 5]

Case 2: Periodic Mode Transition Probabilities

The second case considers the situation where the transition matrices are periodic. This is relevant when modeling infrastructure systems such as the air traffic network, which exhibit strong time-of-day patterns.

Definition 11. *The Markovian transition matrix $\Pi_t = [\pi_{ij}(t)]$ is defined to be periodic with fundamental period $K \in \mathbb{Z}^+$ if*

$$\pi_{i,j}(t+nK) = \pi_{i,j}(t),\ \forall\, n \in \mathbb{Z},\ i,j \in \{1,\ldots,M\}. \tag{4.19}$$

For systems such as the air traffic network that exhibit daily temporal patterns, $K = 24$ (assuming that each time-step is 1 hour long).

The analysis of systems with periodic transition matrices follows a similar style as before. Suppose we denote $D_0 = B_{K-1}B_{K-2}\cdots B_0$. Then, the dynamics of the augmented state vector $\mathbf{q}(t)$ is now given by:

$$\mathbf{q}(k+nK) = B_{k-1}\cdots B_0\ \mathbf{q}(nK), \forall\, n \in \mathbb{Z}^+; k \in [0, K-1] \tag{4.20}$$

We can now bound $\|\mathbf{q}(nK)\|$ as follows:

$$\begin{aligned} & \mathbf{q}(K) && = D_0\,\mathbf{q}(0) \\ \implies\ & \mathbf{q}(nK) && = {D_0}^n\,\mathbf{q}(0), \forall\, n \in \mathbb{Z}^+ \\ \implies\ & \|\mathbf{q}(nK)\| && \leq \rho(D_0)^n\,\|\mathbf{q}(0)\|, \end{aligned} \tag{4.21}$$

The flavor of all the subsequent analysis is similar to the case with a constant transition matrix. The resultant Theorems (without proof) are presented below.

Theorem 6. *If the transition probabilities are time-varying and periodic with length K, i.e.,* $\pi_{ij}(t+K) = \pi_{ij}(t)$*, then the PMJLS is Mean Stable if and only if the spectral radius of the matrix* $D_k = B_{k+K-1}B_{k+K-2}\cdots B_k$ *is less than 1, for some* $k \in [0,K]$.

Theorem 7. *If the transition probabilities are time-varying and periodic with length K, then* $\rho(D_k) < 1$*, for some* $k \in [0,K]$ *is sufficient for the Almost Sure Stability of the PMJLS.*

Theorem 1 and 6 provide a test for MS based on the spectral radius of a matrix. These are easy to compute, and can also be easily performed for large scale systems. It is also worth remembering that these are sufficient and not necessary for ASS, i.e., if the conditions for MS are not satisfied, it does not tell us anything about ASS.

4.5 The Asymptotic Evolution of Airport Delays

The airport delay MJLS model involves periodic transitions (with time period $K = 24$) between 12 discrete modes, some of which are stable, while others are unstable (Figure 4-9).

To evaluate stability of the resulting PMJLS, we compute the spectral radius of D_0 as $\rho(D_0) = 0.67 < 1$, to conclude that the system is MS, and therefore ASS (Theorem 6). This finding is consistent with our expectation that delays will tend to decay over the course of a 24-hour period in expectation, and further validates the MJLS model. Interestingly, if we had not incorporated the time-varying transition matrix and instead considered the average matrix $\Pi_t = \Pi \; \forall t$, then $\rho(B) = 1.06$, and by Remark 1 the system would not be MS. This result highlights the critical role that the discrete mode transitions to 'Low NAS decreasing' (and other decaying delay modes) during the later part of the day play in stabilizing the system.

The spectral radius of B_t gives an upper bound on the amplification (i.e., the worst-case magnification) of the total system delay at time t, over the next time-step (an hour, in our case). Figure 4-10, which plots $\rho(B_t)$ as a function of time indicates that the topology transitions are such that the delays tend to decay after 8 pm, thus resulting in low airport delays overnight. Additionally, from Proposition 3, we can also show that the worst-case amplification of delays from t_1 to t_2,

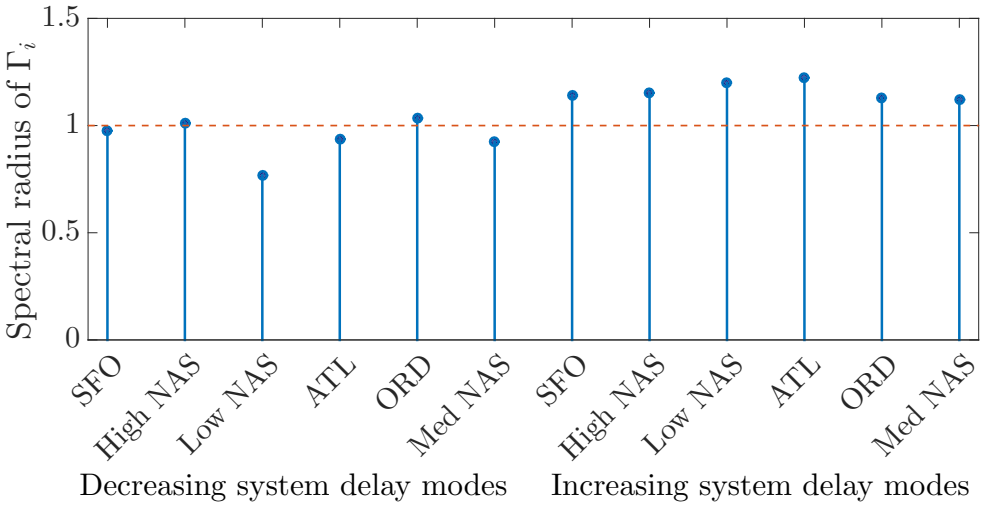


Figure 4-9: Spectral radius of the individual modes of the MJLS model for airport delays.

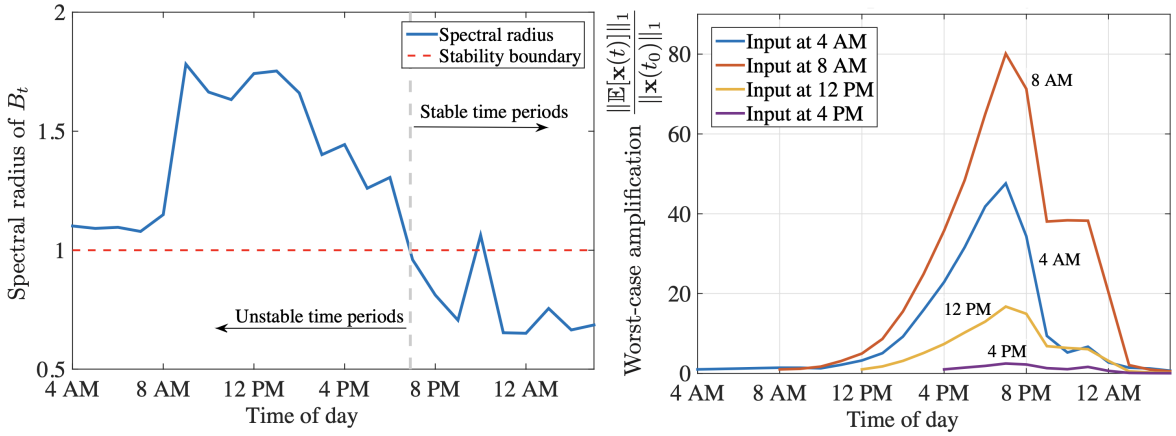


Figure 4-10: (Left) Worst-case delay magnification (over the next hour) of 1 unit of total delay initiated at a particular hour of the day. (Right) Worst-case amplification of 1-unit of time delay injected at four different times (4 am, 8 am, 12 pm and 4 pm), when allowed to propagate. All times are in US EST.

$\frac{\|\mathbb{E}[\mathbf{x}(t_2)]\|_1}{\|\mathbf{x}(t_1)\|_1}$, is equal to $\rho(B_{t_2-1}\ldots B_{t_1+1}B_{t_1})$. Using this result, we plot the worst-case amplification of a 1-unit total delay initialized at different points in time (4 am, 8 am, 12 pm and 4 pm) in Figure 4-10. Delays initiated earlier in the day have a greater potential to cascade throughout the system and get magnified (up to 80 times by 8 pm) as compared to those introduced during the later part of the day. Our analysis emphasizes the need to ensure disruption-free operations during the early part of the day for improved system performance – an operational practice regularly employed by airline operations managers and air traffic controllers.

Our results illustrate the role that stability analysis plays in model validation and in identifying key aspects of the delay dynamics. Finally, we point readers towards recent work on structural [170] and finite-time [171] stability, which have yielded interesting insights.

Chapter 5

Control of Switched Network Models

Long-term strategies for reducing air traffic delays require expensive infrastructure and technological investments. Tactical control approaches, at the time-frame of a few hours, such as traffic flow management (e.g., ground delay programs or airspace flow programs resulting in flight swaps and cancellations) can be effective in mitigating congestion due to unforeseen disruptions [172]. In the context of our MJLS model for air traffic delays, these actions correspond to either interventions on the nodal (continuous) state to decrease delays at airports, or interventions on the network topology to decrease the propagation of delays. Figure 5-1 shows an example of how flight cancellations are used to reduce demand, and consequently, reduce delays for the remaining flights. Figure 5-2 shows how GDP issuance correlates with the delay network topology, and suggests how GDPs can be used to induce delay topology transitions. The development of control strategies for such networked systems will enable efficient, resilient and robust operations in several other critical infrastructures [173, 174, 175].

Prior work: The control of processes on networks has been well-studied in the context of epidemiology [156, 144] and resilient infrastructures [176, 177, 178]. However, controllers for systems with switching topologies have received less attention [165, 179]. The most popular approach to control the nodal state of a MJLS is to use a mode-dependent state feedback controller [165, 180]. Dynamic programming [181], convex programming [182], particle-based approximate methods [183], and receding horizon controllers [184, 185] have also been developed. These controllers all modify the nodal state $\mathbf{x}(t)$ to achieve the desired control objective. The control of mode transitions, by which one alters the transition probability Π_t or induces specific mode transitions,

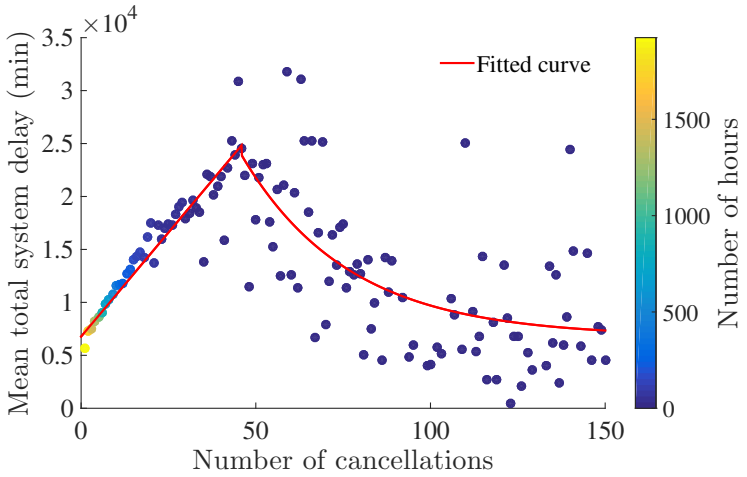


Figure 5-1: Mean system delay vs. number of flight cancellations, for hourly data from 2011-12 grouped by the number of flight cancellations. The color of each dot denotes the number of hours for each of the groups. While delays and cancellations both increase initially, cancellations are used to strategically reduce traffic demand when disruptions are severe, resulting in lower delays for flights that continue to operate.

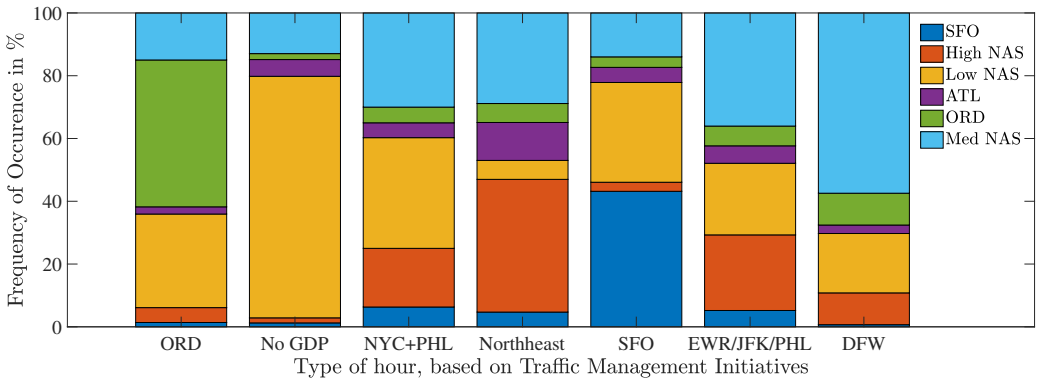


Figure 5-2: Frequency of occurrence of delay-networks for each GDP type. 8 GDP and 6 delay type-of-hours are obtained via the clustering technique described in Chapter 2. Observe that each GDP type corresponds to a higher occurrence of a specific delay network topology.

has been impractical in many traditional settings, and therefore not been explored [186]. However, increasing autonomy has enabled the ability to modify the structure of network interactions in many infrastructure systems.

Key Contributions: We list the main contributions of this chapter.

1. We develop optimal node and topology controllers for PMJLS models. These controllers leverage the positive-state property of the system to identify actions that minimize the expected cost-to-go for a finite time horizon. We finally merge the two controllers to developed an integrated controller that can identify optimal combinations of node and topology actions.

2. We implement these controllers on the airport delay PMJLS model to (i) demonstrate the superior performance of the integrated controller, and (ii) provide potential strategies for system operators to optimally recover from delay disruptions.

Chapter Outline: We present the problem formulation in Section 5.1, main results in Section 5.2, experimental evaluation on airport delay models in Section 5.3, and further extensions in Section 5.4.

5.1 Problem setup

Let the continuous state of the system be $\mathbf{x}(t)$, and the discrete mode be $m(t)$ at time t. The objective of a controller is to optimally trade off the control effort with the benefits of a smaller continuous state $\mathbf{x}(t)$. Since $\mathbf{x}(t) \geq 0$ for a PMJLS, it is reasonable to use the 1-norm for continuous state magnitude. The control action at time t minimizes the cost-to-go, and the objective function for the controller is expressed as

$$\text{Minimize} \quad \mathbb{E}[\sum_{\tau=t}^{T} \|\mathbf{x}(\tau)\|_1] + \text{ control penalty } (t) \tag{5.1}$$

We consider three different controllers:

1. **Node or continuous state control.** The control action $\mathbf{u}(t) \in \mathbb{R}^{N\times 1}$ modifies the continuous state as $\mathbf{x}(t+1) = \Gamma_{m(t)}\mathbf{x}(t) - \mathbf{u}(t)$, while ensuring that $\mathbf{x}(t) \geq \mathbf{0}$ and $\mathbf{0} \leq \mathbf{u}(t) \leq U_{max}\mathbf{1}$. The action $\mathbf{u}(t)$ incurs a control penalty of $\beta^T(t)\mathbf{u}(t)$, where $\beta(t) \in \mathbb{R}^{N\times 1}_{\geq 0}$ is the cost per unit action on each node.

2. **Topology or mode control** A mode control action forces a transition to a particular discrete mode at time $t+1$, instead of allowing it to be drawn from the distribution Π_t. The control penalty for forcing a transition from mode i at time t to mode j at $t+1$ is given by $\alpha_{i,j}(t) \in \mathbb{R}_{\geq 0}$. There is no penalty in the absence of a forced transition.

3. **Integrated control** An integrated control action involves a combination of continuous state and mode control.

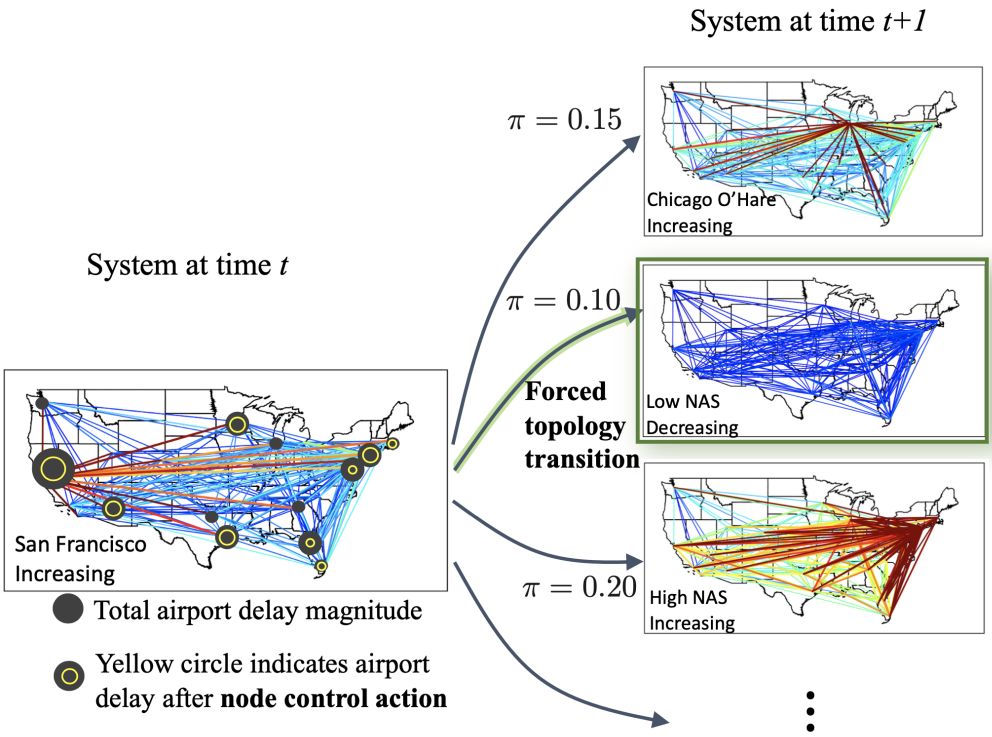


Figure 5-3: Illustration of the possible control actions: nodal intervention, and forced topology transition.

A visualization of these three controllers is presented in Figure 5-3. In the next section, we will compute policies for all these three controllers.

5.2 Optimal Control Policy

In our analysis, we ignore the state reset matrices $J_{i,j}$ for simplicity; however, the results can be trivially extended to incorporate them. We will start with some preliminary definitions.

Definition 12. *We define the matrix* $S = [s_{ij}] \in \mathbb{R}^{N \times NM}$ *as*

$$s_{ij} = \begin{cases} 1, \text{ if } j = i + kN, \text{ for } k \in \mathbb{N} \\ 0, \text{ otherwise} \end{cases} \tag{5.2}$$

Proposition 4. $\mathbb{E}[\mathbf{x}(t)] = S\mathbf{q}(t)$

Proof.

$$\begin{aligned} \mathbb{E}[\mathbf{x}(t)] &= \sum_{i=1}^{M} \mathbb{E}[\mathbf{x}(t)\, \mathbb{1}_{m(t)=i}] \\ &= \sum_{i=1}^{M} \mathbf{q}_i(t) \\ &= S\mathbf{q}(t) \end{aligned}$$

□

Definition 13. *We define* $C_{t_1,t_2} \in \mathbb{R}^{N \times NM}$ *as*

$$C_{t_1,t_2} = S(\mathrm{I}_{MN \times MN} + B_{t_1} + B_{t_1+1}B_{t_1} + \ldots + B_{t_2-1} \cdots B_{t_1}), \tag{5.3}$$

Definition 14. *We define the vector* $\mathbf{c}_{ij} \in \mathbb{R}^{N \times 1}$, $i = 1, \ldots, N$, *and* $j = 1 \ldots, M$ *in terms of the sub-matrices created from* C_{t_1,t_2} *as*

$$C_{t_1,t_2} = \begin{bmatrix} \mathbf{c}_{11}^T & \cdots & \mathbf{c}_{1M}^T \\ \vdots & \ddots & \vdots \\ \mathbf{c}_{N1}^T & \cdots & \mathbf{c}_{NM}^T \end{bmatrix} \tag{5.4}$$

Definition 13 is used to simplify the first term of the objective function in Equation 5.1. This gives us the following proposition:

Proposition 5. *The expected sum of future states* $\sum_{t=t_1}^{t_2} \mathbb{E}[\mathbf{x}(t)]$ *is equal to* $C_{t_1,t_2}\mathbf{q}(t_1)$.

Proof. From Proposition 3, we have $\mathbf{q}(t+1) = B_t\mathbf{q}(t)$. Therefore, for any $t > t_1$, we have

$$\mathbb{E}[\mathbf{x}(t)] = S\mathbf{q}(t) = SB_{t-1}\dots B_{t_1}\mathbf{q}(t_1) \tag{5.5}$$

Therefore, we have

$$\begin{aligned}
\sum_{t=t_1}^{t_2} \mathbb{E}[\mathbf{x}(t)] &= \mathbb{E}[\mathbf{x}(t_1)] + \mathbb{E}[\mathbf{x}(t_1+1)] + \ldots + \mathbb{E}[\mathbf{x}(t_2)] \\
&= S\mathbf{q}(t_1) + SB_{t_1}\mathbf{q}(t_1) + \ldots + B_{t_2-1}\cdots B_{t_1}\mathbf{q}(t_1) \quad \text{(using Equation 5.5)} \\
&= S(\mathrm{I}_{MN\times MN} + B_{t_1} + B_{t_1+1}B_{t_1} + \ldots + B_{t_2-1}\cdots B_{t_1})\mathbf{q}(t_1) \\
&= C_{t_1,t_2}\mathbf{q}(t_1) \quad \text{(using Definition 13)}
\end{aligned}$$

□

Proposition 6. $\|\mathbb{E}[x(t)]\|_1 = \mathbb{E}[\|x(t)\|_1] = \|SB_{t-1}B_{t-2}\dots B_{t-k}\mathbf{q}(t-k)\|_1$

Proof. Since $\mathbf{x}(t)$ evolves as a PMJLS, $\mathbf{x}(t) \geq \mathbf{0}$ for all t. This means that the 1-norm is simply the sum of the elements, and no absolute value operator is required. Thus, using linearity of the expectation operator and Proposition 3, we get the desired result. □

Proposition 7. $\mathbb{E}[\sum_{\tau=t_1}^{t_1} \|\mathbf{x}(\tau)\|_1] = \|C_{t_1,t_2}\ \mathbf{q}(t_1)\|_1$

Proof. Follows directly from Propositions 5 and 6 □

We present the optimal node, topology and integrated control actions that should be taken when the system has a continuous state $\mathbf{x}(t)$, and discrete mode $m(t)$, at time t, in order to minimize the objective function in Equation 5.1. A quick overview of the main results: Theorem 8 presents the node controller, Theorem 9 presents the topology controller, and Theorem 10 presents the integrated controller.

Theorem 8 (Node control)**.** *The node control action* $\mathbf{u}^*(t)$ *given by*

$$u_i^*(t) = \begin{cases} \max\{U_{max},\ \sum_j [\Gamma_{m(t)}]_{ij} x_j(t)\} & \text{if } \beta_i(t) \le \eta_i \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where $\eta^T = \sum_{i,j} \pi_{m(t),j}(t)\mathbf{c}_{ij}^T \in \mathbb{R}^{1\times N}$ *and* $C_{t+1,T}^{(\pi)} = [\mathbf{c}_{ij}^T]_{i=1:N,\ j=1:M}$ *minimizes the function in Equation 5.1. The minimum value is equal to*

$$(\mathbf{1}^T + \eta^T \Gamma_{m(t)})\mathbf{x}(t) + (\beta^T(t) - \eta^T)\mathbf{u}^*(t) \tag{5.7}$$

Proof. Since a node control action $\mathbf{u}(t)$ only affects the state at time $t+1$, we have

$$\begin{aligned} & \min \quad \mathbb{E}[\sum_{\tau=t}^{T} \|\mathbf{x}(\tau)\|_1] + \beta^T \mathbf{u}(t) \\ = \ & \min \quad \mathbb{E}[\sum_{\tau=t+1}^{T} \|\mathbf{x}(\tau)\|_1] + \beta^T \mathbf{u}(t) \\ = \ & \min \quad \left\| C_{t+1,T}\ \mathbf{q}(t+1) \right\|_1 + \beta^T \mathbf{u}(t) \end{aligned}$$

We know that

$$\begin{aligned} \mathbf{q}_i(t+1) &= \mathbb{E}[\mathbf{x}(t+1)\,\mathbb{1}_{m(t+1)=i}] \\ &= \mathbb{E}[\mathbf{x}(t+1) \mid m(t+1) = i, m(t))]\mathbb{P}[m(t+1) = i \mid m(t)] \\ &= [\Gamma_{m(t)}\mathbf{x}(t) - \mathbf{u}(t)]\pi_{m(t),i}(t) \end{aligned}$$

Substituting the expression for $\mathbf{q}(t+1)$ in the first term of the objective function, we get

$$
\begin{aligned}
& \left\| C_{t+1,T}\ \mathbf{q}(t+1) \right\|_1 \\
= & \left\| \begin{bmatrix} \mathbf{c}_{11}^T & \cdots & \mathbf{c}_{1M}^T \\ \vdots & \ddots & \vdots \\ \mathbf{c}_{N1}^T & \cdots & \mathbf{c}_{NM}^T \end{bmatrix} \begin{bmatrix} (\Gamma_{m(t)}\mathbf{x}(t) - u(t))\pi_{m(t),1}(t) \\ \vdots \\ (\Gamma_{m(t)}\mathbf{x}(t) - u(t))\pi_{m(t),i}(t) \\ \vdots \end{bmatrix} \right\|_1 \\
= & \left\| \mathbf{z} - \begin{bmatrix} \mathbf{c}_{11}^T & \cdots & \mathbf{c}_{1M}^T \\ \vdots & \ddots & \vdots \\ \mathbf{c}_{N1}^T & \cdots & \mathbf{c}_{NM}^T \end{bmatrix} \begin{bmatrix} \pi_{m(t),1}(t)\mathbf{u}(t) \\ \vdots \\ \pi_{m(t),i}(t)\mathbf{u}(t) \\ \vdots \end{bmatrix} \right\|_1 \quad \text{where } \mathbf{z} \geq \mathbf{0} \text{ does not depend on } \mathbf{u}(t) \\
= & \|\mathbf{z}\|_1 - \left\| \left[\sum_{i,j} \mathbf{c}_{ij}^T \pi_{m(t),j}(t) \right] \mathbf{u}(t) \right\|_1 \\
= & \|\mathbf{z}\|_1 - \eta^T \mathbf{u}(t) \quad \text{where } \eta^T = \sum_{i,j} \pi_{m(t),j} \mathbf{c}_{ij}^T
\end{aligned}
$$

This reduces the optimization problem to a variable-separable linear program.

$$
\begin{aligned}
\min \quad & (\beta^T - \eta^T)\mathbf{u}(t) && (5.8) \\
\textit{such that} \quad & 0 \leq \mathbf{u}(t) \leq \mathbf{1}_{N\times 1} U_{max} && (5.9) \\
& \mathbf{u}(t) \leq \Gamma_{m(t)}\mathbf{x}(t) && (5.10)
\end{aligned}
$$

This gives the desired optimal control action $\mathbf{u}^*(t)$. Using this optimal control action and $\mathbf{z}$, we get the optimal objective function value. □

The optimal node control policy is a threshold rule that identifies nodes at which intervention is justified, and uses the largest control action subject to the limit of U_{max}, and such that $\mathbf{x}(t+1) \geq \mathbf{0}$. Note that $\mathbf{u}^*(t) = \mathbf{0}$ if no action can decrease the objective function.

Proposition 8. *When the topology is forced to transition to mode k at time $t+1$, the value of the*

objective function (Equation 5.1) is

$$\|\mathbf{x}(t)\|_1 + \left\| C_{t+1,T}\, \mathbf{q}^{(k)}(t+1) \right\|_1 + \alpha_{m(t),k}(t) \tag{5.11}$$

where $\mathbf{q}_i^{(k)}(t+1) = \begin{cases} \Gamma_{m(t)}\mathbf{x}(t), & \text{if } i = k \\ \mathbf{0}, & \text{otherwise} \end{cases}$ *for* $i \in \{1,\ldots,M\}$

Proof. Separate the terms in Equation 5.1, and simplify using Propositions 2 and 3. □

Theorem 9 (Topology control)**.** *The optimal topology control action chooses the mode that yields the lowest objective function value (Equation 5.1) from among the following options:*

(i) No topology control, with objective function value given by Proposition 5.

(ii) Topology transition to any mode $k \in \{1,\ldots,M\}$*, with objective function value given by Proposition 8.*

Proof. The cases (i) and (ii) decompose the domain of the optimization. Hence, the global minimum is the best solution among these locally optimal solutions. □

The optimal topology control action is the result of evaluating $(M+1)$ sub-problems, and then choosing the best action (Theorem 9). The enumeration of the $(M+1)$ cases is tractable in practice since the matrices $C_{t+1,T}$ can be computed offline.

Proposition 9. *The optimal node control action* $\mathbf{u}^*(t)$ *conditioned on a forced topology transition to mode k at* $t+1$ *(i.e.,* $m(t+1) = k$*) is obtained by setting* $\eta^T = \sum_i \mathbf{c}_{ik}^T$ *in Theorem 8.*

Proof. Use Definition 9 and Proposition 2 to obtain an appropriate $\mathbf{q}(t+1)$. The rest of the proof follows that for Theorem 8. □

Theorem 10 (Integrated control)**.** *The optimal integrated (i.e., node and topology) control action at time t is the one that results in the lowest objective function value from among the following options:*

(i) No forced topology transition, and optimal node control as in Theorem 8.

(ii) Forced topology transition to some mode $k \in \{1,\ldots,M\}$, and a conditionally optimal node control action, as in Proposition 9.

Proof. Similar domain decomposition argument as done for the proof of Theorem 9. □

The integrated controller prescribes a potential topology transition and a control $\mathbf{u}^*(t)$ on the node state as described in Theorem 10. It is important to note that the integrated controller controller does not simply choose between the node and topology controllers, but rather chooses the optimal solution (i.e., the minimizer of Equation 5.1) over the space of all possible combinations of node and topology control actions.

5.3 Evaluation of Controller Performance

We aim to minimize airport delays over the course of a 24-hour operational day (4 am EST to 3 am EST). The initial conditions for the PMJLS model, $\mathbf{x}(0)$ and $m(0)$, are drawn from historical data (2011-12). We study the performance of the controllers in simulation, for varying values of node and topology control action penalties. In our numerical experiments, $\mathbf{x}(0)$ is normalized, $U_{max} = 100$, and the control penalties are parametrized as $\beta(t) = \beta\mathbf{1}$ and $\alpha_{ij}(t) = \frac{\alpha(25-t)}{\pi_{ij}(t)}$, where α and β are non-negative scalars. The node control penalty, β, is the unit cost for reducing the delay at an airport. The topology transition penalty is proportional to the time remaining during that day, since actions are more challenging to implement early in the day, and affect a higher volume of downstream traffic. Furthermore, the topology control costs are assumed to be inversely proportional to the transition probabilities, to reflect the property that rare transitions are more difficult to induce than the ones that occur naturally with a high probability.

For a set of parameters (α,β) and a control policy, we measure the effectiveness of a strategy in terms of its delay cost:

$$\text{Delay cost } (\alpha,\beta,\text{control policy}) = \frac{\sum_{\tau=1}^{24} \mathbb{E}\left[\|\mathbf{x}(\tau)\|_1 + \text{control penalty}(\tau)\right]}{\sum_{\tau=1}^{24} \mathbb{E}\left[\|\mathbf{x}(\tau)\|_1 \mid \text{no control action}\right]} \tag{5.12}$$

The delay cost (or simply the cost) should be less than or equal to 1 for any reasonable controller; lower the delay cost, the better the controller performance. Finally, we note that the evolu-

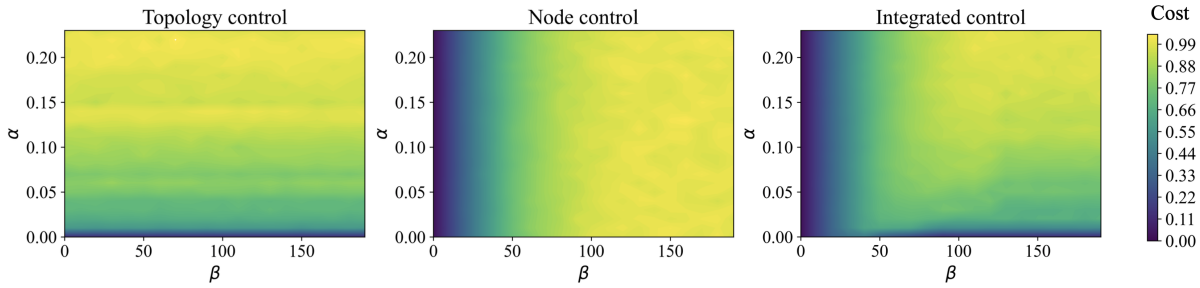


Figure 5-4: Delay cost contours for varying α and β, for (left) topology control, (center) node control, and (right) integrated control. The yellow regions denote the highest delay costs. Smaller penalties (values of α and β) result in lower costs, and that integrated control obtains lower costs over a larger range of penalties.

Control strategy	**Delay cost**	**Number of control actions**		
		Node	Topology	Both
Node controller	0.79	2.02	-	-
Topology controller	0.85	-	3.22	-
Integrated controller	0.70	1.91	2.26	0.07

Table 5.1: Controller performance averaged over $\alpha \in [0, 0.25]$ and $\beta \in [0, 200]$.

tion of $\mathbf{x}(t)$ is stochastic and thus the expectation in Equation 5.12 is estimated empirically using $1,000$ simulations for each value of α and β.

5.3.1 Comparison of Node, Topology and Integrated Control

From Figure 5-4 (left and center) we observe that smaller values of α and β result in lower delay costs for the topology and node controller, respectively. This finding is consistent with our expectation that the effectiveness of a controller would decrease if the penalty for an intervention is high. The integrated controller jointly optimizes over the entire space of node and topology actions to obtain greater benefits over a wider range of α and β values (Figure 5-4, right).

Table 5.1 quantifies the average performance of the controllers over the entire parameter space. The integrated controller has the lowest average cost, and hence the best overall performance. Two major factors contribute to the lower costs: Firstly, the integrated controller can switch between node-only and topology-only control. This observation is qualitatively confirmed from Figure 5-4, where we see that the integrated controller captures the low cost regions of both the node and topology controllers. Secondly, the integrated controller can uniquely combine node and topology control actions to obtain lower costs than either of them can. For instance, $\alpha = 0.13$ and $\beta =$

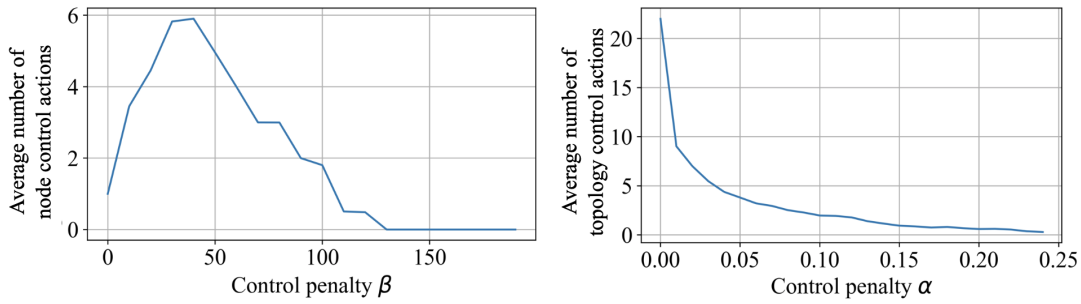


Figure 5-5: Number of control actions for (left) optimal node control, and (right) optimal topology control, for varying values of control action penalty, in our air traffic delay network example.

120 results in node, topology, and integrated control costs of 0.98, 0.99, and 0.94, respectively. An analysis of the number of control actions (Table 5.1) yields similar insights: Although the integrated controller mostly chooses either a node-only or topology-only action, it also chooses to do both of them on occasion, in order to obtain lower costs.

Remark 2. *Node control can achieve significant delay cost reductions, even with a small number of control actions.*

Small penalties for control actions (i.e., small values of α and β) result in lower costs (Figure 5-4). However, lower delay costs are not necessarily associated with more control interventions, as shown in Figure 5-5. Here, we see that smaller control penalties can be associated with either a small or a large number of control actions, depending on the nature of the actions. In particular, the number of node control actions performed for small values of β is just 1, as the first action $\mathbf{u}(1)$ can be used to reduce the delays very effectively.

Remark 3. *The node, topology and integrated control policies are optimal one-shot controllers.*

These three controllers prescribe the optimal action to be taken at time t assuming that no other action will be taken at any future time, i.e., the objective function in Equation 5.1 optimizes for the best one-shot control action that could be performed at time t.

Remark 4. *The node, topology, and integrated controllers minimize the cost-to-go (Equation 5.1), and not the delay cost (Equation 5.12).*

It is therefore possible that other controllers with potentially lower delay costs may exist. Furthermore, the delay cost of an integrated controller may not always be lower than that of a topology or

node controller, even though the integrated controller is the optimal one-shot controller that minimizes the cost-to-go. For instance, when $\alpha = 0$ and $\beta = 20$, the node, topology and integrated control costs are 0.35, 0.11, and 0.33 respectively. In this case, the integrated controller, which is determined using one-shot optimization, is unable to incorporate the fact that mode transitions are free at all future times, and any actions on nodes, even though attractive currently, might increase the delay cost in the long run.

Remark 5. *Integrated controllers are a good choice for practical implementation.*

Real-world implementation occurs in settings with uncertain dynamics, temporal and spatial constraints on the control actions, and with a rolling time horizon. One-shot controllers that minimize the cost-to-go are therefore a good choice in practice since they are quite robust to model and control action uncertainty. Our numerical simulations suggest that the integrated controller not only performs the best on average over a wide range of parameters (Table 5.1), but is also never the worst-performing controller of the three alternatives.

5.3.2 Partial Controllability

The ability to control the continuous nodal state or topology transitions may be limited in practice. We consider two examples of partial controllability: The first one restricts the ability to reduce delays (i.e., node control) to only a subset of airports, for example, due to regulatory constraints. The second scenario affords limited temporal flexibility in implementing control actions, for example, due to coordination efforts between multiple ATC facilities. For each of these cases, we estimate the delay costs to evaluate the system-wide impact of these restrictions. These scenarios also help evaluate the resilience of the system (by identifying critical airports, and control action timings), and help quantify the ability of the system to regulate its performance under degraded operational conditions.

Figure 5-6 plots the effect of limiting node action to only one airport. The top four airports in the US which have the largest individual ability to reduce total system delays (for $\alpha = 0.1$ and $\beta = 10$) are Atlanta (ATL), Chicago O'Hare (ORD), San Francisco (SFO), and Los Angeles (LAX). In other words, if we were forced to focus our delay reduction efforts at only one airport, choosing from within this set would be most effective.

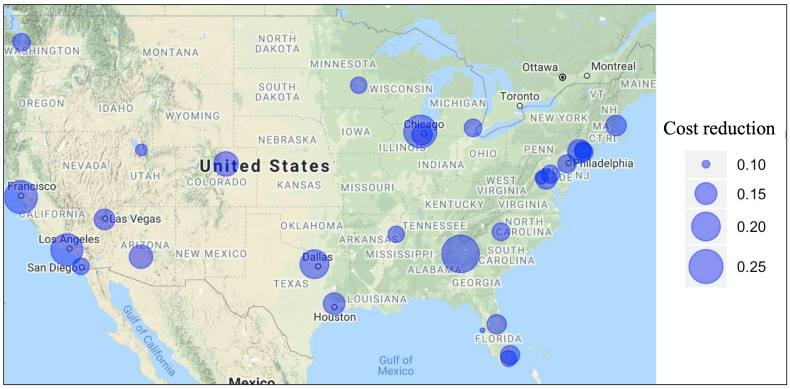


Figure 5-6: The size of the circle at an airport is proportional to the reduction in delay costs when the node control component of an integrated controller (with $\alpha = 0.1$ and $\beta = 10$) is limited to only that airport.

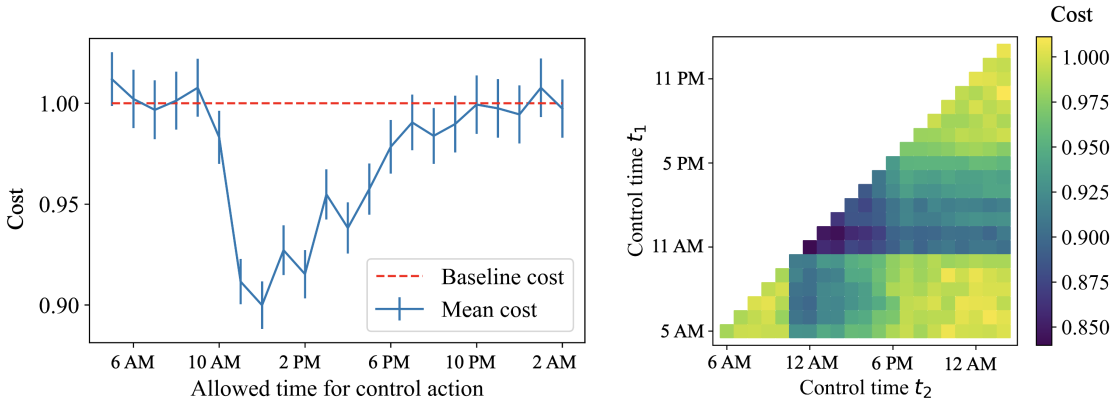


Figure 5-7: Delay costs (with the 99% confidence intervals) when integrated control actions are limited to (left) once in a day, at time t, and (right) twice in a day, at t_1 and t_2, where $t_1 < t_2$. Parameter values: $\alpha = 0.1$ and $\beta = 10$. All times are in EST.

The delay cost when restricting control actions to particular times of the day is shown in Figure 5-7. The ability to reduce delays varies significantly based on the chosen time(s). In particular, Figure 5-7 (left) shows that if we are allowed to perform an integrated control action only once during a day, doing it early in the morning or late at night is not effective. In the former case, delays have not significantly built up for the control to be effective (and the topology control cost is higher), whereas in the latter case, most of the day's traffic and delays have already been realized, and very little is left to be controlled. Figure 5-7 (left) suggests that around noon Eastern Time is the ideal time to implement the solution of an integrated controller, to decrease delays by almost 10%. This notion of restricting the timing of control actions can be extended to two time-periods, t_1 and t_2, as shown in Figure 5-7 (right): the results suggest that 11 am and 12 pm Eastern Time

are the most effective times to apply control actions to reduce delay costs.

5.4 Towards strategic planning and control

In this chapter, we made use of the positivity property of the MJLS to develop controllers which induce discrete mode transitions. According to our problem setup, we observe the continuous and discrete state of the system at the current time, and use our knowledge of the system dynamics to compute an optimal action that will be performed immediately. This is a very reactive setup, with minimal considerations of any future control decisions that will be performed (as highlighted in Remarks 3 and 4). A strategic (i.e., more proactive) control framework is beneficial, as it can :

1. improve efficiency, by minimizing the total delay costs, and
2. provides preparation time to implement the control actions in practical settings, as many of the resultant decisions (e.g., implement a GDP, cancel flights) involve multiple stakeholders (e.g., airports, FAA, airlines).

We will describe our ideas towards this goal.

1. **Strategic topology control**: Deterministic mode transitions can be planned in advance, even before $t = 0$. A control policy obtained in this manner would be a set of decision rules of the form "*When in mode* 4 *at* $t = 7$*, force a transition to mode* 5". Identifying this set of decision rules is a combinatorial optimization problem over the space of all possible mode forced transitions. However, we observe that greedy heuristics sequentially build this set of decision rules, while simultaneously evaluating its benefits using an updated π_{ij} matrix with Proposition 5 can work well in practice.

2. **Strategic node controllers:** Strategic planning of node control action before $t = 0$ is more challenging. Because the state $\mathbf{x}(t)$ is constrained to be non-negative, it is challenging to identify an action $\mathbf{u}(t)$ apriori that will ensure non-negativity at $t+1$ for all realizations of the stochastic process. Instead, we apriori identify time-periods where the control action is most effective, and provide estimates of the expected required control signal so that the system operator may be prepared in advance.

Chapter 6

Prediction of Network Delays

The prediction of air traffic delays, even a few hours in advance, has the potential to improve system performance by enabling the ATC to take proactive preventive measures, and by helping airlines plan recovery operations better. This chapter demonstrates the use of machine learning and modeling techniques to predict delays in air traffic networks a few hours (or even a day) ahead of time. Using different classes of models – ranging from a specialized hybrid system model of network delay propagation (MJLS) to more generic Artificial Neural Network (ANN) models – we show that the applicability of the model, as well as its performance, varies depending on the underlying prediction problem. While the main objective of this section is to build models for predicting delays, we would also like to understand the performance of recently popular black-box approaches vis-a-vis our carefully curated MJLS model.

Prior work: Air traffic delay prediction has been an active topic of research over the past few years. Departure delay distributions have been predicted in [187]. In [136] and [138], the authors assess the impact of weather, and use a Weather-Impacted Traffic Index (WITI) to predict delays. Bayesian Networks have been proposed in [131] to capture the subsystem level interactions and its impact on system-wide delay. In [101] and [121], the authors identified important network features of delay, and used it to predict delays on the top 100 delayed Origin-Destination pairs (OD-pairs) using Random Forest methods. This thesis extends this body of literature by comparing the performances of different classes of air traffic delay prediction models.

Recently, Artificial Neural Networks (ANN) and deep learning have received significant attention in a wide range of applications, including the prediction of air traffic delays [188, 36, 37].

Recurrent Neural Networks (RNN) have been used to model sequences of arrival and departure flight data [37]. The accuracy was shown to improve by using deeper RNN architectures. In our application of ANN to the problem of network delay prediction, we do not focus on the benefits of deep architectures. Instead, we wish to quantify the performance of ANNs for different kinds of feature vectors.

Key Contributions: The contributions of this chapter are summarized below.

1. The ANN models, which are relatively simple to build, are found to be effective for delay classification problems (i.e., predicting whether the delay on a link will be above or below a threshold). On the other hand, for regression problems (i.e., predicting the actual delay level on a link), the specialized MJLS models perform the best.

2. The importance of features differed by problem and prediction horizon. For instance, the time-of-day was the most important factor driving the accuracy of OD-pair delay classification. However, for the OD-pair delay regression problem, time-of-day became less important at longer prediction horizons, and network factors (type-of-day) gained prominence.

3. Our results serve as a valuable baseline for delay prediction algorithm refinements that account for more detailed features such as weather disruptions and Traffic Management Initiatives (e.g., Ground Delay Programs).

Chapter Outline: The network delay prediction problem is defined in Section 6.1. Section 6.2 describes the features for prediction, and Section 6.3 presents the techniques that will be used. In Section 6.4, we explain the data sources and the datasets that will be used for the prediction and regression problems. Sections 6.5, 6.6 and 6.7 present the results of the OD classification, OD regression, and airport regression problems respectively.

6.1 Problem Setup

We start with a quick summary of the delay network representation of the system delays. The delay network at time t is the representation of the network of all the OD-pair delays at that time. It is a weighted, directed graph in which the weight of an edge corresponds to that OD-pair delay

at that time. Figure 6-1 is an illustration of a simple delay network. The nodes of the network are airports and there are arrows (or directed edges) with numbers that indicate each of the OD-pair delays (i.e., the edge weights) at time t.

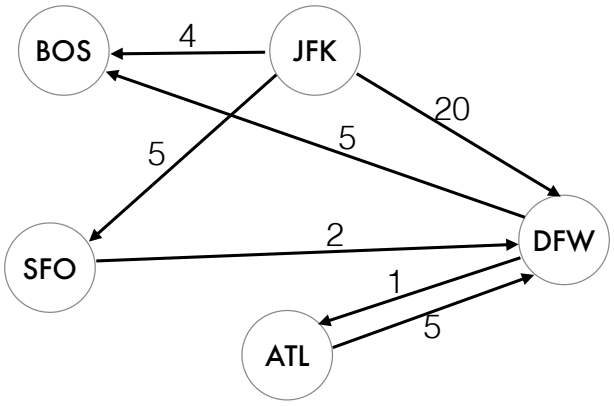


Figure 6-1: An example of a delay network. The edge weights are the OD-pair delays in minutes.

We can also calculate the airport delay from the delay network. For example in Figure 6-1, the airport delay of Dallas Fort Worth (DFW) is $\frac{1+5}{2} = 3$ min.

With this overview, we will present the following delay prediction problems:

1. **Classification of OD-pair delays**: In this case, we predict whether or not the delay on an OD-pair, Δt hours in the future, will exceed a pre-specified delay value (henceforth referred to as the *threshold*). Δt is the *prediction horizon*, namely, the number of hours into the future for which the prediction is made. The resulting problem is one of classification, in which we want to associate the future delay with one of the two categories (or classes): 'above threshold' or 'below threshold'. A range of prediction horizons (from 2-24 hr), along with different classification thresholds (30 min, 60 min, 90 min) will be considered.

2. **Prediction of OD-pair delays**: Here, we predict the OD-pair delays, time Δt hours in the future. By contrast to the classification problem, the actual value of OD-pair delays (in minutes) is predicted here, and not just whether or not the delay is above a threshold. In other words, this is a regression problem.

3. **Prediction of airport delays**: Similar to the OD-pair delay prediction, we predict the delay value for an airport, Δt hours into the future. Once again, a range of prediction horizons,

from 2-24 hr, are considered.

We compare several classes of methods for solving the classification and regression problems. For the first method, Artificial Neural Networks, we use standard network architectures for both the regression and classification problems. The second method is a classical technique from the machine learning literature: Classification and Regression Trees (CARTs). The third method, standard linear regression, applies only to the regression problems. Finally, we present a Markov Jump Linear System (MJLS) model, and demonstrate its applicability for both the OD-pair and airport delay regression problems.

Feature selection is a key aspect of machine learning problems, and greatly influences the performance of the algorithms. Therefore, we first describe in detail the feature vectors that are used to train the different classes of models.

6.2 Feature vectors

Except for the MJLS model, all the methods we consider in this chapter are supervised learning algorithms. This means that when they are trained, they are presented with input-output mappings, and the model learns appropriate parameters from these training samples. For the classification problem, the output is a binary (0-1) vector; for the regression problems, it is either the OD-pair delay or the airport delay.

The input presented to train the model, called the *feature vector*, can be selected in multiple ways. We first describe the different feature vectors that are used for the OD-pair classification and regression. Then, we present the feature vectors that are used for the airport delay regression. The notation used is as follows: At time t, the delay at time $(t + \Delta t)$ needs to be predicted. In other words, the feature vectors can only use information that is known at or before t.

The following are factors that are considered for the OD-pair classification and regression problems.

- **Time of day**: Delays show temporal patterns. They tend to be small or zero between midnight and 6 am, and start increasing in the morning. By noon, most OD-pairs have a non-zero delay because of high traffic and associated congestion. External factors like weather also

cause delays during the day. They tend to peak towards the evening, when congestion effects are the highest, and finally trail off at night once traffic decreases. Therefore, the time of day is an important feature.

- **Day of week**: Traffic, and consequently delay patterns, depends on the day of the week, making it a potential feature [189, 121, 190].

- **Season**: Since weather disruptions exhibit seasonality, the year is grouped into seasons based on the delays.

- **OD-pair delays**: In addition to the current OD-pair delay, the progression of OD-pair delays (for example, the delay for the past 2 hours), is an important indicator of delay trends. A high delay at 4 pm on the JFK-SFO link may indicate that the situation will worsen by 6 pm, when demand peaks.

- **Delays on adjacent OD-pairs**: Since delays tend to propagate in the network [30, 133, 132, 31], the delay on a particular OD-pair is influenced by those on adjacent OD-pairs (Figure 6-2). This feature is most important for short prediction horizons, but for longer horizons, delays on non-adjacent links could also become important.

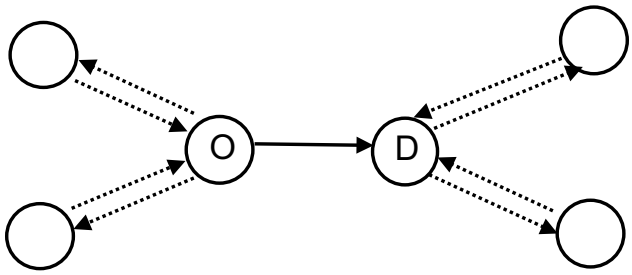


Figure 6-2: The dotted edges are adjacent links to the solid line OD-pair.

- **Type-of-hour (delay mode)**: While the adjacent OD-pair delay is a local property, the type-of-hour or delay mode is a more global, network-centric measure of the delay. Characteristic delay modes are identified by clustering delay networks [140]. These modes are also categorized depending on whether they are associated with increasing or decreasing delay trends. Each hour is associated with a delay mode (Table 6.1) and a representative delay network.

Types-of-hours (Delay Modes)	Occurrence (% of hours)
San Francisco Increasing	4
San Francisco Decreasing	1.8
Atlanta Increasing	2.4
Atlanta Decreasing	2.8
Chicago Increasing	2.9
Chicago Decreasing	1.3
High NAS Increasing	1.9
High NAS Decreasing	2
Medium NAS Increasing	9.3
Medium NAS Decreasing	6.1
Low NAS Increasing	38.1
Low NAS Decreasing	27.1

Table 6.1: Delay modes in 2011-12.

Prior work has also used the delay mode as a feature in delay prediction [101]. This feature incorporates the effect of weather conditions (like IMC or VMC) on the network delay.

- **Type-of-day**: Delay effects of one day may persist and affect the next day. The type-of-day variable captures this effect by grouping days into one of six categories based on the sequence of delay networks for the day [140]. Since the type-of-day for the current day may not be known at the time of making the prediction, the previous type-of-day variable is used.

In summary, we have three temporal variables (time, day of week and season), two local delay variables (OD-pair delay and adjacent OD-pair delay) and two network delay variables (delay mode and type of day). Using these features, we create 7 candidate feature vectors that are used for OD-pair delay classification and prediction. The different vectors incorporate different factors. We study the importance of these factors by studying the performance of prediction algorithms on different feature vectors.

F1 = [OD-Delay(t)]

F2 = [OD-Delay(t), t]

F3 = [OD-Delay(t), t, day-of-week, season]

F4 = [OD-Delay(t), t, day-of-week, season, type-of-hour, previous type-of-day]

F5 = [OD-Delay(t), OD-Delay($t-1$), OD-Delay($t-2$)]

F6 = [OD-Delay(t), type-of-hour, t]

F7 = [OD-Delay(t), Delay on adjacent OD-pairs(t), t]

The feature vectors for the airport delay prediction include the airport delay variable instead of the OD-pair delays.

$\tilde{F}1$ = [Airport delay(t)]

$\tilde{F}2$ = [Airport delay(t), t]

$\tilde{F}3$ = [Airport delay(t), Airport delay($t-1$), t]

$\tilde{F}4$ = [All airport delays(t), previous type-of-day, t]

6.3 Techniques for predicting network delays

Artificial Neural Networks (ANN): We consider three architectures for the ANN, based on standard models [36, 188, 191].

N1: *Multilayer Perceptron.* This is also referred to as a feedforward net. The network has an input layer, one or more hidden layers, and an output layer. We use two hidden layers with 10 perceptrons each, since there was little performance improvement from adding more perceptrons or layers. The input and the hidden layers use a logistic activation function. The output layer uses a linear transfer function so that the range out of the output is $(-\infty, \infty)$. The architecture N1 can be used for regression as well as for classification. When N1 is used for regression, the output layer has just one neuron that gives the value of the delay. For the classification problem, two output neurons are representing the two classes, 'delay above threshold' and 'delay below threshold', respectively.

N2: *General Regression Neural Network.* This is an efficient, 1-pass learning architecture that employs radial basis functions [192] for regression problems.

Method	Abbrev.	Classification	Regression
Multi-layer perceptron / Feedforward net	N1	✓	✓
Generalized Regression Neural Network	N2		✓
Probabilistic Neural Network	N3	✓	
Classification Tree	CT	✓	
Regression Tree	RT		✓
Linear Regression	LR		✓
Markov Jump Linear System	MJLS		✓

Table 6.2: Summary of the delay prediction techniques for the OD classification and OD/airport regression.

N3: *Probabilistic Neural Network (PNN)*. These have a layer of neurons with radial basis functions, and a final competitive layer that is used for classification. In the competitive layer, the input neurons 'compete' amongst themselves and only one output class is activated. A more detailed discussion of PNNs can be found in [193].

Classification and Regression Trees (CART): Decision trees map input vectors (or observation variables) to target values in the leaves. The branches split the input observation based on the observation value (the elements in the input vector) and this process is recursively done till we reach the leaves. In a Classification Tree (CT), the leaves represent one of the classes. In this chapter, it will represent either a 'delay above threshold' or a 'delay below threshold' class. When the leaf represents the value of a continuous variable, it is called a Regression Tree (RT) [194].

Linear Regression (LR): In the linear regression model, the output is a linear combination of the input variables. The output would be the OD-pair delay, or the airport delay. The input variables could be continuous or categorical (like the time of day or season) and we use standard techniques to learn the coefficients.

Markov Jump Linear System (MJLS): The MJLS model can directly be used for airport delay prediction. The characteristic delay modes (described using the networks A_m) are used for OD pair delay prediction. If the current mode is known, the transition probability gives a future mode distribution. A probability-weighted average of A_m gives an estimate of the OD pair delay on all links.

6.4 Data sets

We use the Bureau of Transportation Statistics (BTS) data for 2011 and 2012 [1]. The data contains the delay values for all flights of commercial airlines that accounted for at least 1% of the passenger traffic. The delay networks are constructed by considering only those links on which there are at least 5 flights a day, on average. If there are multiple flights on an OD-pair, then the delay state of that link at hour t is taken to be the median departure delay of the flights taking off between t and $t+1$ hour. In this manner, we obtain delay networks for each hour of the day, for the two years. The network contains 1,107 OD-pairs and 158 airports. We derive three sets of delay data from this 'master set' in order to study the performance of our prediction algorithms.

Dataset A: This set was used to evaluate the performance of the OD-pair delay classifier and delay level predictor. Data from 2011 was considered for training, and data from 2012 for testing. For each OD-pair, only those data points in which a) delay was non-zero and b) not during overnight hours, i.e., midnight-9 am EST (U.S. Eastern Standard Time), were included. Thereby only periods of non-zero traffic are considered, and outliers were removed. The performance of our algorithms on *Dataset A* are, for a practical prediction scenario, the most relevant. Unless stated otherwise, the results presented in this section refer to *Dataset A*.

Dataset B: This set was used to evaluate the performance of the OD-pair delay classifier and the delay level predictor under more challenging conditions. This data set is balanced, meaning that there are an equal number of high- and low-delay data points. For every OD-pair and a given classification threshold, say 60 min, half the data points have the OD-pair delay above 60 min and the other half has a delay of less than 60 min. All the points with delay above the threshold are chosen (since they are typically fewer in number), and the low delay points are chosen at random. The number of data points differ by OD-pair, since they each experience different durations of high delay periods (or non-negative delay periods). It is important to note that the data points in *Dataset A* and *Dataet B* have no time ordering – consecutive data points could be from time instants that are months apart. The prime motivation for the *Dataset B* is to evaluate classifier performance, but we still use it (with a 60 minute threshold) for the OD-pair regression problem for completeness.

Dataset C: This is used for evaluating the aggregate airport delay predictions. The FAA Core 30 airports, which are airports with high passenger traffic are considered (refer to Table 6.5 for the

complete list). For each airport, all time periods where traffic is non-zero and the time is between 9 am and midnight EST are considered. Data from 2011 is used for training, and 2012 for testing.

In the remainder of the chapter, we evaluate the various models for the classification and regression problems.

6.5 Classification of OD-pair delays

60 min classification threshold and 2 hr prediction horizon: For every OD-pair in *Dataset A*, we train them using the neural network architecture N1 and feature vector F1. The accuracy of classifying the delay on the link as above or below 60 min, with a 2-hr prediction horizon, is shown in Figure 6-3. The OD-pairs are sorted by increasing prediction accuracy. The prediction accuracy varies from 84% to 99%. The average accuracy of using a multi-layer perceptron neural network (N1) with feature vector F1 is 93.6%. The accuracy of the other methods, with feature vectors F1-F7, are shown in Figure 6-4.

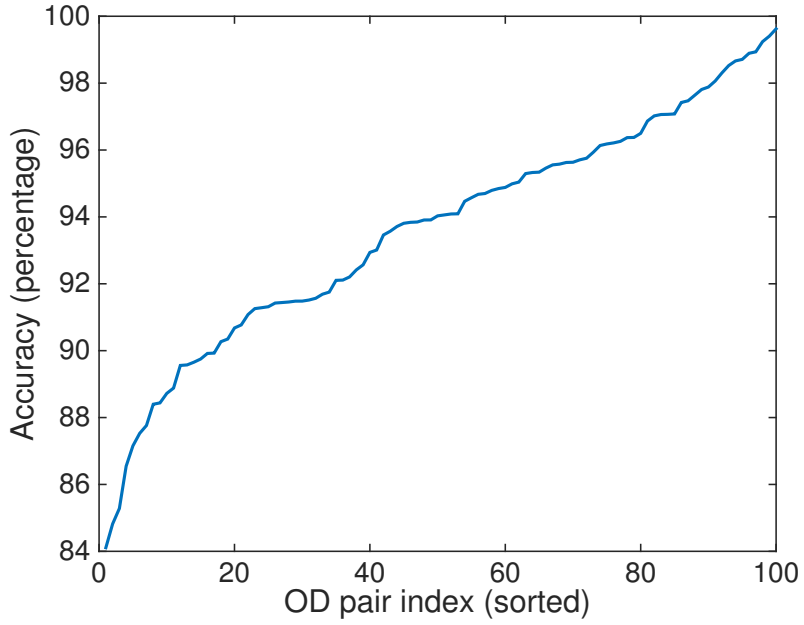


Figure 6-3: Accuracy for predicting OD-pair delays, for a classification threshold of 60 min and prediction horizon of 2 hr, using neural net N1, and feature F1.

The architecture N1 is consistently the best performer among the three methods, with an accuracy of over 93%. The highest accuracy of 93.7% among all the methods was with network N1 and

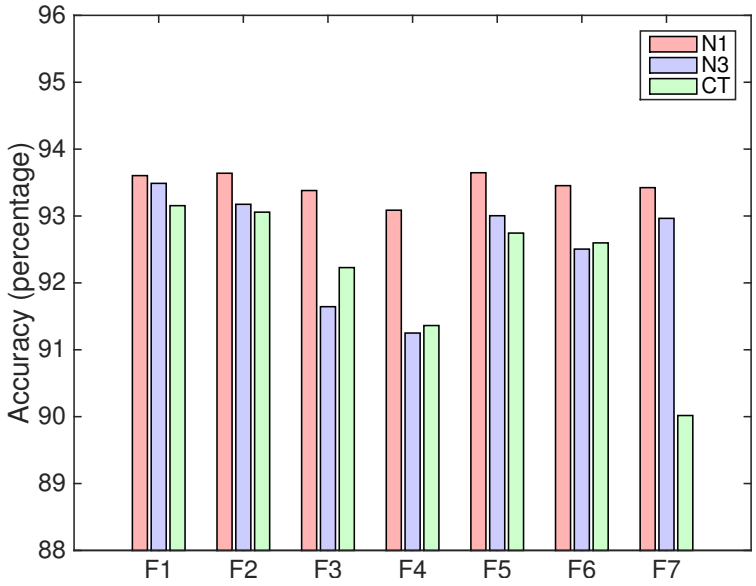


Figure 6-4: Comparison of neural net N1, N3 and classification tree (CT) with features F1-F7, for 2-hr classification of OD-pair delays, with a threshold of 60 min.

feature F2. The feature vector F5 which contains Information about the delay trend (it includes delay at t, $t-1$ and $t-2$) is very close, with an accuracy of 93.6%. It is interesting that the accuracy of the neural network does not change significantly based on the features, and that the addition of network information in F4 does not improve the accuracy of any method. N1 has a small decrease in performance, whereas N3 and CT are significantly worse. Finally, the feature F6, which includes delays from adjacent OD-pairs also gives limited improvement to the neural nets, and worsens the performance of the CT. For this prediction problem, the neural network architectures outperform CT.

Effect of classification threshold: We use architecture N1 with feature vector F2, since it was the best performing method, in order to study the variation of accuracy with classification threshold. We consider 30 min, 60 min and 90 min classification thresholds.

Figure 6-5 shows that for the same OD-pair, accuracy increases as the classification threshold increases. In particular, the average accuracy is 85% for a 30 min threshold, 94% for a 60 min threshold, and 97% for a 90 min threshold. Intuitively, it is easier to predict whether the delay will exceed 90 min, than exceed 30 min, since such high delays will usually be preceded by cues such as increasing delay trends. Prediction with smaller thresholds is harder because of noise and other random fluctuations. It is worth noting that the U.S. Department of Transportation only counts

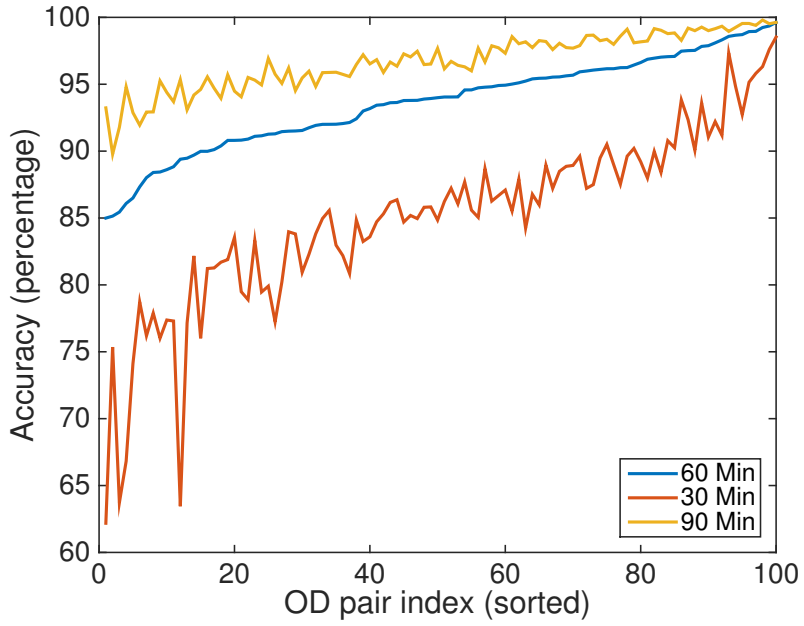


Figure 6-5: Accuracy of OD-pair delay classification with a threshold of 30, 60 and 90 min, and prediction horizon of 2 hr, using neural net N1 with feature F2.

a flight as being delayed if its delay exceeds 15 minutes. Finally, we note that the accuracy in predicting airport delays depends strongly on the airport itself – some airports have very predictable delay patterns, with a 30 min threshold prediction accuracy of over 95%, while others have an accuracy of only 65%. The connectivity of the airport and the variability in the weather patters at the airport can contribute to these differences.

Effect of prediction horizon: When the prediction horizon is increased from 2 hr to 4 hr, 6 hr, or 24 hr, the accuracies shown in Figure 6-4 decrease by less than 1%. The most accurate prediction technique remains N1 with the use of F2. However, there is an increase in the accuracy to 95% (CT with F2) when the prediction horizon is reduced to 1 hr.

Analysis with Dataset B: *Dataset A* is not a balanced dataset. While the accuracy of the best algorithm for a 2 hour prediction with a 60 minute threshold is 93.7%, even a naive classifier that always predicts a delay below the threshold will give an accuracy of 93.5%. On training and testing the algorithms using the balanced *Dataset B*, the classification accuracy of the best algorithm is 71% (Table 6.3). This is a more rigorous statistical analysis of the algorithms, and gives context to the accuracy: a naive classifier, which classifies all data points into any one type, will only be 50% accurate in this case. This demonstrates the benefits of using the specialized prediction techniques.

	N1	N3	CT
F1	66.6	66.7	67.2
F2	69.0	68.9	70.6
F3	64.2	60.6	69.4
F4	64.0	52.5	69.5
F5	65.7	66.2	67.1
F6	66.5	64.3	70.2
F7	67.5	50.1	70.0

Table 6.3: Accuracy (%) of different methods, for a 2 hr prediction and a 60 min classification threshold, using Dataset B.

6.6 Estimation of OD-pair delays

The OD-pair delay is predicted using neural nets N1 and N2, a regression tree (RT), linear regression (LR) and the Markov Jump Linear System (MJLS) model. The prediction error for an OD-pair is the median of the absolute error across all the data points in the test set (the year 2012). The prediction error for a method is defined as the mean prediction error over all the 100 OD-pairs.

Comparison of methods: For a 2 hr prediction horizon, the MJLS model has the lowest prediction error of 4.7 min. Among the other methods, the neural net N2 with feature F7 gives the lowest error of 8.4 min (Figure 6-6). To place these errors in context, it is important to compare them to the mean error across all OD pairs in the test set, which is 7.2 min. The Generalized Regression Neural Network (N2) is the better performing neural network, and it is marginally better than the Regression Tree. The delays on all adjacent links (F7) is a more significant feature than the network-theoretic features like previous type-of-day or type-of-hour (F4 and F6). This makes intuitive sense, because in the short term (i.e., 2 hr), an OD-pair delay is unlikely to be affected by delay on links that are more than one hop away. The error for N2 is, however, almost 80% more than the MJLS. This reflects the inability of the neural network to extract out complex features like the principal eigenvector, which form the basis for the modes in the MJLS model.

The distribution of prediction error for each of the 100 OD-pairs (Figure 6-7) shows a peak for

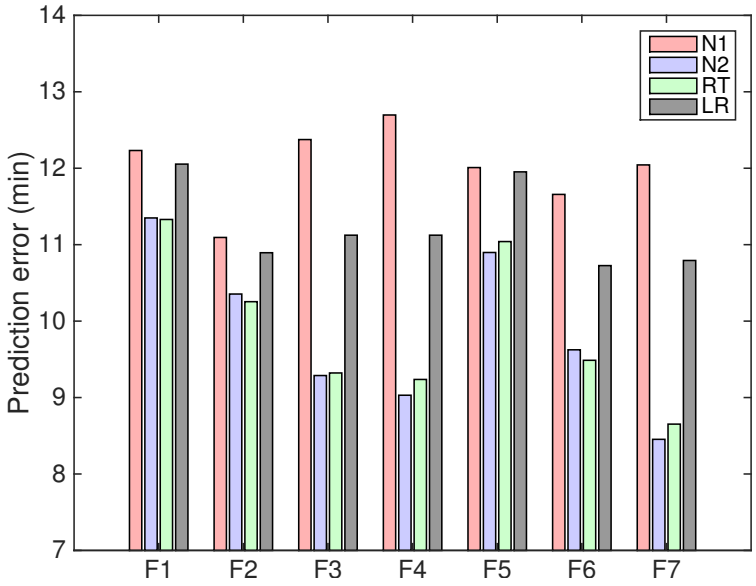


Figure 6-6: Average prediction error (in min) over 100 OD-pairs for a 2 hr prediction horizon. Neural networks (N1 and N2), Regression Tree, and Linear Regression, with feature vectors F1-F7 are considered.

the MJLS model at lower delay values. The tail of the distribution is longer for the Regression Tree, with the prediction error being as high as 25 min for one of the OD-pairs.

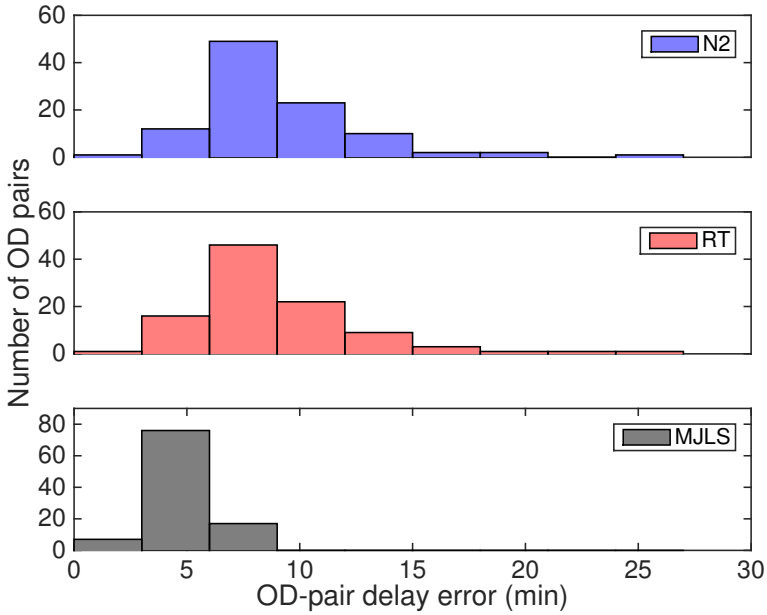


Figure 6-7: Distribution of OD-pair delay prediction errors for N2 and RT (both with feature F7), and the MJLS model.

Effect of prediction horizon: The trend of MJLS being the best prediction model followed by N2 and RT (both with feature F7) holds for a prediction horizon of 4 hr, 6 hr, and 24 hr (Figure

6-8).

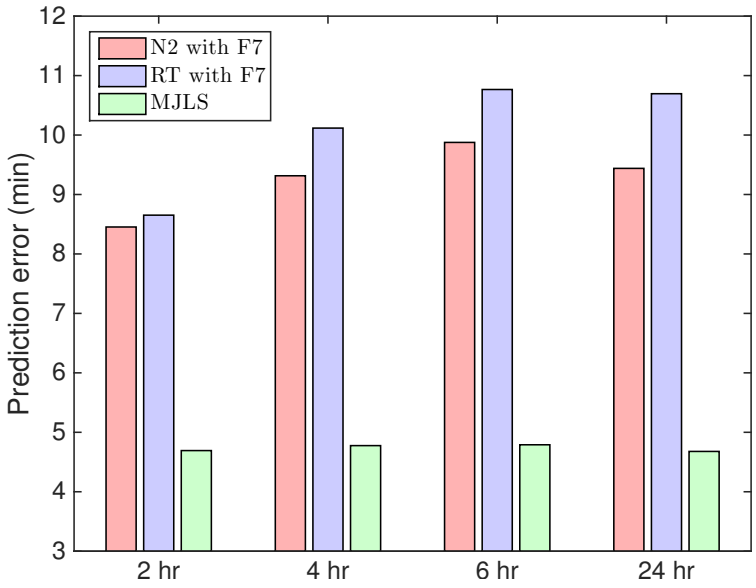


Figure 6-8: Distribution of OD-pair delay prediction errors for N2, RT, and MJLS, for a 2 hr, 4 hr, 6 hr and 24 hr prediction horizon

Analysis with Dataset B: With the balanced *Dataset B*, it is natural that the prediction errors will increase. While most models and feature vectors show an increase by a factor of 3 (Table 6.4), the N2 architecture with the F7 feature vector is robust and predicts the delay with an average error of 18.5 min. The MJLS model has an error of 44 min, and is not the best algorithm. With a lot of high delay data points in this set, a simple (although it is reasonably powerful, as seen with *Dataset A*) MJLS model is not able to capture the complex nonlinear delay dynamics that govern these high delay instances. The neural network performs much better in such situations. In this balanced dataset, the mean of the delay that is being predicted is 50.7 min.

	N1	N2	RT	LR
F1	42.2	39.4	39.4	44.2
F2	36.1	32.1	30.9	32.6
F3	44.9	36.0	31.1	34.9
F4	46.1	32.4	33.8	45.9
F5	44.0	37.6	37.4	43.5
F6	39.7	31.0	30.1	33.8
F7	42.6	18.5	29.5	77.54

Table 6.4: Mean OD-pair delay prediction error (in min) for 2 hr prediction horizon with *Dataset B*. For comparison, the MJLS model has a mean error of 44 min.

6.7 Estimation of airport delays

The airport delay regression is used to predict the average delay levels of outgoing links from an airport. This is a measure of the delay disruptions, or the quality of service at the airport.

Comparison of methods: We use the neural networks N1 and N2, a Regression Tree (RT), a Linear Regression (LR) and the MJLS model to predict the delay state of an airport. The feature vectors are $\tilde{F}1$-$\tilde{F}4$, as described in Sec. 6.2. For the MJLS, the current time, mode and the current delay at all airports is the feature. For each of the 30 airports in the FAA Core 30 list, the airport delay 2 hr in the future is predicted. The prediction error for an airport is the median error across all the data points. The average prediction error across all the 30 airports is defined as the prediction error for the particular algorithm (for the corresponding feature vector). Figure 6-9 shows the average prediction error for the different algorithms. The error for the MJLS model (not shown in the plot) is 6.8 min, and it is the lowest among all the models. The mean of the airport delays in the test data set is 14.7 min.

Among the models compared in Figure 6-9, the neural network N1 (with feature $\tilde{F}2$), which is a multi-layer perceptron network gives the lowest error of 7.1 min. The neural network outperforms the other classical techniques of Regression Tress and Linear Regression for the airport delay prediction; however, it does not perform as well as the MJLS. The MJLS model incorporates network effects and temporal dynamics (through the time-dependent transition matrices). Our neural network architecture does not learn all the temporal features because it treats each data point independently as a new observation, and not as a time-series. However, our neural network model

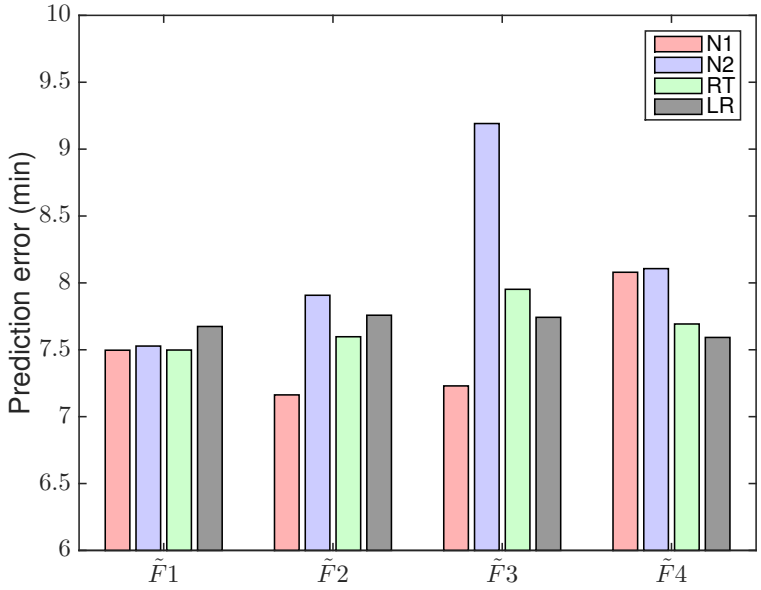


Figure 6-9: Comparison of all models for a 2 hr prediction of airport delay. The MJLS model (not shown) results in a prediction error of 6.8 min.

is simpler, and can be developed without any assumptions or intuition about the delay dynamics. We also observe that feature vectors F2 and F3, which contain temporal information, give better performance with neural network N1 than feature F4, which contains network effects (previous type-of-day). For predictions only 2 hr into the future, the current dynamics is a more important factor than the network state of the previous day. We emphasize that our neural network architectures are very simple; using a LSTM architecture with tens of layers and thousands of parameters can lead to better predictive performance. However, our focus is on comparing the delay predictions from simple models.

Effect of prediction horizon: For prediction horizons of 2, 4, 6, and 24 hr, the MJLS had the lowest prediction error among all the models. Among the other models, the lowest error is obtained by neural network N2 (for 2 and 4 hr prediction horizons), Regression Tree (for a 6 hr prediction horizon) and Linear Regression (for a 24 hr prediction horizon). The performance of these four methods are plotted in Figure 6-10.

While the neural network is better for lower prediction horizons, RT and LR become better for longer horizons. Feature $\tilde{F}4$, i.e., the previous type-of-day, becomes more useful at longer prediction horizons. The local dynamics at t and $t-1$ are intuitively not very useful when a prediction needs to be made at $t+24$. The superior performance of the MJLS model highlights

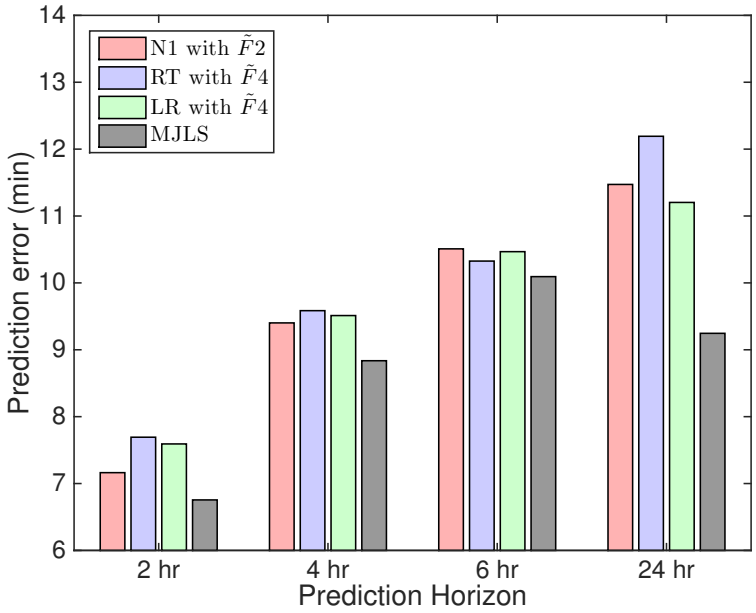


Figure 6-10: N2, RT, LR and MJLS prediction errors of airport delays, for 2 hr, 4 hr, 6 hr, and 24 hr prediction horizons.

the importance of specialized, physically interpretable models. Naturally, the MJLS prediction error will increase with increasing prediction horizons. This can be ascribed to the increasing uncertainty about the delay dynamics over longer time scales. However, the prediction error for the 24 hr horizon is lower than the 6 hr horizon. This apparent anomaly is due to the distribution of our data points. When making a 6 hour prediction, the MJLS uses delay at 4 am to predict the delay at 10 am. Since the model is multiplicative on the initial condition, it is extremely sensitive to the low delay, and fluctuations that are characteristic of a 4 am delay. On the other hand, when 24 hr predictions are made, delays during high traffic periods are used to make predictions. There is also valuable information about the previous type of day that is used, which explains why the error drops despite the increase in prediction horizon. The neural network and other models are not multiplicatively dependent on the initial state, and therefore do not exhibit such behavior. Finally, the median MJLS prediction error, as well as the median delay, by airport is shown in Table 6.5.

Airport	Delay	IQR	2 hr	4 hr	6 hr	24 hr
ATL	11.4	15.0	3.4	4.6	6.03	5.7
BOS	13.5	28.7	7.1	9.7	10.9	9.7
BWI	20	32.1	8.4	11.5	13.2	12.0
CLT	12.8	27.5	6.0	7.5	8.5	7.5
DCA	10.3	22.4	6.6	8.8	10.2	9.0
DEN	16.4	21.0	4.3	5.4	6.4	7.4
DFW	16.3	20.2	5.0	7.0	8.4	7.7
DTW	16.3	35.1	8.3	10.8	12.2	11.5
EWR	25.3	44.3	9.8	12.6	15.1	14.4
FLL	14.5	31.2	8.6	11.6	13.2	11.7
HNL	2.5	6.9	2.7	3.4	3.7	3.0
IAD	28.3	53.6	12.9	16.4	17.2	16.6
IAH	20.9	31.7	7.3	9.6	11.4	10.8
JFK	17.1	29.6	8.5	11.2	12.6	10.7
LAS	14.0	19.5	4.4	5.4	6.3	6.6
LAX	13.8	15.9	4.0	4.8	5.8	6.4
LGA	9.5	25.6	6.0	8.0	9.1	7.4
MCO	15.1	28.1	6.9	8.6	9.6	9.2
MDW	19.9	30.4	7.4	10.0	12.0	11.6
MEM	3.0	22.0	9.6	14.3	16.2	10.9
MIA	22.3	32.1	9.7	11.7	12.8	13.3
MSP	12.0	26.7	8.1	10.4	11.3	9.6
ORD	18.5	29.4	5.2	7.1	8.9	8.8
PHL	12.3	23.7	8.6	11.3	12.2	10.4
PHX	12.2	15.1	3.7	4.5	5.1	5.6
SAN	10.3	17.1	5.2	6.0	6.5	6.2
SEA	11.3	15.4	4.8	6.0	6.6	5.7
SFO	20.9	36.4	6.3	8.4	10.3	10.4
SLC	10.3	18.6	5.6	6.5	7.2	6.0
TPA	11.4	29.7	8.3	11.5	13.5	11.3

Table 6.5: Median delay and the Inter Quartile Range for each airport along with the airport delay prediction errors for 2, 4, 6, and 24 hr horizons on *Dataset C*, using the MJLS model. All quantities are in minutes

6.8 Discussions

It is apparent that the best choice of delay prediction method depends on (1) the specific classification or regression problem, (2) the dataset (balanced vs. unbalanced), and (3) the prediction horizon.

For the classification problem at a 2 hr prediction horizon and 60 min threshold, we achieved a 94% accuracy, for *Dataset A*. For the balanced dataset, the accuracy dropped to 70%. Similar accuracy was achieved in [101] using the time-of-day as a prediction variable. However, by using other network delay states and random forests, they were able to achieve a much higher accuracy. This result suggests that ensemble methods using artificial neural networks may be an interesting topic for further study.

For the OD-pair regression problem, the MJLS model performed the best when considering *Dataset A*. The identification and effective use of spatial delay patterns (delay modes), and the temporal evolution (time-dependent mode transitions), makes the MJLS model a very good predictive tool, with a mean error of 4.7 min. However, when a balanced data set was used, the neural network became the best performer with an OD-pair error of 18.5 min. The MJLS model is not appropriate to use for *Dataset B*, since the model development used the entire 2011 data points (and not the balanced data set) to identify significant delay modes. The closest point of comparison would be [101], where a Random Forest algorithm on a 2007-08 ASPM dataset had an error of 21 min, for a 2 hr prediction horizon.

The airport delay metric has not been predicted in prior literature. We hypothesize that ensemble methods (like random forests) would help boost the accuracy of neural network methods. However, they may still not be as accurate as MJLS (mean error of 6.8 min for predicting the average outbound delay at an airport two hours in the future), because of the specialized dynamical features that are explicitly accounted for by the MJLS model.

Chapter 7

Identification of Outliers in Graph Signals

Outlier or anomaly detection – the identification of data points that differ in a significant manner from the majority of the observations – is an important problem in data analysis, for several reasons. When such data points are included while training models, the resulting models can be unrepresentative of the real system. Furthermore, several commonly-used algorithms (for example, linear or logistic regression, and AdaBoost) are particularly sensitive to outliers. Anomaly detection also plays an important role in system health monitoring and diagnosis. Outliers could correspond to valid observations resulting from unusual, off-nominal, or unexpected events in the system. In such cases, outliers represent interesting observations that merit further investigation. These situations motivate the need to identify outliers, and to provide interpretable rationales for them being classified as such. When the data corresponds to observations of a networked system, an observation may be an outlier not only because of its absolute value (for example, too large or too small), but also because it corresponds to an unusual distribution of values across the nodes of the network. The growing ubiquity of networked systems in science and engineering further motivates the development of outlier identification methods for signals on networks.

In this chapter, we consider the problem of identifying outliers in data obtained from networked systems. We abstract the network interactions in the form of a graph with N vertices, and consider the data observations as signals supported on the vertices. Thus, each element of the observation vector (also referred to as a data point or a graph signal vector) is a scalar value associated with a graph vertex. The strength of the inter-dependencies between the signal values at pairs of nodes determines the edge weights of the graph.

Outlier detection in graph signals poses a different set of challenges when compared to anomaly detection in other types of data. The underlying relationships between the signals at different vertices of the graph result in a narrower class of "nominal" data points compared to the situation when the signal at each vertex is independent of that at other vertices. Therefore, conventional outlier detection techniques for multi-dimensional data sets that test whether the norm of the data vector is outside a certain range of values would be overly conservative for graph signals. Simply using the magnitude of the signal at a vertex is not sufficient to detect if the observation is an outlier or not. A particular observation of the graph signal may be considered an outlier because the *spatial distribution* of the signal magnitudes across the vertices significantly differs from other observations within the data set. In our work, we formalize these different notions of outlier characteristics, and derive criteria for detecting outliers in graph signals.

Prior work: Several techniques exist to detect outliers in multidimensional data sets. One approach is to identify clusters, and any observation that falls far away from a cluster is defined as an outlier [195]. Alternatively, an underlying multivariate statistical distribution of the data is assumed, and tests are performed to see if an observation lies at the extremes of the distribution [196, 197]. Techniques from signal processing such as filtered wavelet transforms for multidimensional data sets have also been considered [198]. In our work, we propose and analyze an outlier detection method that extends the above-mentioned signal processing techniques to analyze the spatial distribution of data in the graph domain.

Previous literature related to graph-based data have used information theory to identify structural features [199] and spatial outliers [200]. One particular characteristic, *Total Variation (TV)*, a measure of the smoothness of a signal supported on a graph, has been used as a feature vector for classification [201]. While the outlier detection problem has been addressed in several contexts (for example, time-series data [202]), it has received little attention in the context of graph signals. Our work attempts to fill this gap by using TV as a metric to identify graph signals with off-nominal or unexpected spatial distributions across the vertices. For a comprehensive overview of general graph signal processing techniques, we refer the reader to [203].

Key Contributions: We enumerate the key contributions of this chapter.

1. We formalize notions of outliers in graph signals based on the spatial distribution of the signal across the nodes.

2. Analytical bounds for the outliers are derived for the special case when a) the graph signal is a multi-variate Gaussian, and b) the strength of the interaction between adjacent graph vertices is known only upto certain bounds. The availability of such analytic expressions allows us to perform outlier detection even in the case of limited datasets, when there is insufficient data for empirical techniques.

3. We define an empirical approach to identify outliers when the data is not Gaussian by using skew-adjusted bounds.

4. We identify properties of the spectral decomposition of the graph signal that helps in interpreting why a data point was classified as an outlier.

5. We demonstrate the utility of the outlier detection and interpretation framework with the example of an airport delay signal dataset and discuss how this would be useful for airline operational managers and air traffic controllers.

Although we only discuss the airport delay network example in detail, the methods proposed in this chapter (and thesis) have a broad range of applications, including transportation, Internet-of-Things, power systems, communication networks, and biological systems.

Chapter Outline: The notations used for this discussion is introduced in Section 7.1. Next, we introduce the Total Variation (TV) as a metric to identify outliers based on the distribution of the graph signal in Section 7.2. Analytical bounds for the outliers when the data is a multivariate Gaussian is presented in Section 7.3. We use simple simulations to gain intuition on these bounds in Section 7.4, and motivate the need for a more general notion in case of skewed distributions. The skew-adjusted definitions for outliers are presented in Section 7.5. Finally, we present an application of our outlier detection framework to analyze delay disruptions in airport networks.

7.1 Setup and Notations

Consider a data set with M observations $\mathscr{O}_M = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}, \ldots, \mathbf{x}^{(M)}\}$, with each observation $\mathbf{x}^{(i)} \in \mathbb{R}^{N\times 1}$ and $\mathbf{x}^{(i)} = \left(x_1^{(i)}, \ldots, x_N^{(i)}\right)^\mathsf{T}$. We drop the superscript when talking about a generic observation or data point, and refer to it as $\mathbf{x}$. The N elements (or features) of $\mathbf{x}$ are not independent,

and their pairwise interactions are captured using an undirected graph $G = (V,E)$ with $|V| = N$ vertices and adjacency matrix $A = [a_{ij}]$ describing the edges E. We assume that the interaction between any two elements is symmetric, hence $A = A^\intercal$. The observation $\mathbf{x}$ can equivalently be considered as signals x_i supported on vertices $i \in V$ on the graph with connectivities and weights given by A.

The empirical mean of the signal at vertex i is $\hat{\mu}_i = \frac{1}{M}\sum_{k=1}^{M} x_i^{(k)}$. For each pair of unique vertices (i,j) and given the set of observations $\mathscr{O}_M$, we can compute the sample Pearson correlation coefficient, denoted as $r_{ij|\mathscr{O}_M}$. For brevity, we may denote it as r_{ij}.

$$r_{ij|\mathscr{O}_M} = \frac{\sum_{k=1}^{M}\left(x_i^{(k)} - \hat{\mu}_i\right)\left(x_j^{(k)} - \hat{\mu}_j\right)}{\sqrt{\sum_{k=1}^{M}\left(x_i^{(k)} - \hat{\mu}_i\right)^2}\sqrt{\sum_{k=1}^{M}\left(x_j^{(k)} - \hat{\mu}_j\right)^2}}. \tag{7.1}$$

In terms of notation, bold-face fonts indicate vectors, random variables are given in upper case, and a "hat" represents an empirically-derived quantity.

The graph signal vector $\mathbf{x}$ are assumed to be specific realizations of a random variable $\mathbf{X} = (X_1,\ldots,X_N)^\intercal \in \mathbb{R}^{N\times 1}$, where $\mathbf{X}$ is a multivariate random variable. The correlation coefficient between the signals on two adjacent vertices i and j is given by

$$\rho_{ij} = \frac{\mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]}{\sqrt{\mathbb{E}[(X_i - \mu_i)^2]\mathbb{E}[(X_j - \mu_j)^2]}}. \tag{7.2}$$

The sample correlation coefficient, $r_{ij|\mathscr{O}_M}$, is a consistent estimator of ρ_{ij}, *i.e.*, $\lim_{M\to\infty}\left(r_{ij|\mathscr{O}_M}\right) = \rho_{ij}$.

We will now define some basic terms from spectral graph theory.

Definition 15 (Graph Laplacian)**.** *The (combinatorial) graph Laplacian L with respect to a graph with adjacency matrix A is $L = D - A$, where $D = [d_{ij}] \in \mathbb{R}^{N\times N}$ is the diagonal degree matrix of the graph, with $d_{ii} = \sum_{j=1}^{N} a_{ij}$.*

The graph Laplacian L is a real symmetric matrix with a full set of orthogonal eigenvectors. The normalized eigenvectors are denoted by $v_i \in \mathbb{R}^{N\times 1}$, $i \in \{1,\ldots,N\}$, with $v_i^\intercal v_j = \delta_{ij}$, where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}. \tag{7.3}$$

All the eigenvalues satisfy $Lv_i = \lambda_i v_i$. We sort the eigenvalues such that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$. Since the graph Laplacian has row sums of 0 (Definition 15), $v_1 = \mathbf{1}$ is the *constant* eigenvector corresponding to the eigenvalue $\lambda_1 = 0$. Furthermore, the multiplicity of eigenvalues equal to 0 is the number of connected components in the underlying graph. Thus, if the correlation network is fully connected, then $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_N$, and span $(\{v_1, ..., v_N\}) = \mathbb{R}^{N\times 1}$. Consequently, any vector $\mathbf{x} \in \mathbb{R}^{N\times 1}$ can be written as a linear combination of $\{v_1, ..., v_N\}$; i.e., there exist scalars α_i such that $\mathbf{x} = \sum_{i=1}^{N} \alpha_i v_i$.

Definition 16 (Graph Fourier Transform, or GFT)**.** *The Graph Fourier Transform (GFT) of a graph signal vector* $\mathbf{x}$ *is the set of scalars* $\{\alpha_1, ..., \alpha_N\}$ *where* $\alpha_i = v_i^\intercal \mathbf{x}$.

To draw an analogy to the classical Fourier transform, the eigenvectors are equivalent to sinusoids on graphs, and the eigenvalues correspond to discrete frequencies. The scalar $\alpha_i \in \{\alpha_1, ..., \alpha_N\}$ represents the magnitude of contribution of the i^{th} eigenvector of "frequency" λ_i. Similar to the notion of spectral energies for the classical Fourier transform, larger eigenvalues are associated with eigenvectors having higher graph spectral energies, as follows.

Definition 17 (Spectral and total energy)**.** *The spectral energy of* $\mathbf{x}$ *corresponding the* i^{th} *eigenvector is* α_i^2*, and the total energy of* $\mathbf{x}$ *is given by* $\|\mathbf{x}\|_2^2 = \sum_{i=1}^{N} \alpha_i^2$.

The graph Laplacian can also be used to compute a measure of the "smoothness" of a graph signal $\mathbf{x}$, called the *total variation*.

Definition 18 (Total variation, or TV)**.** *The total variation (TV) of a graph signal* $\mathbf{x}$ *with respect to the graph Laplacian L is defined as:*

$$TV(L, \mathbf{x}) = \frac{1}{2} \sum_{i \neq j} a_{ij}(x_i - x_j)^2 = \mathbf{x}^\intercal L \mathbf{x}. \tag{7.4}$$

For brevity, we write $TV(\mathbf{x})$*, dropping the reference to the graph Laplacian L. When the TV is computed with respect to the random vector* $\mathbf{X}$*, we denote it as* $TV(\mathbf{X})$.

The total variation is a measure of the smoothness of a graph signal. A graph signal with a lower value of TV is said to be *smooth* in comparison to another signal with a higher TV.

Proposition 1. *Suppose we have a data point* $\mathbf{x}^{(k)} \in \mathscr{O}_M$ *and its GFT* $\{\alpha_1^{(k)}, ..., \alpha_N^{(k)}\}$. *Then, the following two statements are equivalent:*

(i) $TV\left(\mathbf{x}^{(k)}\right) = \left(\mathbf{x}^{(k)}\right)^{\mathsf{T}} L\mathbf{x}^{(k)}$.

(ii) $TV\left(\mathbf{x}^{(k)}\right) = \sum_{i=1}^{N} \left(\alpha_i^{(k)}\right)^2 \lambda_i$.

Proposition 1 formalizes the relationship between the GFT, the TV of a graph signal, and the spectral and total energies of a graph signal. Larger contributions of the eigenvector v_i to the GFT of $\mathbf{x}$ (i.e., larger values of α_i) result in a higher TV, translating to a less smooth graph signal. Similarly, the more energetic eigenvectors (i.e., larger values of λ_i) contribute to a higher TV, resulting in a less smooth graph signal. Since the eigenvalues are sorted in ascending order with respect to index $i \in \{1, ..., N\}$, we compare eigenvalue magnitudes using the index i.

7.2 Using the Total Variation Metric to Detect Outliers

Suppose the edge weights of the adjacency matrix are equal to the correlation coefficient, i.e., $a_{ij} = r_{ij}$. For a pair of nodes $i, j \in V$ connected by an edge with weight r_{ij}, the contribution to the TV is $r_{ij}\left(x_i - x_j\right)^2$. We consider the following possible scenarios:

Case A: Suppose the graph signals from i and j has a high positive correlation, i.e., $r_{ij} \to 1$. In this case, we would expect that both the signals x_i and x_j are large, or both are small. Thus, when a newly observed signal on the nodes is as expected (i.e., it is consistent with historical trends), $r_{ij}\left(x_i - x_j\right)^2$ is small since $\left(x_i - x_j\right)$ is small. However, when the observed signal on the nodes differ from historical trends, $\left(x_i - x_j\right)$ is large, and consequently $r_{ij}\left(x_i - x_j\right)^2$ is large.

Case B: When the graph signals at nodes i and j are uncorrelated, i.e. $r_{ij} \to 0$, we do not expect any relation between the observations x_i and x_j. Thus, $\left(x_i - x_j\right)$ could be large. However, since r_{ij} is small, $r_{ij}\left(x_i - x_j\right)^2$ is small.

The above reasoning is valid when the signs of the correlation coefficients are all the same. If only a small fraction of correlation coefficients have differing signs, a projective affine transformation can be applied to $r_{ij|\mathscr{O}_M}$, and the intuition still holds. The analysis of networks with mixed-sign correlation coefficients is a direction for future research. Cases A and B motivate the use of TV as

a metric for outlier detection in terms of a graph signal's spatial distribution. Since the only case where a high TV may occur is when the r_{ij} is large, and $\left(x_i - x_j\right)$ is large, such an occurrence is deemed to be *unexpected given historic observations*. The TV metric yields an aggregate representation of the behavior of $\mathbf{x}$ across the entire graph. We now define the notion of a *weak outlier in distribution*.

Definition 19. *An observation* $\mathbf{x}$ *is considered a* weak distribution outlier of level k *or a* weak outlier in distribution of level k *(abbreviated as Weak-OID) if*

$$TV(\mathbf{x}) \notin \left[\mathbb{E}\left[TV(\mathbf{X})\right] - k\sqrt{Var\left[TV(\mathbf{X})\right]}, \mathbb{E}\left[TV(\mathbf{X})\right] + k\sqrt{Var\left[TV(\mathbf{X})\right]}\right], \tag{7.5}$$

for some $k \geq 0$*. In other words, an observation is considered to be a weak outlier in distribution if its TV does not lie within k standard deviations of its expected value.*

Next, we will define outliers in scale (or magnitude). This definition is the classic one we would use for any multivariate signal, i.e., call an observation an outlier if the value of its norm $\|\mathbf{x}\|$ is significantly different from its mean $\mathbb{E}[\|\mathbf{x}\|]$.

Definition 20. *An observation* $\mathbf{x}$ *is considered to be a* scale outlier of level k *or an* outlier in scale of level k *(abbreviated as OIS) if*

$$\|\mathbf{x}\| \notin \left[\mathbb{E}[\|\mathbf{X}\|] - k\sqrt{Var[\|\mathbf{X}\|]}, \mathbb{E}[\|\mathbf{X}\|] + k\sqrt{Var[\|\mathbf{X}\|]}\right],$$

for some $k \geq 0$*. In other words, an observation is considered to be an outlier in scale of level k if its norm does not lie within k standard deviations of its expected value.*

The notion of outliers in scale distinguishes the effects of a graph signal's magnitude from its spatial distribution. The identification of outliers solely based on their spatial distribution requires a stronger definition of outliers in distribution. Our first definition was called a *weak* outlier in distribution because it does not account for TV scaling quadratically with the graph signal's magnitude. In other words, since the TV scales quadratically with the entries of the vector $\mathbf{x}$, a vector with all entries doubled will have a higher TV, even though the relative values of the node signals remain the same. We therefore condition the expectation and variance of $\mathrm{TV}(\mathbf{X})$ with respect to its realized norm $\|\mathbf{X}\| = \|\mathbf{x}\|$ to define stronger notions of outliers.

Definition 21. *An observation* $\mathbf{x}$ *is considered to be a* strong distribution outlier of level k *or a* strong outlier in distribution of level k *(abbreviated as Strong-OID) if* $TV(\mathbf{x}) \notin [\mathscr{A}, \mathscr{B}]$*, where*

$$\mathscr{A} = \mathbb{E}\left[TV(\mathbf{X}) \mid \|\mathbf{X}\| = \|\mathbf{x}\|\right] - k\sqrt{Var\left[TV(\mathbf{X}) \mid \|\mathbf{X}\| = \|\mathbf{x}\|\right]}$$
$$\mathscr{B} = \mathbb{E}\left[TV(\mathbf{X}) \mid \|\mathbf{X}\| = \|\mathbf{x}\|\right] + k\sqrt{Var\left[TV(\mathbf{X}) \mid \|\mathbf{X}\| = \|\mathbf{x}\|\right]},$$

for some $k \geq 0$*. In other words, an observation is considered to be a strong outlier in distribution if its TV does not lie within k standard deviations of its expected value, conditioned on the realized norm* $\|\mathbf{X}\| = \|\mathbf{x}\|$.

In the subsequent sections and propositions, for analytical tractability we consider only the 1-norm for all graph signal vectors, i.e. $\|\mathbf{x}\| = \sum_{i=1}^{N} |x_i|$. This assumption is generally acceptable, as it is true that for many physical systems, the vertex signals is always a non-negative quantity (*e.g.*, delays at an airport, number of cars at an intersection, etc). Furthermore, we assume that all graph signals are non-negative, so the 1-norm for these graph signal vectors becomes equivalent to $\|\mathbf{x}\| = \sum_{i=1}^{N} x_i$. We justify this particular assumption as many relevant physical parameters such as airport delays are non-negative signals supported on nodes.

To summarize our various definitions of outliers, Figure 7-1 presents a graphical description of these definitions. Note that all these bounds can be empirically calculated when given a dataset $\mathscr{O}_M$. We try to derive analytical bounds for these outliers under certain assumptions on the distribution of the data. In particular, we will present our progress in computing bounds for Weak-OID when the data is a multivariate Gaussian. Deriving analytical bounds for strong outliers in distribution remains an open problem since the conditioning on the realized norm complicates constructing analytical bounds. Via simulation, we evaluate empirically-derived bounds for strong outliers in distribution, and show that the gap between the strong and weak outlier bounds depends on $\boldsymbol{\mu}$ and Σ. Thus, we will finally present an approach to obtain meaningful bounds when data is non-Gaussian and skewed in the end.

7.2.1 Interpreting the cause of an Outlier classification

One advantage of our method is the ability to provide interpretations at multiple stages of the outlier classification process. If $\mathbf{x} \in \mathbb{R}_{\geq 0}^{N \times 1}$ is determined to be a strong OID via the appropriate

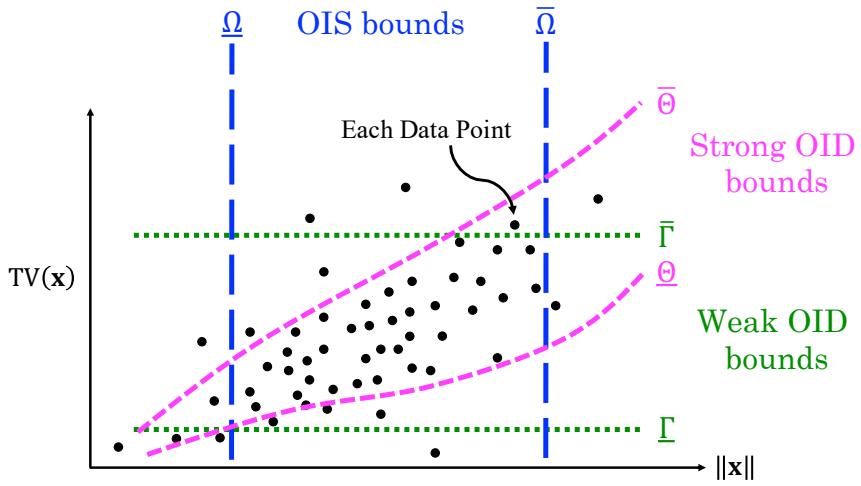


Figure 7-1: Notional representation of outlier bounds in the TV vs $\|\mathbf{x}\|$ space.

bounds, then we can compute the spectrum of the signal $\{\alpha_1^2, \ldots, \alpha_N^2\}$ using Definition 17. Using the spectral energy percentage contained in each eigenvector mode, we can retrieve the dominant eigenvector modes by selecting those with significant energy contributions. Typically, when a data point is classified as an outlier, it would have a higher share of its energy in the high-frequency eigenmodes. Thus, identifying the specific eigenmodes can help narrow down the components of the graph signal, and in particular the group of nodes, whose unexpected values caused the entire signal to be classified as an outlier.

7.3 Outlier Detection Bounds: The Case of Gaussian Signals

In this section, we assume that $\mathbf{X} \overset{iid}{\sim} \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. $\boldsymbol{\mu} \in \mathbb{R}^{N\times 1}$ is the vector of means and $\Sigma \in \mathbb{S}^{N\times N}$ is the positive semi-definite covariance matrix. We first present the bounds for outliers in scale. Then, we derive analytic expressions for the expectation and variance of the TV for the case with "complete information" (that is, ρ_{ij} is known) as well as the "partial information" case (where only bounds on ρ_{ij} are known).

While the values for the mean and variance of the TV can be obtained from the set of observations $\mathcal{O}_M$, our contribution lies in deriving analytical closed-form expressions for such bounds. This is useful for two reasons: (1) Even though we have accurate information about the means

and variances of signals at each node, there may be insufficient data points to reliably estimate the expectation and variance of TV; (2) We can obtain intuition regarding how the properties of the signals (data set), *i.e.* $\boldsymbol{\mu}$ and Σ, determine the bounds for outlier detection.

7.3.1 Bounds for outliers in scale

We derived bounds for $\mathbb{E}[\mathrm{TV}(\mathbf{X})]$ and $\mathrm{Var}[\mathrm{TV}(\mathbf{X})]$ in the previous subsection to examine weak outliers in distribution. Here we focus on bounds for outlier in scale. Recall that we are assuming $\mathbf{x}$ has non-negative entries, allowing for the redefinition of the 1-norm as $\|\mathbf{X}\|_1 = \sum_i X_i$. We have that the expectation of $\|\mathbf{X}\|$ is:

$$\mathbb{E}\left[\sum_i X_i\right] = \sum_i \mathbb{E}\left[X_i\right] = \sum_i \mu_i. \tag{7.6}$$

Similarly, the variance of $\|\mathbf{X}\|$ is:

$$\mathrm{Var}\left[\sum_i X_i\right] = \sum_i \mathrm{Var}\left[X_i\right] + \sum_{i\neq j} \rho_{ij}\sigma_i\sigma_j. \tag{7.7}$$

In the setting with complete information, since the correlations are known, the expression for the variance simplifies to $\mathrm{Var}\left[\sum_i X_i\right] = \mathbf{1}^\intercal \Sigma \mathbf{1}$. In the setting with partial information, if we know that the correlations are bounded by $0 \leq \nu_{ij} < \rho_{ij} < \varepsilon_{ij} \leq 1$, we get that

$$\sum_i \sigma_i^2 + \sum_{i\neq j} \nu_{ij}\sigma_i\sigma_j < \mathrm{Var}\left[\sum_i X_i\right] < \sum_i \sigma_i^2 + \sum_{i\neq j} \varepsilon_{ij}\sigma_i\sigma_j. \tag{7.8}$$

Finally, we note that the TV can be bounded by the Rayleigh quotient since the graph Laplacian is a Hermitian matrix. Let $\lambda_{\max}$ be the largest eigenvalue of L, then we have

$$\mathrm{TV}(\mathbf{x}) = \mathbf{x}^\intercal L \mathbf{x} \leq \lambda_{\max} \|\mathbf{x}\|_2^2 \leq \lambda_{\max} \|\mathbf{x}\|_1^2 \tag{7.9}$$

However, the upper bound provided by the Rayleigh quotient is not useful for outlier detection as it is extremely conservative in practice. The bounds that we have obtained are much tighter than the ones that would be derived via the Rayleigh quotient.

7.3.2 Expectation of TV

We first compute the expectation of the TV for a random graph signal $\mathbf{X}$. Note that computing the expectation does not require the distribution to be a multivariate Gaussian, only that it has a finite mean and variance.

$$\begin{aligned}\mathbb{E}[\mathrm{TV}(\mathbf{X})] &= \mathbb{E}\left[\frac{1}{2}\sum_{i\neq j}\left\{\rho_{ij}(X_i - X_j)^2\right\}\right] \\ &= \frac{1}{2}\sum_{i\neq j}\left\{\rho_{ij}\left(\mathbb{E}\left[X_i^2\right] + \mathbb{E}\left[X_j^2\right] - 2\,\mathbb{E}\left[X_iX_j\right]\right)\right\}\end{aligned} \tag{7.10}$$

Substituting $\mathbb{E}\left[X_iX_j\right] = \mu_i\mu_j + \rho_{ij}\sigma_i\sigma_j$ from (7.2) and $\mathbb{E}\left[X_i^2\right] = \mu_i^2 + \sigma_i^2$ in (7.10), we get

$$\mathbb{E}[\mathrm{TV}(\mathbf{X})] = \frac{1}{2}\sum_{i\neq j}\left\{\rho_{ij}\left[(\mu_i - \mu_j)^2 + (\sigma_i^2 + \sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j)\right]\right\}. \tag{7.11}$$

Equation (7.11) gives the contribution of each edge to the TV. The contribution of each edge – alternatively, of each unique pair of vertices – depends on the difference in signal means, variances, and correlation. We examine some specific cases that impose certain conditions on the means, variances, and correlations:

1. If there is no correlation between the signals at any two vertices, *i.e.* $\rho_{ij} = 0$ for all vertices $i \neq j$, then $\mathbb{E}[\mathrm{TV}(\mathbf{x})] = 0$.

2. If there is perfect correlation between the signals at any two vertices, *i.e.*, $\rho_{ij} = 1$ for all vertices $i \neq j$, then $\mathbb{E}[\mathrm{TV}(\mathbf{x})] = \frac{1}{2}\sum_{i\neq j}\left\{(\mu_i - \mu_j)^2 + (\sigma_i - \sigma_j)^2\right\}$.

3. If the means are the same for all signals, *i.e.*, $\mu_i = \mu_j$ for all vertices $i \neq j$, then $\mathbb{E}[\mathrm{TV}(\mathbf{X})] = \frac{1}{2}\sum_{i\neq j}\left\{\rho_{ij}\left[\sigma_i^2 + \sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j\right]\right\}$.

4. If all pairwise vertex signals have the same means ($\mu_i = \mu_j$), variances ($\sigma_i = \sigma_j = \sigma$), and correlation coefficient ($\rho_{ij} = \rho$) for all vertices $i \neq j$, then

$$\mathbb{E}[\mathrm{TV}(\mathbf{X})] = \sum_{i\neq j}\left\{\rho\sigma^2(1-\rho)\right\} = N(N-1)\rho\sigma^2(1-\rho). \tag{7.12}$$

7.3.3 Variance of TV

The computation of the variance of the TV is more involved. While we can derive a closed-form analytic equation, it cannot be expressed in a simple form. The outline of the derivation is as follows:

$$\mathrm{Var}\left[\mathrm{TV}(\mathbf{X})\right] = \mathbb{E}\left[\mathrm{TV}(\mathbf{X})^2\right] - \mathbb{E}\left[\mathrm{TV}(\mathbf{X})\right]^2. \tag{7.13}$$

The second term, $\mathbb{E}\left[\mathrm{TV}(\mathbf{X})\right]^2$, is known from (7.11). The first term can be expanded as

$$\mathbb{E}\left[\mathrm{TV}(\mathbf{X})^2\right] = \frac{1}{4}\mathbb{E}\left[\left(\sum_{i\neq j}\left\{\rho_{ij}(X_i - X_j)^2\right\}\right)^2\right]. \tag{7.14}$$

Expanding (7.14) and using linearity of the expectation operator gives us an expression in terms of $\mathbb{E}\left[X_i^4\right]$, $\mathbb{E}\left[X_i^3 X_j\right]$, and $\mathbb{E}\left[X_i^2 X_j^2\right]$. We know $\mathbb{E}\left[X_i^4\right] = \mu_i^4 + 6\mu_i^2\sigma_i^2 + 3\sigma_i^4$. The analytical expression for higher-order moments of the product of dependent Gaussian random variables was derived in [204], with a more implementable form that also accounts for non-zero means given in [205].

Proposition 2 (Isserlis (1918) and Kan (2008))**.** *Suppose* $\mathbf{X} = (X_1, \ldots, X_N)^\mathsf{T} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$*, where* Σ *is an* $N \times N$ *positive semi-definite matrix. For non-negative integers* s_1 *to* s_N*, we have*

$$\mathbb{E}\left[\prod_{i=1}^{N} X_i^{s_i}\right] = \sum_{\nu_1=0}^{s_1} \cdots \sum_{\nu_N=0}^{s_N} \sum_{r=0}^{[s/2]} \binom{s_1}{\nu_1} \cdots \binom{s_N}{\nu_N} \times \left\{ \frac{\left(\frac{\boldsymbol{h}^\mathsf{T}\Sigma\boldsymbol{h}}{2}\right)^r (\boldsymbol{h}^\mathsf{T}\boldsymbol{\mu})^{s-2r}}{r!(s-2r)!} \right\},$$

where $s = s_1 + \cdots + s_N$ *and* $\boldsymbol{h} = \left(\frac{s_1}{2} - \nu_1, ..., \frac{s_N}{2} - \nu_N\right)^\mathsf{T}$.

Proof. See [204] and [205]. □

Substituting the results from Proposition 2 in the expansion of (7.14), we can obtain an analytical expression for $\mathrm{Var}[\mathrm{TV}(\mathbf{X})]$ as a function of $\boldsymbol{\mu}$ and Σ. While the variance of the TV involves a significant number of terms and cannot be written in a simple algebraic form, it can be easily evaluated symbolically and numerically using a computer. We give the following case when we can evaluate the variance of the TV easily:

Proposition 3. *If* $\mathbb{E}[TV(\mathbf{X})] = 0$ *and* $\rho_{ij} \geq 0$, $\forall i, j$ *(or* $\rho_{ij} \leq 0$, $\forall i, j$*), then* $Var[TV(\mathbf{X})] = 0$.

Proof. Since $\text{TV}(\mathbf{X}) \geq 0$ (or $\text{TV}(\mathbf{X}) \leq 0$), $\mathbb{E}[\text{TV}(\mathbf{X})] = 0 \implies \text{TV}(\mathbf{X}) = 0$. Hence, we get $\text{Var}[\text{TV}(\mathbf{X})] = 0$. □

However, note that $\mathbb{E}[\text{TV}(\mathbf{X})] = 0$ is not a necessary condition for $\text{Var}[\text{TV}(\mathbf{X})] = 0$.

The analytic expressions we derived for the expectation and variance of the TV allow us to compute the bounds given in Definition 19, thereby identifying weak outliers in distribution.

7.3.4 Expectation and variance of TV given partial information

We consider the setting where the mean and variance of the signal at each vertex is known, *i.e.*, $\boldsymbol{\mu}$ and σ_i^2 is known for all vertices $i \in V$, but the correlation between any pair of vertex signals, ρ_{ij}, is not known precisely. This can happen in real systems if a data-reporting agent at each vertex can only obtain local vertex information and does not share information with other agents. In such cases, we can only obtain marginal information regarding individual means and variances at every vertex. While the exact interdependency between two pairwise vertices may not be available, we develop a theory regarding detecting outliers in graph signals using only an approximate bound on the nature of the interaction. Specifically, we assume that the observations are drawn from a multivariate Gaussian distribution with a fixed $\boldsymbol{\mu} \in \mathbb{R}^{N\times 1}$ and $\Sigma \in \mathbb{S}^{N\times N}$, but the precise value of ρ_{ij} is unknown.

The rest of the results that we prove will assume that all the correlations are non-negative, i.e., $\rho_{ij} \geq 0 \; \forall i, j \in V$. In fact, we only require is that the correlations are of the same sign, but do not consider the case. The same sign of the correlation is required so that we can clearly ascribe any increase or decrease in the TV to a corresponding increase or decrease in the gap between x_i and x_j.

We derive tight bounds on $\mathbb{E}[\text{TV}(\mathbf{X})]$ and $\text{Var}[\text{TV}(\mathbf{X})]$ when we are only given bounds on each correlation coefficient $0 \leq \nu_{ij} < \rho_{ij} < \varepsilon_{ij} \leq 1$. We first present the setup leading to the two propositions that quantify the corresponding bounds on $\mathbb{E}[\text{TV}(\mathbf{X})]$ and $\text{Var}[\text{TV}(\mathbf{X})]$.

Every edge between unique pairs of vertices i and j in a graph obtained from the set of observations $\mathscr{O}_M$ is assigned a weight $r_{ij|\mathscr{O}_M}$. Note that $r_{ij|\mathscr{O}_M}$ is a consistent estimator of ρ_{ij}, although

the estimator is biased [206] with

$$\mathbb{E}[r_{ij|\mathscr{O}_M}] = \rho_{ij}\left(1 - \frac{1-\rho_{ij}^2}{2M} + O\left(\frac{1}{M^2}\right)\right). \tag{7.15}$$

Additionally, $r_{ij|\mathscr{O}_M}$ is a random variable with a valid probability density function $f(r_{ij|\mathscr{O}_M} \,|\, \rho_{ij})$ that has an explicit form expressible in terms of the Euler gamma function $\Gamma(x)$ [207, 208]. It is important to note that the probability density function is dependent only on the number of observations M, and independent of any new realizations of the random variables X_i and X_j. We will make use of this fact in the proofs for our propositions.

With the above setup, we can now redefine TV for an *unobserved* graph signal vector $\mathbf{X} \sim \mathscr{N}(\boldsymbol{\mu}, \Sigma)$ with respect to the graph Laplacian $L \in \mathbb{S}^{N\times N}$ constructed using $\mathscr{O}_M$. $\mathrm{TV}(\mathbf{X})$ is thus a derived random variable. In Proposition 4, we provide bounds on $\mathbb{E}[\mathrm{TV}(\mathbf{X})]$, and in Proposition 5, we proceed to derive bounds on $\mathrm{Var}[\mathrm{TV}(\mathbf{X})]$.

Proposition 4. *Suppose that* $0 \leq \nu_{ij} < \rho_{ij} < \varepsilon_{ij} \leq 1$ *for all unique pairs of vertices* $i, j \in V$*. Then, there exists scalars* δ_1 *and* δ_2*, with* $\delta_2 \geq 0$*, such that* $\max\{0, \delta_1\} \leq \mathbb{E}[TV(\mathbf{X})] < \delta_2$*.*

Proof. Since $r_{ij|\mathscr{O}_M}$ is a random variable dependent only on M previous observations, and X_i, X_j are currently unobserved random variables, the expectation operator factorizes over the expression for the TV:

$$\begin{aligned}\mathbb{E}[\mathrm{TV}(\mathbf{X})] &= \frac{1}{2}\sum_{i\neq j}\left\{\mathbb{E}\left[r_{ij|\mathscr{O}_M}\right]\mathbb{E}\left[(X_i - X_j)^2\right]\right\} \\ &= \frac{1}{2}\sum_{i\neq j}\left\{\mathbb{E}\left[r_{ij|\mathscr{O}_M}\right]\left((\mu_i-\mu_j)^2 + \sigma_i^2 + \sigma_j^2 - 2\rho_{ij}\sigma_i\sigma_j\right)\right\}.\end{aligned} \tag{7.16}$$

Since the bias of $r_{ij|\mathscr{O}_M}$ is given in (7.15), for any M, there exists a $\gamma_{ij} > 0$ that is a function of M (and $\lim_{M\to\infty} \gamma_{ij} = 0$) such that

$$\left|\mathbb{E}\left[r_{ij|\mathscr{O}_M}\right] - \rho_{ij}\right| < \gamma_{ij} \tag{7.17}$$

$$\Leftrightarrow \rho_{ij} - \gamma_{ij} < \mathbb{E}[r_{ij|\mathscr{O}_M}] < \rho_{ij} + \gamma_{ij}. \tag{7.18}$$

We can use the fact that $\nu_{ij} < \rho_{ij} < \varepsilon_{ij}$ in order to rewrite the bounds of (7.18),

$$\max\left\{0, \nu_{ij} - \gamma_{ij}\right\} \leq \mathbb{E}\left[r_{ij|\mathscr{O}_M}\right] < \varepsilon_{ij} + \gamma_{ij}. \tag{7.19}$$

The maximum operator is included since $\nu_{ij} - \gamma_{ij}$ can be negative, but we know that the expectation of a non-negative random variable $\mathrm{TV}(\mathbf{X})$ is bounded below by 0. We aim to use (7.19) in conjunction with the bounds on ρ_{ij} to bound (7.16). We focus first on deriving the upper bound; such an upper bound is given by evaluating (7.16) for the largest-possible contributions from the positive terms, and the smallest-possible deductions from the negative term. This gives:

$$\begin{aligned}\mathbb{E}[\mathrm{TV}(\mathbf{X})] = \frac{1}{2}\sum_{i\neq j}\Big\{&\underbrace{\mathbb{E}\left[r_{ij|\mathscr{O}_M}\right]}_{<\varepsilon_{ij}+\gamma_{ij}}\left((\mu_i-\mu_j)^2+\sigma_i^2+\sigma_j^2\right)\\ &-2\underbrace{\mathbb{E}\left[r_{ij|\mathscr{O}_M}\right]\rho_{ij}}_{<\nu_{ij}\max\left\{0,\nu_{ij}-\gamma_{ij}\right\}}\sigma_i\sigma_j\Big\}.\end{aligned} \tag{7.20}$$

Therefore, this gives the upper bound $\mathbb{E}[\mathrm{TV}(\mathbf{X})] < \delta_2$, where

$$\delta_2 = \frac{1}{2}\sum_{i\neq j}\left\{\tilde{\varepsilon}_{ij}\left((\mu_i-\mu_j)^2+\sigma_i^2+\sigma_j^2\right) - 2\tilde{\nu}_{ij}^{+}\nu_{ij}\sigma_i\sigma_j\right\} \tag{7.21}$$

along with rewriting $\tilde{\varepsilon}_{ij} = \varepsilon_{ij} + \gamma_{ij}$ and $\tilde{\nu}_{ij}^{+} = \max\left\{0, \nu_{ij} - \gamma_{ij}\right\}$.

To get the lower bound, we evaluate (7.16) for the smallest-possible contribution from the positive terms and the largest-possible contribution in terms of magnitude from the negative terms. This gives $\mathbb{E}[\mathrm{TV}(\mathbf{X})] > \delta_1$, where

$$\delta_1 = \frac{1}{2}\sum_{i\neq j}\left\{\tilde{\nu}_{ij}^{+}\left((\mu_i-\mu_j)^2+\sigma_i^2+\sigma_j^2\right) - 2\,\tilde{\varepsilon}_{ij}\varepsilon_{ij}\sigma_i\sigma_j\right\} \tag{7.22}$$

Note that δ_1 and δ_2 are functions of the bounds on ρ_{ij}^{+} and M, so we have that

$$\lim_{\substack{\nu_{ij}\to\rho_{ij}^{+}\\ \varepsilon_{ij}\to\rho_{ij}^{+}}}(\delta_1) = \lim_{\substack{\nu_{ij}\to\rho_{ij}^{+}\\ \varepsilon_{ij}\to\rho_{ij}^{+}}}(\delta_2) = \mathbb{E}[\mathrm{TV}(\mathbf{X})]. \tag{7.23}$$

□

Proposition 5. *Suppose* $0 \leq \nu_{ij} < \rho_{ij}^{+} < \varepsilon_{ij} \leq 1$ *for all unique pairs of vertices* $i, j \in V$*. Then, there exists scalars* δ_3 *and* δ_4*, with* $\delta_4 \geq 0$*, such that* $\max\{0, \delta_3\} \leq Var[TV(\mathbf{X})] < \delta_4$.

Proof. The idea behind the proof is similar to the proof for Proposition 4. We expand $\text{Var}[\text{TV}(\mathbf{X})]$ as done in (7.13) and (7.14). Proposition 2 can then be used to obtain the appropriate higher-order moments. This gives $\text{Var}[\text{TV}(\mathbf{X})]$ as a scalar quantity that depends on $\mathbb{E}\left[r_{ij|\mathscr{O}_M}^{+}\right]$ and ρ_{ij}^{+}. Finally, these two terms can be bounded appropriately to obtain the desired bounds on $\text{Var}[\text{TV}(\mathbf{X})]$. □

Using Propositions 4 and 5, we can compute the worst-case bounds for weak outliers in distribution of level k, given by $\text{TV}(\mathbf{x}) \notin \left[\max\left\{0, \delta_1 - k\sqrt{\delta_4}\right\},\ \delta_2 + k\sqrt{\delta_4}\right] \implies \mathbf{x}$ is a weak outlier in distribution.

7.4 Understanding the Bounds Using Simulations

We aim to convey two ideas via simulations: First, we compute the bounds on strong distribution outliers using simulations, and compare them against the theoretically-derived weak distribution outlier bounds. We observe that the difference between these two bounds depends on the underlying $\boldsymbol{\mu}$ and Σ of the data, and we show two examples to highlight that dependency. Second, we provide some more intuition on the partial information case by empirically evaluating the mean and variance of the TV for a range of ρ. We observe non-monotonic variations with ρ which are difficult to predict *a priori*, highlighting the importance of our analytical bounds from Propositions 4 and 5.

7.4.1 Strong and weak bounds on TV for synthetic signals

We set up two simulations to evaluate the gap between the strong and weak outliers in distribution. For both simulations, $M = 1 \times 10^6$ data points are generated using a multivariate Gaussian distribution, but they differ in the mean $\boldsymbol{\mu}$ and covariance Σ. This data is plotted on a $\text{TV}(\mathbf{x})$ versus $\|\mathbf{x}\|$ axis, and all outlier bounds are computed empirically. In particular, we compute $\hat{\mathbb{E}}\left[\text{TV}(\mathbf{X}) \mid \|\mathbf{X}\| = \|\mathbf{x}\|\right]$ and $\widehat{\text{Var}}\left[\text{TV}(\mathbf{X}) \mid \|\mathbf{X}\| = \|\mathbf{x}\|\right]$ by appropriately binning along the $\|\mathbf{x}\|$ axis. In the same plot, we add the theoretically derived bounds too, and also color each observation based on the likelihood of that observation given the multivariate Gaussian pdf.

In both the simulations (Figure 7-2 and 7-3) we see the value of our outlier detection techniques even when the underlying distribution of the data is known. This is because a simple pdf-based outlier detection technique would classify all the black-colored points as outliers. However, our technique distinguished between the low likelihood data points based on the change in the relative values across the nodes.

In the first simulation, we choose $N = 2$, $\sigma_1 = \sigma_2 = 1$, $\rho_{12} = 0.5$ and $\boldsymbol{\mu} = (545.34, 582.13)^\intercal$. The results in Figure 7-2 shows that the bounds for the weak and strong outliers in distribution are very close. Thus, the computation of the weak distribution bounds theoretically would be a good approximation to the strong distribution bounds.

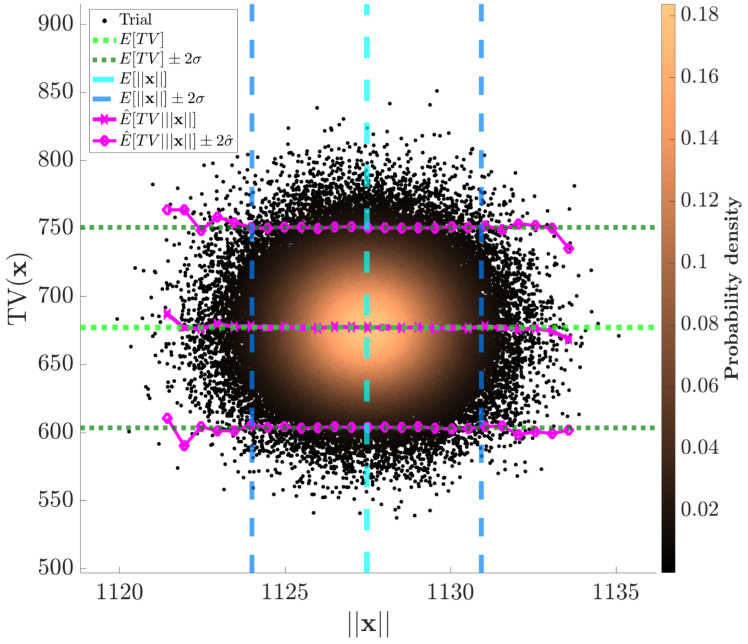


Figure 7-2: TV versus 1-norm of the graph signal for a generic bi-vertex graph, with scale outlier, weak outlier in distribution, and empirically-derived strong outlier bounds. Each observation $(\|\mathbf{x}\|, \mathrm{TV}(\mathbf{x}))$ is colored by its probability density $f_{\mathbf{X}}(\mathbf{X} = \mathbf{x})$.

For the second simulation, we have that $N = 30$, and we estimate $\boldsymbol{\mu}$ and Σ from a real-world data set of air traffic delay signals (see Section 7.6). We see that the weak outliers in distribution bounds tend to be overly-liberal (for low values of $\|\mathbf{x}\|$) or overly-conservative (for large values of $\|\mathbf{x}\|$) in terms of distinguishing outliers. This behavior is expected, as the weak bounds are not conditioned on $\|\mathbf{x}\|$. For future work, we are interested in deriving an analytical strong outlier in distribution bound. Figure 7-3 also shows the usefulness of TV as an outlier-distinguishing metric

over simply using the density of the underlying distribution; the former takes into account pairwise interactions and the strength of that interaction, whereas the latter does not.

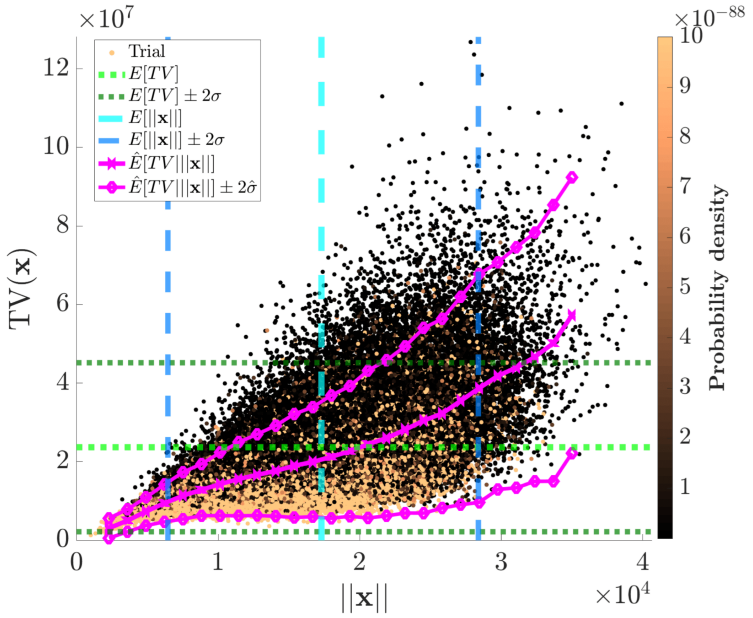


Figure 7-3: TV versus 1-norm of simulated graph signals within a 30-vertex graph; data set from the US air transportation network.

7.4.2 Expectation and variance of TV as a function of ρ

Through simulations, we demonstrate the utility of our analytical bounds on $\mathbb{E}[\mathrm{TV}(\mathbf{X})]$ and $\mathrm{Var}[\mathrm{TV}(\mathbf{X})]$ for the setting with partial information regarding correlations. Using the results from Section 7.3.2, it is theoretically possible – although computationally intractable – to evaluate the exact expectation and variance of the TV for a set of discretized values of $\rho_{ij} \in \left[\nu_{ij}, \varepsilon_{ij}\right] \subseteq [0,1]$ given the signal mean and variance at each vertex. The intractability is apparent in two locations: First, the search space is exponential in the number of edges, and a discretization of ρ_{ij} into N_ρ intervals for each edge requires $N_\rho^{N\times(N-1)}$ evaluations for $\mathbb{E}[\mathrm{TV}(\mathbf{X})]$ and $\mathrm{Var}[\mathrm{TV}(\mathbf{X})]$. Secondly, the evaluated functions are not convex in ρ_{ij}, indicating that methods such as gradient descent cannot be used to obtain worst-case bounds. We demonstrate this non-monotonic behavior in Figure 7-4. However, since our bounds derived in Propositions 4 and 5 are tight, this allows for a computationally efficient strategy to evaluate the optimization over the search space of $0 \leq \nu_{ij} < \rho_{ij} < \varepsilon_{ij} \leq 1$.

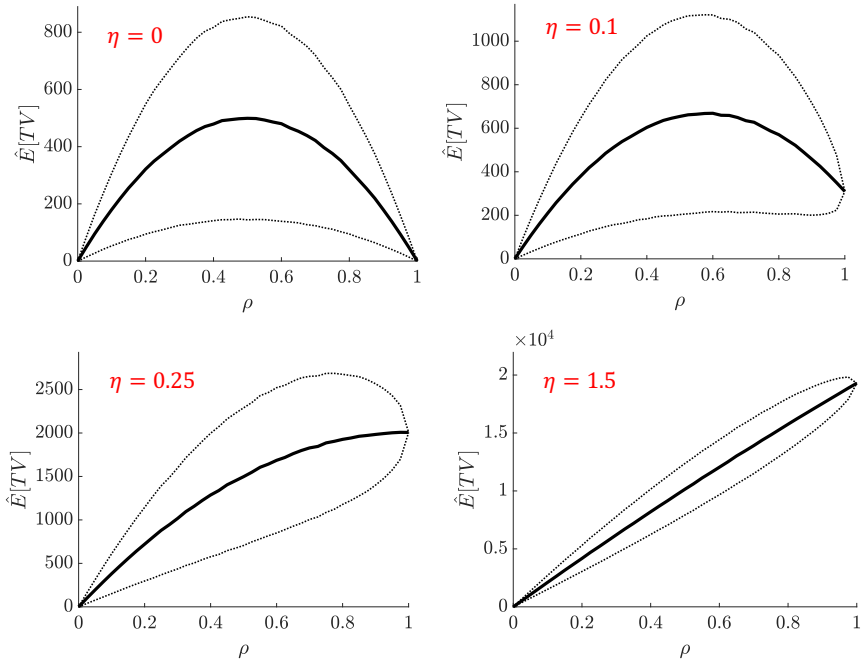


Figure 7-4: Empirically-derived curves for the expectation and variance of TV as a function of correlation and parameterized by $\boldsymbol{\mu}$ via η.

For the simulations depicted in Figure 7-4, we draw $M = 5 \times 10^4$ data points from a multivariate Gaussian distribution with $N = 5, \sigma_i = 10, \forall i$, and $\rho_{ij} = \rho, \forall i, j$ where we vary ρ independently within the interval $[0, 1]$. We consider four examples, each initialized with a different $\boldsymbol{\mu}$. These four examples have differing ranges for the values of the vertex signal means, parameterized by a "tolerance factor" η. We choose $\eta \in \{0, 0.1, 0.25, 1.5\}$ and $\mu_i \overset{iid}{\sim} 100(1-\eta) + 200\eta X_U$, where $X_U \overset{iid}{\sim} \text{Unif}(0,1)$. Higher η indicates that signals have higher baseline difference in terms of magnitudes across a pair of vertices.

If $\eta = 0$, then $\mu_i = \mu_j$ for all unique pairs of vertices $i, j \in V$, and we degenerate to the case depicted in (7.12) where $\mathbb{E}[\text{TV}(\mathbf{X})]$ is quadratic in ρ. For larger tolerances η, we see that $\mathbb{E}[\text{TV}(\mathbf{X})]$ becomes monotonic and closer to a linear function, as depicted in Figure 7-4. The dotted lines in Figure 7-4 show the $\pm\sqrt{\text{Var}[\text{TV}(\mathbf{X})]}$ bounds around the expected TV. Figure 7-4 highlights the dynamic behavior of the bounds as a function of ρ, even in our relatively constrained setting of $\rho_{ij} = \rho, \forall i, j \in V$.

We conclude this section with a few remarks. The simulations confirm that empirical results shown in Figure 7-4 match theoretical predictions (not shown for simplicity) from the previous

section. At $\rho = 1$, all four examples had zero variance since $\sigma_i = \sigma$, $\forall i \in V$. If the variances were allowed to vary, *i.e.* $\sigma_i \neq \sigma_j$, then $\text{Var}[\text{TV}(\mathbf{X})] \neq 0$ at $\rho = 1$. Furthermore, we emphasize that our derived bounds are tight and characterize the exploration of the entire search space of variations in correlation. However, not all choices of ρ_{ij} lead to a valid positive semi-definite covariance matrix Σ. Enforcing this constraint analytically may result in tighter bounds, and is an open problem.

7.5 Outlier Detection Bounds: Skewed Distributions

From the example in Figure 7-3, we can clearly see that a non-Gaussian data (airport delay signals in this case) results in a non-symmetric distribution of $\|\mathbf{x}\|$, $TV(\mathbf{x})$ and $TV(\mathbf{x})|\,\|\mathbf{x}\| = C$. Thus, in this section, we refine our definitions of outliers and propose a non-parametric approach based on the interquartile range (IQR). We explicitly account for skewed distributions by adjusting the bounds based on the *medcouple* statistic [209, 210]. This turns out to be critical in our application, where it would not be appropriate to assume symmetry in the TD or TV distributions since both are non-negative quantities. Given a univariate sample set $\{y_1, y_2, \ldots, y_K\}$, we define the value of its first quartile as Q_1, the median of the sample set as Q_2, and the third quartile value as Q_3. The IQR is the difference between the third and first quartile, i.e. $IQR = Q_3 - Q_1$. The medcouple statistic is defined as follows:

$$MC = \underset{y_i \leq Q_2 \leq y_j}{\text{med}} h(y_i, y_j), \tag{7.24}$$

note that we take a median conditioned on $y_i \leq Q_2 \leq y_j$, where for all $y_i \neq y_j$, the kernel function $h(\cdot,\cdot)$ is given by:

$$h(y_i, y_j) = \frac{(y_j - Q_2) - (Q_2 - y_i)}{y_j - y_i}. \tag{7.25}$$

The medcouple statistic is a robust measure of skewness [209], and will be used to adjust the IQR in order to form outlier bounds for asymmetric distributions [210]. For simplicity, we proceed with our analysis for the case where all distributions are skewed to the right, i.e. $MC > 0$. Thus, we only utilize the adjusted box-plot formulas for the right-skewed case. The reference [210] presents more details for the left-skewed case.

The first notion of a graph signal outlier that we will define identifies *outliers in scale*, meaning that $\mathbf{x}$ is an outlier based solely on the magnitude of the TD, and not the distribution. This notion

of an outlier is the simplest, most commonly used, and most intuitive. This definition states that the graph signal is an outlier in scale if the 1-norm of the graph signal lies outside a skew-adjusted empirically determined central region:

Definition 22. *A data point* $\mathbf{x}$ *is classified as a* skew-adjusted outlier in scale (S-OIS) *if*

$$\|\mathbf{x}\| \notin \left[\underline{\Omega},\, \overline{\Omega}\right],$$

where the lower bound $\underline{\Omega} \coloneqq Q_1 - 1.5e^{-4\times MC}IQR$*, the upper bound* $\overline{\Omega} \coloneqq Q_3 + 1.5e^{3\times MC}IQR$*, and the IQR and MC* ≥ 0 *are defined on the set* $\left\{\left\|\mathbf{x}^{(1)}\right\|, \ldots, \left\|\mathbf{x}^{(M)}\right\|\right\}$.

Definition 23. *A data point* $\mathbf{x}$ *is classified as a* weak skew-adjusted outlier in distribution (weak S-OID) *if*

$$TV(\mathbf{x}) \notin \left[\underline{\Gamma},\, \overline{\Gamma}\right]$$

where the the lower bound $\underline{\Gamma} \coloneqq Q_1 - 1.5e^{-4\times MC}IQR$*, the upper bound* $\overline{\Gamma} \coloneqq Q_3 + 1.5e^{3\times MC}IQR$*, and the IQR and MC* ≥ 0 *are defined on the set* $\left\{\mathbf{x}^{(1)\intercal}L\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)\intercal}L\mathbf{x}^{(M)}\right\}$.

Definition 24. *A data point* $\mathbf{x}$ *is classified as a* strong skew-adjusted outlier in distribution (strong S-OID) *if*

$$TV(\mathbf{x}) \notin \left[\underline{\Theta},\, \overline{\Theta}\right]$$

where the lower bound $\underline{\Theta} \coloneqq Q_1 - 1.5e^{-4\times MC}IQR$*, the upper bound* $\overline{\Theta} \coloneqq Q_3 + 1.5e^{3\times MC}IQR$*, and the IQR and MC* ≥ 0 *are defined on the set* $\left\{TV(\mathbf{x}^{(i)}) \mid TV(\mathbf{x}^{(i)}) \in [TV(\mathbf{x}) \pm \varepsilon]\ \forall i\right\}$ *for some interval width* ε.

Note that all these three skew-corrected outlier definitions are based on data and can be empirically calculated given $\mathscr{O}_M$. No assumption is made regarding the distributions of the graph signals or their conditional distributions. One consideration to keep in mind about the strong S-OID is to choose the interval width ε such that there are sufficient data points in each interval to compute the medcouple statistic robustly.

7.6 Applications to Airport Delays

High delays at major airports are known to have significant economic impacts; when analyzing system performance involving multiple airports and routes, airport-centric delay metrics, such as the sum of all inbound arrival delays and outbound departure delays, are commonly used. However, due to the strong underlying network connectivity, the delays at some airports are highly correlated with delays at other airports. This could be due to geographic proximity, traffic volumes, delay propagation, airline schedules, and traffic flow management procedures. For example, nor'easter systems typically affect several East Coast airports, resulting in correlated delays. There is consequently the notion of a typical or *expected* spatial distribution of delays, reflected by the statistical correlations between delays at different airports. We illustrate this notion with the example of Boston Logan International (BOS) and New York LaGuardia (LGA) airports. The delays at these airports are positively correlated, due to the high traffic volume between them, as well as their geographic proximity. When either airport experiences high delays, we expect high delays at the other airport as well. However, high delays at one of the airports but low delays at the other would be an unexpected occurrence.

Thus, airport delays can take on two forms: (1) The obvious problematic scenario when delays are high across all airports; (2) a subtler scenario when delays are high, but with an unexpected distribution across a specific set of airports. The former corresponds to outliers in scale and can be identified via classical metrics, whereas the latter corresponds to outliers in distribution, and can be identified by the methods presented in this chapter. Two disruptive events could result in the same total delay, but affect very different airports spatially, resulting in different operational impacts. The overarching theme of our work to identify and analyze days in terms of both the magnitude and the spatial distribution of delays. Such an inventory is essential for developing playbooks that will mitigate the element of surprise for controllers and flow managers due to unexpected delay distributions.

7.6.1 Correlation networks

We obtain delay statistics from the FAA Aviation System Performance Metrics (ASPM) database for the time period 2008 to 2017 [103]. The analysis is limited to the busiest 30 airports in the US

(FAA Core 30); we then compute the total delay at each of these airports during each day, defined as 0000Z to 2359Z. The total delay at an airport (i.e., a node in the graph) is the sum of the arrival and departure delays of all flights at that airport during the day. Consequently, we obtain 3,653 graph signal vectors (each of dimension 30), one corresponding to each day in the data set. The edge weights of the graph are the sample Pearson correlation coefficients based on the 10 years of data. It is worth noting that all correlations estimated from data are strictly positive. Thus, we have a graph with 30 nodes, $\binom{30}{2} = 435$ edges, and 3,653 instances of delay signal vectors on these nodes.

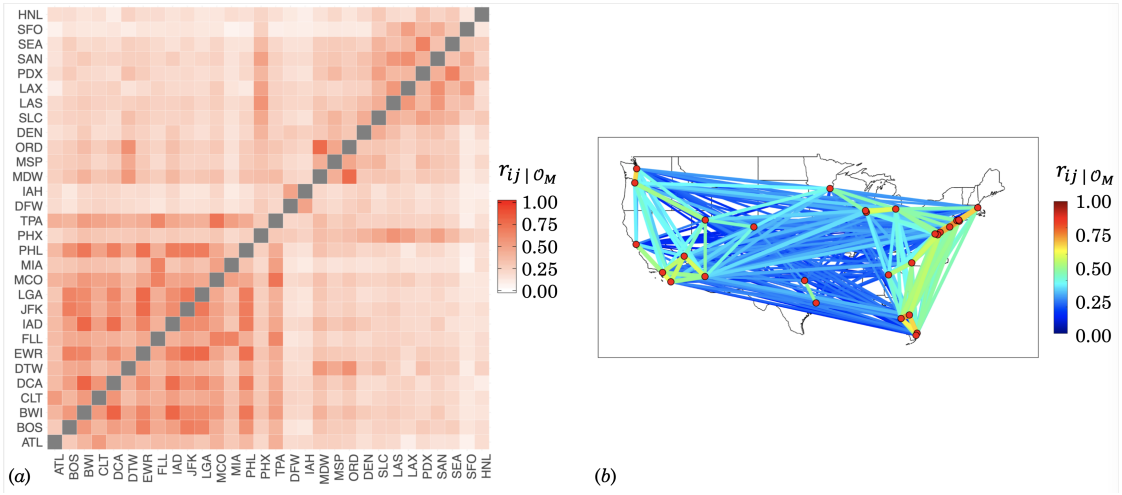


Figure 7-5: (a) Heat map displaying the delay correlation between the top 30 airports; (b) Correlations shown with geographical context. Higher correlations are also emphasized with wider lines

We visualize the resultant correlation matrix in Figure 7-5 using a heat map (left) and as a network (right). We note that the east coast airports have a high delay correlation among them. This is also true for the west coast airports. The main reasons for the high correlation among these two groups are the high flight traffic, and geographical proximity. This means that a delayed flight taking off in the east coast (or west coast) would most likely go to another east coast airport (or west coast) and continue to propagate the delay there. Geographical proximity also means that these airports are likely to experience the same weather disruptions or the same traffic management initiatives that result in flight delays. One example of geographical proximity playing a major role in delay correlations is the case of O'Hare (ORD) and Midway (MDW). These two Chicago-area airports are less than 20 miles away with no commercial traffic between them. However, their

delays are highly correlated.

7.6.2 Identifying outliers

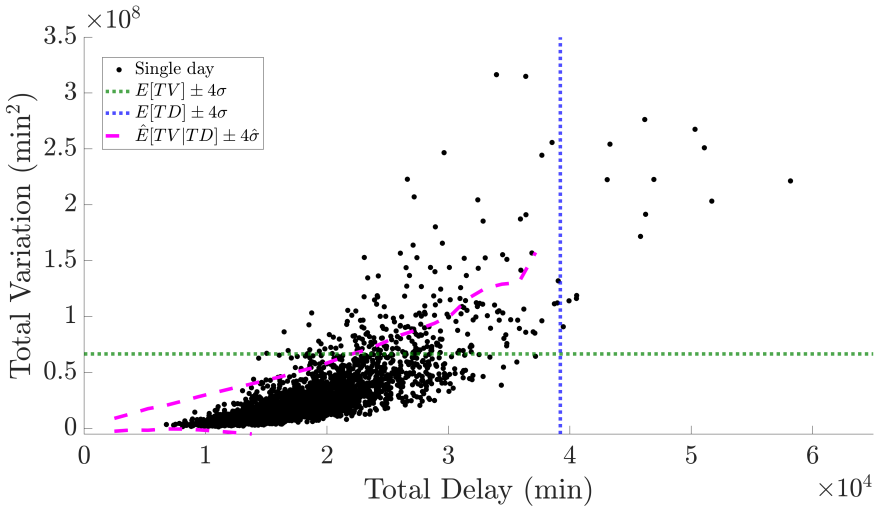


Figure 7-6: TV versus TD for all days in 2008-2017 with level $k = 4$ weak and strong outlier bounds demarcated.

Figure 7-6 plots the TV and TD for the airport delay graph signals for each day in the 2008-2017 data set (3,653 days). The bounds for outliers in scale as well as for the weak and strong outliers in distribution are computed for level $k = 4$ and plotted. It is worth noting that the lower bounds for the outliers in scale and the weak outlier in distribution are negative and not plotted. This highlights the significant level of conservatism in these bounds in relation to the thresholds for strong outliers in distribution. The primary factor that makes the weak bounds conservative is that they are not dependent on the TD, which does not allow them to capture the increasing variance in the TV for higher values of TD.

We summarize the statistics ontained from Figure 7-6: 167 days (4.6 % of days) were classified as strong distribution outliers, 221 days (6.0% of days) were classified as weak outliers in distribution only, no days were classified as outliers in scale only, and 14 days (0.4% of days) were classified as both weak outliers in distribution as well as outliers in scale.

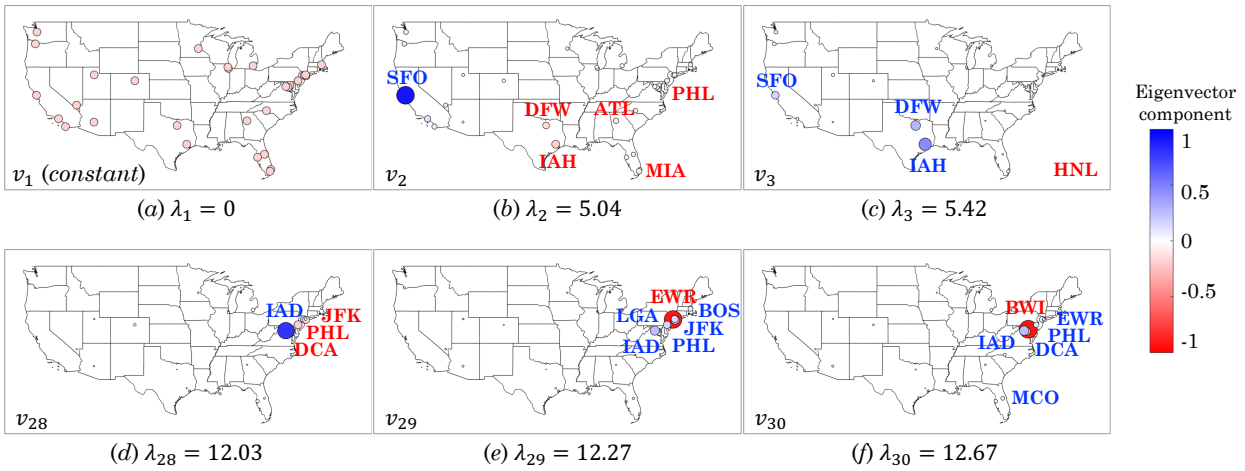


Figure 7-7: Top eigenvector modes.

7.6.3 Spectral analysis: Eigen modes and spectral energy

We compute the Laplacian L from the correlation matrix plotted in Figure 7-5, and compute its eigenvectors ($\lambda_1 < \cdots < \lambda_{30}$) and eigenvector modes ($v_1, \cdots v_{30}$). These eigenvector modes form the basis for the space of airport delay signals. An airport delay graph signal vector for any day can be decomposed into linear combinations of eigenvector modes v_1 through v_{30}. The eigenvector modes v_i corresponding to higher indices i are said to be more energetic and have a higher TV.

In Figure 7-7, we plot the three most energetic, and three least energetic eigenvector modes. These figures describe the basic relative delay patterns that can be combined to reconstruct the system delay. Hence, the sign of the eigenvector value at an airport does not indicate whether it contributes to increasing or decreasing delays. Rather, all it means is that airports with the same sign contribute in a similar way to the total delay signal. The eigenvector modes that are smooth (v_1, v_2, and v_3) correspond to expected delay patterns. For example, v_2 indicates that it is expected for SFO delays to have an opposite trend compared to DFW, IAH, ATL, PHL, and MIA. In contrast, the highly energetic eigenvector modes, v_{28}, v_{29}, and v_{30} represent rarer, more unexpected delay patterns. For example, v_{29} indicates that it is very unexpected for the delays at EWR to trend opposite to the other airports in the east coast (BOS, PHL, and IAD) and the other New York City airports themselves (JFK, LGA).

For each day in 2008 through 2017, we compute the spectral energy contributions of all 30 eigenvector modes, and plot the average contributions across the entire 10-year time frame in

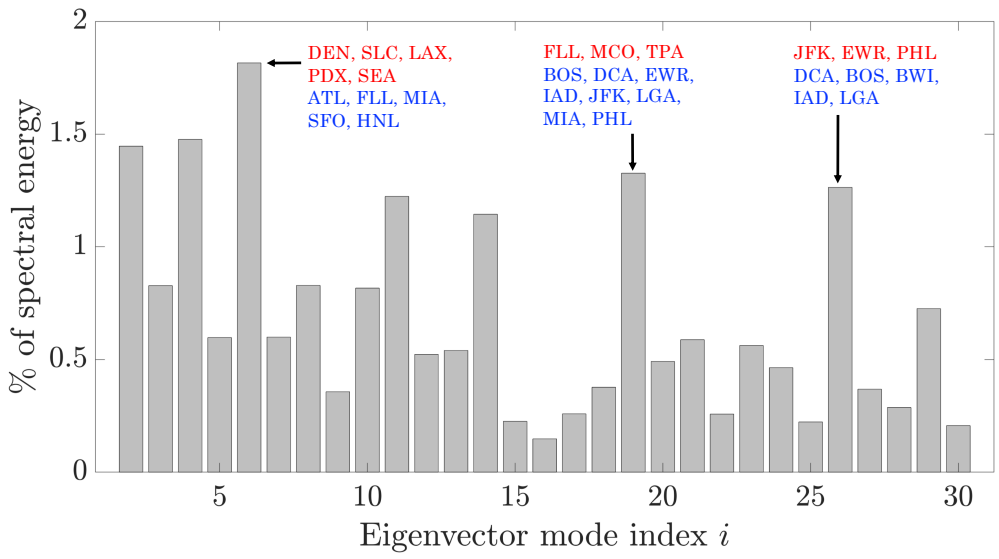


Figure 7-8: Spectral Energy.

Figure 7-8. The first mode v_1, which is the constant mode, accounts for 80% of the energy, and we only plot the energy contribution of the remaining modes v_2 through v_{30}. A higher percentage of spectral energy contribution indicates that a particular eigenvector mode – and hence, a particular delay pattern – contributes more to the overall delay state $\mathbf{x}$ of a particular day.

Figure 7-8 shows the percentage of spectral energy distributed over all the modes. We explicitly point out three modes with high spectral energy, and list out the airports with a significant contribution to the eigenvector. Mode v_{26} contributes approximately 1.2% of the spectral energy and indicates opposite delay trends in two groups of very geographically proximate east coast airports (JFK, EWR and PHL on one group, and DCA, BWI, IAD and BOS in the other group). Another example of a dominant mode, with over 1.3% of the average spectral energy is v_{19}, which indicates wide variations in an otherwise tightly coupled set of east coast airports.

7.6.4 Extensions

The techniques presented in this chapter have been extended to further applications in the context of airport delays. A list of some of these works are presented below:

- Applications to other networks: We consider airline-specific networks (American, Delta, United, and Southwest airlines) in [211]. We also compare the US airport network with the

Chinese airport network in [212]

- Relation to system disruptions: We have correlated the occurrences of these outliers with other disruptions in the system, such as hurricanes, winter storms and airport outages [211]. Along those dimensions, we also evaluate the correlation of these delay outliers on flight cancellations in [211]

- Dynamics of disruptions and recovery: The distribution of airport delays can be shown to be an additional dimension to quantify, analyze, and predict disruptions and recoveries in the national airspace. In particular, in [213] we can show that outliers can be used to define and identify disruption-recovery trajectories. Finally, in [214], we discuss potential control strategies to nudge the system from a disrupted state (i.e., outlier state) to a nominal state that has a more expected distribution of delays across airports.

Chapter 8

Conclusions

This dissertation developed techniques to model the evolution of airport delays using ideas from switched-systems theory, graph signal processing, and machine learning. The key idea behind our modeling approach was to (i) simplify the complex network interactions by identifying a small, finite set of representative network topologies using graph clustering, (ii) use historical flight delay data to learn the model parameters, and (iii) identify an appropriate topology transition policy to model the dynamics on time-varying networks. This resulted in the Markov jump linear system (MJLS) model for airport delays. We developed this delay model for the US, validated it, and demonstrated its superior predictive performance in comparison to other benchmarks. We then used the MJLS model to identify appropriate interventions that minimize delays, and developed novel controllers for regulating delays. Our findings suggest that (i) optimal interventions that target highly-connected airports at Atlanta, San Francisco and Chicago can provide maximum system-wide benefits, and (ii) the most effective time for such interventions is between 11 and 2 pm ET. Finally, we developed a metric (called the total variation) to analyze not just the total delays in the system, but also the spatial distribution of delays across the airports.

While this dissertation adopted a network-centric perspective on air transportation, it is important to note that several optimization and control techniques have long been an integral part of air traffic management. Successful applications of such techniques include airport surface congestion management [215, 216, 21], traffic flow management through large-scale optimization [217, 218], tactical conflict resolution [174, 219], equitable resource allocation [220, 221], and airline operations scheduling [222, 172]. Our methods add to this body of literature by offering a unique feature

– scalability. Methods that scale for large-problem instances are important to ensure safety and enable the growth in manned and unmanned air traffic operations over the coming years (see Table 8.1).

Type of traffic	2018	2035
Commercial flights	80	156
Urban air mobility	0	2,500
Delivery drones	0	16,667
Inspector drones	1	58
Hobby drones	12	44

Table 8.1: Estimates of hourly air traffic over Paris [3]

Although network models can analyze large systems, several advances are needed to fully realize their potential for practical decision-making. Some future directions of research are listed below:

- **Multilayer networks** promise to deliver more accurate and comprehensive representations of air transportation system dynamics (e.g., with separate and interacting network layers for delays, cancellations, traffic flows, and capacity); however, data-driven modeling, analysis and control of such systems remains an open challenge. As a starting point, one could consider developing switched network models for multilayered networks where the dynamics of each layer is a function of all the other layers, and the topology transitions across layers are correlated.

- **Graph learning** techniques such as graph neural networks, or graph convolution networks offer significant representation power for complex dynamics. Leveraging these methods to learn using small datasets (as compared to traditional machine learning problems), model non-stationary processes, and integrate with interpretable controllers can help us better leverage all available data.

- **Resilient design** of networked systems can prevent cascading failures, such as those seen in power grids, road networks, and computer systems. Predicting these failures is challenging because first-principles models tend to break down due to factors unforeseen by the modeler. Using operational data, from both nominal and off-nominal system states to build network

models that quantify resilience can be of great value. To make any resilience measure practically useful, we need to rigorously validate them. This is not easy since failures come in several unique forms, and are rare events. Thus, the challenge would be to construct demonstrably correct measures of resilience for large-scale networks.

- **Using network models to guide practical operations** requires us to bridge the gap between aggregate models of system performance to specific operational decisions. There has been recent interest in the research community towards this direction [46, 47, 48, 49], but they are typically limited to very simple actions such as controlling traffic lights. For more complex interventions, such as those needed to limit the spread of epidemics (by a combination of vaccination, social-distancing, and restricted mobility), or preventing the spread of flight delays (through a combination of flight cancellations, airspace metering and flight swaps) is an open problem.

- **Objectives beyond efficiency and robustness**, such as equity and fairness have not been considered when planning control interventions. Such objective functions are typically non-linear and require the use of other tools, such as reinforcement learning. For example, adding a penalty on the maximum delays experienced by an airport in the entire network is a non-linear optimization problem that is challenging to solve at scale. Recent progress, as described in [214, 223, 224] can offer an interesting starting point for further research.

Bibliography

[1] Bureau of Transportation Statistics, "Airline On-Time Statistics and Delay Causes," 2017.

[2] M. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, A. Odoni, E. Peterson, L. Sherry, A. Trani, and B. Zou, "Total delay impact study," 2010.

[3] K. Balakrishnan, J. Polastre, J. Mooberry, R. Golding, and P. Sachs, "Blueprint for the Sky: The roadmap for the safe integration of autonomous aircraft," tech. rep., Airbus UTM, 2018.

[4] International Air Transport Association (IATA), "Economic Performance of the Airline Industry: 2019 End-Year Report," December 2019.

[5] Federal Aviation Administration, "Passenger boarding (enplanement) and all-cargo data for u.s. airports," 5 2016.

[6] B. A. Hamilton, "Urban air mobility (uam) market study," 2018.

[7] Joint Economic Committee, US Senate, "Your Flight has Been Delayed Again: Flight Delays Cost Passengers, Airlines, and the US Economy Billions," 2008.

[8] S. Bratu and C. Barnhart, "An Analysis of Passenger Delays Using Flight Operations and Passenger Booking Data," *Air Traffic Control Quarterly*, vol. 13, no. 1, pp. 1–27, 2005.

[9] Amadeus and T2RL, "Shaping the future ofAirline Disruption Management (IROPS)," 2016.

[10] News12, "JetBlue flight from Florida to Westchester delayed over 15 hours," 2021.

[11] N. Rupp, D. Owens, L. Plumly, *et al.*, "Does competition influence airline on-time performance," *Advances in airline economics*, vol. 1, pp. 251–272, 2006.

[12] M. J. Mazzeo, "Competition and service quality in the us airline industry," *Review of industrial Organization*, vol. 22, no. 4, pp. 275–296, 2003.

[13] L. Dray, A. Evans, T. Reynolds, and A. Schäfer, "Mitigation of aviation emissions of carbon dioxide: analysis for europe," *Transportation research record*, vol. 2177, no. 1, pp. 17–26, 2010.

[14] D. P. Thipphavong, R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. H. Goodrich, J. Homola, *et al.*, "Urban air mobility airspace integration concepts and considerations," in *2018 Aviation Technology, Integration, and Operations Conference*, p. 3676, 2018.

[15] H. Balakrishnan, "Control and optimization algorithms for air transportation systems," *Annual Reviews in Control*, vol. 41, pp. 39–46, 2016.

[16] New York Times, "Airport Delays Are Expected to Persist for Several Days," 2010.

[17] D. Long and S. Hasan, "Improved predictions of flight delays using lminet2 system-wide simulation model," in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, p. 6961, 2009.

[18] L. Marla, B. Vaaben, and C. Barnhart, "Integrated disruption management and flight planning to trade off delays and fuel burn," *Transportation Science*, vol. 51, no. 1, pp. 88–111, 2017.

[19] Simple Flying, "Alaska Airlines Canceled 400 Flights During Last Week's Snow Storm," 2021.

[20] J. Lee, L. Marla, and A. Jacquillat, "Dynamic disruption management in airline networks under airport operating uncertainty," *Transportation Science*, vol. 54, no. 4, pp. 973–997, 2020.

[21] A. Jacquillat and A. R. Odoni, "A roadmap toward airport demand and capacity management," *Transportation Research Part A: Policy and Practice*, vol. 114, pp. 168–185, 2018.

[22] L. Bianco and M. Bielli, "Air traffic management: Optimization models and algorithms," *Journal of Advanced Transportation*, vol. 26, no. 2, pp. 131–167, 1992.

[23] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín-Campo, "Collision avoidance in air traffic management: A mixed-integer linear optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 47–57, 2010.

[24] J.-B. Gotteland and N. Durand, "Genetic algorithms applied to airport ground traffic optimization," in *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, vol. 1, pp. 544–551, IEEE, 2003.

[25] C. Barnhart, A. M. Cohn, E. L. Johnson, D. Klabjan, G. L. Nemhauser, and P. H. Vance, "Airline crew scheduling," in *Handbook of transportation science*, pp. 517–560, Springer, 2003.

[26] M. Hansen, "Micro-level analysis of airport delay externalities using deterministic queuing models: a case study," *Journal of Air Transport Management*, vol. 8, no. 2, pp. 73–87, 2002.

[27] A. Jacquillat and A. R. Odoni, "Endogenous control of service rates in stochastic and dynamic queuing models of airport congestion," *Transportation Research Part E: Logistics and Transportation Review*, vol. 73, pp. 133–151, 2015.

[28] A. Agustı, A. Alonso-Ayuso, L. F. Escudero, C. Pizarro, *et al.*, "On air traffic flow management with rerouting. part ii: Stochastic case," *European Journal of Operational Research*, vol. 219, no. 1, pp. 167–177, 2012.

[29] A. Nilim and L. El Ghaoui, "Algorithms for air traffic flow management under stochastic environments," in *Proceedings of the 2004 American Control Conference*, vol. 4, pp. 3429–3434, IEEE, 2004.

[30] P. Fleurquin, J. Ramasco, and V. Eguiluz, "Data-driven modeling of systemic delay propagation under severe meteorological conditions," in *11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, (Lisbon, Portugal), June 2015.

[31] P. Fleurquin, J. Ramasco, and V. Eguiluz, "Systemic delay propagation in the US airport network," *Scientific Reports*, p. 1159, 2013.

[32] N. Pyrgiotis and A. Odoni, "On the impact of scheduling limits: A case study at newark liberty international airport," *Transportation Science*, vol. 50, no. 1, pp. 150–165, 2016.

[33] N. Kenan, A. Jebali, and A. Diabat, "An integrated flight scheduling and fleet assignment problem under uncertainty," *Computers & Operations Research*, vol. 100, pp. 333–342, 2018.

[34] H. Balakrishnan and B. G. Chandran, "Optimal large-scale air traffic flow management," 2014. www.mit.edu/ hamsa/pubs/BalakrishnanChandran_ATFM.pdf.

[35] B. Yu, Z. Guo, S. Asian, H. Wang, and G. Chen, "Flight delay prediction for commercial air transport: A deep learning approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 125, pp. 203 – 221, 2019.

[36] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural network design*. Martin Hagan, 2014.

[37] Y. J. Kim, S. Choi, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pp. 1–6, IEEE, 2016.

[38] G. Spiers, P. Wei, and D. Sun, "Algebraic connectivity maximization of the air transportation network," in *American Control Conference*, 2012.

[39] P. Wei, G. Spiers, and D. Sun, "Algebraic connectivity maximization for air transportation networks," *IEEE Transactions on Intelligent Transportation Systems*, no. 2, pp. 685–698, 2014.

[40] O. Lordan, J. M. Sallan, P. Simo, and D. Gonzalez-Prieto, "Robustness of the air transport network," *Transportation Research Part E*, pp. 155–163, 2014.

[41] C. Nicolaides, L. Cueto-Felgueroso, M. C. Gonzalez, and R. Juanes, "A Metric of Influential Spreading during Contagion Dynamics through the Air Transportation Network," *PLoS One*, vol. 7, no. 7, 2012.

[42] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community Structure in Time-Dependent, Multiscale, and Multiplex Networks," *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[43] R. Guimera, S. Mossa, A. Turtschi, and L. N. Amaral, "The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 22, pp. 7794–7799, 2005.

[44] O. Lordan, J. M. Sallan, P. Simo, and D. Gonzalez-Prieto, "Robustness of the air transport network," *Transportation Research Part E: Logistics and Transportation Review*, vol. 68, pp. 155 – 163, 2014.

[45] Y. Wan, C. Taylor, S. Roy, C. Wanke, and Y. Zhou, "Dynamic queuing network model for flow contingency management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1380–1392, 2013.

[46] M. Yildirimoglu, I. I. Sirmatel, and N. Geroliminis, "Hierarchical control of heterogeneous large-scale urban road networks via path assignment and regional route guidance," *Transportation Research Part B: Methodological*, vol. 118, pp. 106–123, 2018.

[47] G. S. van de Weg, H. L. Vu, A. Hegyi, and S. P. Hoogendoorn, "A hierarchical control framework for coordination of intersection signal timings in all traffic regimes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1815–1827, 2018.

[48] H. Fu, N. Liu, and G. Hu, "Hierarchical perimeter control with guaranteed stability for dynamically coupled heterogeneous urban traffic," *Transportation Research Part C: Emerging Technologies*, vol. 83, pp. 18–38, 2017.

[49] Z. Li, R. Al Hassan, M. Shahidehpour, S. Bahramirad, and A. Khodaei, "A hierarchical framework for intelligent traffic management in smart cities," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 691–701, 2017.

[50] P. Bolzern, P. Colaneri, and G. D. Nicolao, "Stochastic stability of positive markov jump linear systems," *Automatica*, vol. 50, no. 4, pp. 1181–1187, 2014.

[51] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, no. 6, 2004.

[52] M. Salathe, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, "A high resolution human contact network for infectious disease transmission," *Proceedings of the National Academy of Sciences*, vol. 107, no. 51, pp. 22020–22025, 2010.

[53] K. Dietz, "Epidemics and rumours: A survey," *Journal of the Royal Statistical Society. Series A (General)*, vol. 130, no. 4, pp. 505–528, 1967.

[54] R. Albert, I. Albert, and G. Nakarado, "Structural vulnerability of the North American power grid," *Physical Review E*, vol. 69, p. 025103, 2004.

[55] P. Crucitti, V. Latora, and M. Marchiori, "A topological analysis of the Italian electric power grid," *Physica A: Statistical Mechanics and its Applications*, vol. 338, pp. 92–97, 2004.

[56] M. Crovella and B. Krishnamurthy, *Internet Measurement: Infrastructure, Traffic and Applications*. 2006.

[57] V. Kalapala, V. Sanwalani, A. Clauset, and C. Moore, "Scale invariance in road networks," *Physical Review E*, vol. 73, p. 026130, 2006.

[58] C. von Ferber, Y. H. T. Holovatch, and V. Palchykov, "Public transport networks: empirical analysis and modeling," *The European Physical Journal B*, vol. 68, pp. 261–275, 2009.

[59] J. Manitz, *Statistical Inference for Propagation Processes on Complex Networks*. PhD thesis, Georg-August-Universität Göttingen, 2014.

[60] H. Tu, Y. Xia, H. H.-C. Iu, and X. Chen, "Optimal robustness in power grids from a network science perspective," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 1, pp. 126–130, 2018.

[61] M. Parandehgheibi and E. Modiano, "Robustness of interdependent networks: The case of communication networks and the power grid," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 2164–2169, IEEE, 2013.

[62] J.-W. Wang and L.-L. Rong, "Robustness of the western united states power grid under edge attack strategies due to cascading failures," *Safety science*, vol. 49, no. 6, pp. 807–812, 2011.

[63] P. Gai and S. Kapadia, "Contagion in financial networks," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 466, no. 2120, pp. 2401–2423, 2010.

[64] H. Amini, R. Cont, and A. Minca, "Resilience to contagion in financial networks," *Mathematical finance*, vol. 26, no. 2, pp. 329–365, 2016.

[65] J. E. Maruyama Rentschler, M. A. B. Obolensky, and M. G. M. Kornejew, "Candle in the wind? energy system resilience to natural shocks," *Energy System Resilience to Natural Shocks (June 17, 2019). World Bank Policy Research Working Paper*, no. 8897, 2019.

[66] S. Markose, S. Giansante, and A. R. Shaghaghi, "'too interconnected to fail'financial network of us cds market: Topological fragility and systemic risk," *Journal of Economic Behavior & Organization*, vol. 83, no. 3, pp. 627–646, 2012.

[67] M. Alhazmi, P. Dehghanian, S. Wang, and B. Shinde, "Power grid optimal topology control considering correlations of system uncertainties," *IEEE Transactions on Industry Applications*, vol. 55, no. 6, pp. 5594–5604, 2019.

[68] P. A. Papp and R. Wattenhofer, "Network-aware strategies in financial systems," *arXiv preprint arXiv:2002.07566*, 2020.

[69] Bureau of Transportation Statistics, "Traffic Data for U.S Airlines and Foreign Airlines U.S. Flights," 2017.

[70] A. Cook, H. A. Blom, F. Lillo, R. N. Mantegna, S. Miccichè, D. Rivas, R. Vázquez, and M. Zanin, "Applying complexity science to air traffic management," *Journal of Air Transport Management*, vol. 42, pp. 149–158, 2015.

[71] J. Fry, I. Humphreys, and G. Francis, "Benchmarking in civil aviation: some empirical evidence," *Benchmarking: An International Journal*, vol. 12, no. 2, pp. 125–137, 2005.

[72] G. Francis, I. Humphreys, and J. Fry, "The nature and prevalence of the use of performance measurement techniques by airlines," *Journal of Air Transport Management*, vol. 11, no. 4, pp. 207–217, 2005.

[73] P. D. Hooper and A. Greenall, "Exploring the potential for environmental performance benchmarking in the airline sector," *Benchmarking: An International Journal*, vol. 12, no. 2, pp. 151–165, 2005.

[74] H. Min and H. Min, "Benchmarking the service quality of airlines in the united states: an exploratory analysis," *Benchmarking: an International journal*, vol. 22, no. 5, pp. 734–751, 2015.

[75] J. Sarkis and S. Talluri, "Performance based clustering for benchmarking of us airports," *Transportation Research Part A: Policy and Practice*, vol. 38, no. 5, pp. 329–346, 2004.

[76] A. Odoni, T. Morisset, W. Drotleff, and A. Zock, "Benchmarking airport airside performance: Fra vs. ewr," in *9th USA/Europe Air Traffic Management R&D Seminar*, 2011.

[77] X. Gao and A. Malkawi, "A new methodology for building energy performance benchmarking: An approach based on intelligent clustering algorithm," *Energy and Buildings*, vol. 84, pp. 607–616, 2014.

[78] M. J. Sharma and S. J. Yu, "Performance based stratification and clustering for benchmarking of container terminals," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5016–5022, 2009.

[79] X. Dai and T. Kuosmanen, "Best-practice benchmarking using clustering methods: Application to energy regulation," *Omega*, vol. 42, no. 1, pp. 179–188, 2014.

[80] E. G. Gomes, J. C. C. B. S. Mello, A. C. R. d. Freitas, *et al.*, "Efficiency measures for a non-homogeneous group of family farmers," *Pesquisa Operacional*, vol. 32, no. 3, pp. 561–574, 2012.

[81] B. Hoffman, J. Krozel, S. Penny, A. Roy, and K. Roth, "A cluster analysis to classify days in the national airspace system," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 5711, 2003.

[82] G. Chatterji and B. Musaffar, "Characterization of days based on analysis of national airspace system performance metrics," in *AIAA guidance, navigation and control conference and exhibit*, p. 6449, 2006.

[83] K. Kuhn, A. Shah, and C. Skeels, "Characterizing and classifying historical days based on weather and air traffic," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pp. 1C3–1–1C3–12, Sep. 2015.

[84] K. D. Kuhn, "A methodology for identifying similar days in air traffic flow management initiative planning," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 1–15, 2016.

[85] K. Ren, A. M. Kim, and K. Kuhn, "Exploration of the evolution of airport ground delay programs," *Transportation Research Record: Journal of the Transportation Research Board*, pp. 1–11, 2018.

[86] A. Estes and D. Lovell, "Identifying Representative Traffic Management Initiatives," in *International Conference on Research in Air Transportation (ICRAT)*, 2016.

[87] Y. Liu and M. Hansen, "Evaluation of the performance of ground delay programs," *Transportation Research Record*, vol. 2400, no. 1, pp. 54–64, 2014.

[88] S. Gorripaty, Y. Liu, M. Hansen, and A. Pozdnukhov, "Identifying similar days for air traffic management," *Journal of Air Transport Management*, vol. 65, pp. 144–155, 2017.

[89] S. R. Grabbe, B. Sridhar, and A. Mukherjee, "Similar days in the nas: an airport perspective," in *2013 Aviation Technology, Integration, and Operations Conference*, p. 4222, 2013.

[90] A. Mukherjee, S. Grabbe, and B. Sridhar, "Classification of Days using Weather Impacted Traffic in the National Airspace System," in *AIAA Aviation Technology, Integration and Operations Conference*, 2013.

[91] S. Grabbe, B. Sridhar, and A. Mukherjee, "Clustering days with similar airport weather conditions," in *14th AIAA Aviation Technology, Integration, and Operations Conference*, p. 2712, 2014.

[92] M. Zanin and F. Lillo, "Modelling the air transport with complex networks: A short review," *The European Physical Journal Special Topics*, vol. 215, no. 1, pp. 5–21, 2013.

[93] K. Gopalakrishnan, H. Balakrishnan, and R. Jordan, "Clusters and communities in air traffic delay networks," in *American Control Conference (ACC), 2016*, pp. 3782–3788, IEEE, 2016.

[94] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, ch. 2, pp. 68–125. John Wiley & Sons, 2009.

[95] L. A. Zager and G. C. Verghese, "Graph similarity scoring and matching," *Applied Mathematics Letters*, pp. 86–94, 2008.

[96] B. Luo, R. Wilson, and E. Hancock, "Spectral clustering of graphs," in *10th International Conference on Computer Analysis of Images and Patterns, CAIP*, 2003.

[97] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, pp. 113–120, 2008.

[98] S. P. Borgatti, "Centrality and network flow," *Social Networks*, pp. 55–71, 2005.

[99] S. E. Schaeffer, "Graph clustering," *Computer Science Review I*, pp. 27–64, 2007.

[100] P.-A. Champin and C. Solnon, "Measuring the similarity of labeled graphs," in *Proceedings of the 5th international conference on Case-based reasoning (ICCBR'03): Research and Development*, pp. 80–95, 2003.

[101] J. J. Rebollo and H. Balakrishnan, "Characterization and prediction of air traffic delays," *Transportation Research Part C*, pp. 231–241, 2014.

[102] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Springer Publishing Company, Incorporated, 1st ed., 2009.

[103] Federal Aviation Administration (FAA), "ASPM database," 2017. http://aspm.faa.gov/.

[104] C. P. Barros and N. Peypoch, "An evaluation of european airlines' operational performance," *International Journal of Production Economics*, vol. 122, no. 2, pp. 525–533, 2009.

[105] B. David Mc A, "Service quality and customer satisfaction in the airline industry: A comparison between legacy airlines and low-cost airlines," *American Journal of Tourism Research*, vol. 2, no. 1, pp. 67–77, 2013.

[106] D. Gillen and T. Hazledine, "The economics and geography of regional airline services in six countries," *Journal of Transport Geography*, vol. 46, pp. 129–136, 2015.

[107] S. J. Forbes and M. Lederman, "Control rights, network structure and vertical integration: Evidence from regional airlines," *Network Structure and Vertical Integration: Evidence from Regional Airlines (November 2005)*, 2005.

[108] S. J. Forbes and M. Lederman, "Does vertical integration affect firm performance? evidence from the airline industry," *The RAND Journal of Economics*, vol. 41, no. 4, pp. 765–790, 2010.

[109] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, pp. 604–632, 1999.

[110] T. Deguchi, K. Takahashi, H. Takayasu, and M. Takayasu, "Hubs and Authorities in the World Trade Network Using a Weighted HITS Algorithm," *PLoS One*, vol. 9, no. 7, 2014.

[111] M. Benzi, E. Estrada, and C. Klymko, "Ranking hubs and authorities using matrix functions," *Linear Algebra and its Applications*, no. 5, pp. 2447–2474, 2013.

[112] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, pp. 7821–7826, 2002.

[113] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2 ed., 2009.

[114] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231, AAAI Press, 1996.

[115] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[116] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, "Cluster analysis 5th ed," 2011.

[117] Federal Aviation Administration, "FAA Advisory Database," 2017.

[118] Federal Aviation Administration, "Airport capacity benchmark report," 2004.

[119] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, pp. 53–65, 1987.

[120] Y. Bai, "Analysis of aircraft arrival delay and airport on-time performance," Master's thesis, University of Central Florida, 2006.

[121] Z. J. Hanley, "Delay Characterization and Prediction in Major U.S. Airline Networks," Master's thesis, Massachusetts Institute of Technology, 2015.

[122] Federal Aviation Administration, "FAA Announces Slot Changes at Newark Liberty International," 2016.

[123] M. Zanin and F. Lillo, "Modelling the air transport with complex networks: A short review," *The European Physical Journal Special Topics*, vol. 215, pp. 5–21, 2013.

[124] N. Pyrgiotis, K. M. Malone, and A. Odoni, "Modelling delay propagation within an airport network," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 60–75, 2013.

[125] J. M. Rosenberger, A. J. Schaefer, D. Goldsman, E. L. Johnson, A. J. Kleywegt, and G. L. Nemhauser, "A stochastic model of airline operations," *Transportation science*, vol. 36, no. 4, pp. 357–377, 2002.

[126] M. Janić, "Modeling the large scale disruptions of an airline network," *Journal of transportation engineering*, vol. 131, no. 4, pp. 249–260, 2005.

[127] V. Bulusu, V. Polishchuk, R. Sengupta, and L. Sedov, "Capacity estimation for low altitude airspace," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, p. 4266, 2017.

[128] A. Jacquillat and A. R. Odoni, "A roadmap toward airport demand and capacity management," *Transportation Research Part A: Policy and Practice*, 2017.

[129] R. Kicinger, J.-T. Chen, M. Steiner, and J. Pinto, "Airport capacity prediction with explicit consideration of weather forecast uncertainty," *Journal of Air Transportation*, vol. 24, no. 12, pp. 18–28, 2016.

[130] V. Ramanujam and H. Balakrishnan, "Estimation of arrival-departure capacity tradeoffs in multi-airport systems," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 2534–2540, IEEE, 2009.

[131] N. Xu, K. B. Laskey, G. Donohue, and C. H. Chen, "Estimation of Delay Propagation in the National Aviation System Using Bayesian Networks," in *6th USA/Europe Air Traffic Management Research and Development Seminar*, June 2005.

[132] S. AhmadBeygi, A. Cohn, Y. Guan, and P. Belobaba, "Analysis of the Potential for Delay Propagation in Passenger Airline Networks," *Journal of Air Transport Management*, vol. 14 No. 5, pp. 221–236, 2008.

[133] M. Jetzki, "The propagation of air transport delays in Europe," Master's thesis, Department of Airport and Air Transportation Research, Aachen University, 2009.

[134] L. Hao, M. Hansen, Y. Zhang, and J. Post, "New York, New York: Two ways of estimating the delay impact of New York airports," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, pp. 245 – 260, 2014.

[135] A. Cook, G. Tanner, S. Cristóbal, and M. Zanin, "Delay propagation Ð new metrics, new insights," in *11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*, (Lisbon, Portugal), June 2015.

[136] A. Klein, S. Kavoussi, D. Hickman, D. Simenauer, M. Phaneuf, and T. MacPhail, "Predicting Weather Impact on Air Traffic," in *Integrated Communication, Navigation and Surveillance (ICNS) Conference*, May 2007.

[137] B. Sridhar and N. Chen, "Short term national airspace system delay prediction," *Journal of Guidance, Control, and Dynamics*, vol. 32, No. 2, 2009.

[138] A. Klein, C. Craun, and R. S. Lee, "Airport delay prediction using weather-impacted traffic index (WITI) model," in *Digital Avionics Systems Conference (DASC)*, 2010.

[139] R. Yao, W. Jiandong, and X. Tao, "A flight delay prediction model with consideration of cross-flight plan awaiting resources," in *International Conference on Advanced Computer Control (ICACC)*, 2010.

[140] K. Gopalakrishnan, H. Balakrishnan, and R. Jordan, "Clusters and Communities in Air Traffic Delay Networks," in *American Control Conference*, July 2016.

[141] Federal Aviation Administration (FAA), "OPSNET website," 2017. http://aspm.faa.gov/.

[142] C. Nowzari, V. Preciado, and G. Pappas, "Analysis and Control of Epidemics: A Survey of Spreading Processes on Complex Networks," *IEEE Control Systems Magazine*, February 2016.

[143] M. Zanin and F. Lillo, "Modelling the air transport with complex networks: A short review," *The European Physical Journal Special Topics*, vol. 215, pp. 5–21, 2013.

[144] C. Ho, M. J. Kochenderfer, V. Mehta, and R. S. Caceres, "Control of epidemics on graphs," in *Proceedings of the Conference on Decision and Control*, 2015.

[145] M. Ogura and V. M. Preciado, "Disease spread over randomly switched large-scale networks," in *Proceedings of the American Control Conference*, 2015.

[146] D. Acemoglu, V. M. Carvalho, A. Ozdaglar, and A. Tahbaz-Salehi, "The network origins of aggregate fluctuations," *Econometrica*, vol. 80, no. 5, pp. 1977–2016, 2012.

[147] I. Herman, D. Martinec, Z. Hurak, and M. Sebek, "Nonzero Bound on Fiedler Eigenvalue Causes Exponential Growth of H-Infinity Norm of Vehicular Platoon," *IEEE Transactions on Automatic Control*, vol. 60, no. 8, pp. 2248–2253, 2015.

[148] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[149] T. Sarkar, M. Roozbehani, and M. Dahleh, "Robustness scaling in large networks," in *American Control Conference*, 2016.

[150] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pp. 1–6, IEEE, 2016.

[151] N. Xu, G. Donohue, K. B. Laskey, and C.-H. Chen, "Estimation of delay propagation in the national aviation system using bayesian networks," in *6th USA/Europe Air Traffic Management Research and Development Seminar*, FAA and Eurocontrol Baltimore, MD, 2005.

[152] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," *arXiv preprint arXiv:1309.6835*, 2013.

[153] K. Gopalakrishnan, H. Balakrishnan, and R. Jordan, "Stability of networked systems with switching topologies," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pp. 2601–2608, IEEE, 2016.

[154] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2012.

[155] A. N. Vargas, E. F. Costa, and J. B. Val, "On the control of markov jump linear systems with no mode observation: application to a dc motor device," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 10, pp. 1136–1150, 2013.

[156] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Analysis and control of epidemics: A survey of spreading processes on complex networks," *IEEE Control Systems*, vol. 36, no. 1, pp. 26–46, 2016.

[157] L. Xiao, A. Hassibi, and J. P. How, "Control with random communication delays via a discrete-time jump system approach," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 3, pp. 2199–2204, IEEE, 2000.

[158] Y. Zhang and Y.-P. Tian, "Consentability and protocol design of multi-agent systems with stochastic switching topology," *Automatica*, vol. 45, no. 5, pp. 1195–1201, 2009.

[159] O. Costa and M. Fragoso, "Stability Results for Discrete-Time Linear Systems with Markovian Jumping Parameters," *Journal of Mathematical Analysis and Applications*, vol. 179, pp. 154–178, 1993.

[160] O. Costa, M. Fragoso, and R. Marques, *Discrete-Time Markov Jump Linear Systems*. USA: Springer-Verlag, 1st ed., 2005.

[161] Z. Du, N. Ozay, and L. Balzano, "Mode clustering for markov jump systems," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 126–130, IEEE, 2019.

[162] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control systems*, vol. 19, no. 5, pp. 59–70, 1999.

[163] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009.

[164] X. Feng, K. A. Loparo, Y. Ji, and H. J. Chizeck, "Stochastic stability properties of jump linear systems," *IEEE transactions on Automatic Control*, vol. 37, no. 1, pp. 38–53, 1992.

[165] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*. Springer Science & Business Media, 2006.

[166] E.-K. Boukas, *Stochastic switching systems: analysis and design*. Springer Science & Business Media, 2007.

[167] Y. Guo, "Stabilization of positive markov jump systems," *Journal of the Franklin Institute*, vol. 353, no. 14, pp. 3428–3440, 2016.

[168] F. Kozin, "A survey of stability of stochastic systems," *Automatica*, vol. 5, no. 1, pp. 95–112, 1969.

[169] P. Bolzern, P. Colaneri, and G. De Nicolao, "On almost sure stability of continuous-time markov jump linear systems," *Automatica*, vol. 42, no. 6, pp. 983–988, 2006.

[170] J. Cavalcanti and H. Balakrishnan, "Sign-stability of positive markov jump linear systems," *Automatica*, vol. 111, p. 108638, 2020.

[171] J. Cavalcanti and H. Balakrishnan, "Finite-time behavior of switching networks," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 6505–6512, IEEE, 2017.

[172] M. Ball, C. Barnhart, G. Nemhauser, and A. Odoni, "Air transportation: Irregular operations and control," *Handbooks in operations research and management science*, vol. 14, pp. 1–67, 2007.

[173] E. D. Vugrin, M. A. Turnquist, and N. J. Brown, "Optimal recovery sequencing for enhanced resilience and service restoration in transportation networks," *International Journal of Critical Infrastructures*, vol. 10, no. 3-4, pp. 218–246, 2014.

[174] F. Hu, C. H. Yeung, S. Yang, W. Wang, and A. Zeng, "Recovery of infrastructure networks after localised attacks," *Scientific reports*, vol. 6, p. 24522, 2016.

[175] L. Chen and E. Miller-Hooks, "Resilience: an indicator of recovery capability in intermodal freight transport," *Transportation Science*, vol. 46, no. 1, pp. 109–123, 2012.

[176] X. Zhang, E. Miller-Hooks, and K. Denny, "Assessing the role of network topology in transportation network resilience," *Journal of Transport Geography*, vol. 46, pp. 35–45, 2015.

[177] T.-D. Nguyen, X. Cai, Y. Ouyang, and M. Housh, "Modelling infrastructure interdependencies, resiliency and sustainability," *International Journal of Critical Infrastructures*, vol. 12, no. 1-2, pp. 4–36, 2016.

[178] A. A. Ganin, E. Massaro, A. Gutfraind, N. Steen, J. M. Keisler, A. Kott, R. Mangoubi, and I. Linkov, "Operational resilience: concepts, design and analysis," *Scientific reports*, vol. 6, 2016.

[179] N. J. Watkins, C. Nowzari, and G. J. Pappas, "Inference, prediction, and control of networked epidemics," *arXiv preprint arXiv:1703.07409*, 2017.

[180] I. Matei, N. C. Martins, and J. S. Baras, "Optimal linear quadratic regulator for markovian jump linear systems, in the presence of one time-step delayed mode observations," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 8056–8061, 2008.

[181] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, no. 10, pp. 1709–1721, 2005.

[182] O. L. V. Costa, E. Assumpção Filho, E. K. Boukas, and R. Marques, "Constrained quadratic state feedback control of discrete-time markovian jump linear systems," *Automatica*, vol. 35, no. 4, pp. 617–626, 1999.

[183] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump markov linear systems using particles," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 104–117, Springer, 2007.

[184] B.-G. Park, J.-W. Lee, and W. H. Kwon, "Robust one-step receding horizon control for constrained systems," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 7, pp. 381–395, 1999.

[185] B.-G. Park and W. H. Kwon, "Robust one-step receding horizon control of discrete-time markovian jump uncertain systems," *Automatica*, vol. 38, no. 7, pp. 1229–1235, 2002.

[186] W. Yewen, J. Zhu, and W. Xie, "Optimal control strategy for discrete-time mjls with controllable markov chain and gaussian white noise," in *Control Conference (CCC), 2015 34th Chinese*, pp. 2188–2193, IEEE, 2015.

[187] Y. Tu, M. O. Ball, and W. S. Jank, "Estimating flight departure delay distributions- a statistical approach with long-term trend and short-term pattern," *Journal of the American Statistical Association*, vol. 103, no. 481, pp. 112–125, 2008.

[188] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[189] E. Mueller and G. Chatterji, "Analysis of aircraft arrival and departure delay characteristics," in *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, p. 5866, 2002.

[190] J. J. Rebollo and H. Balakrishnan, "A network-based model for predicting air traffic delays," in *5th International Conference on Research in Air Transportation*, 2012.

[191] N. K. Bose and P. Liang, "Neural network fundamentals with graphs, algorithms and applications," *McGraw-Hill*, 1996.

[192] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, 1991.

[193] D. F. Specht, "Probabilistic neural networks," *Neural networks*, vol. 3, no. 1, pp. 109–118, 1990.

[194] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[195] A. S. Hadi, "Identifying multiple outliers in multivariate data," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 54, no. 3, pp. 761–771, 1992.

[196] P. Filzmoser, *A multivariate outlier detection method*. na, 2004.

[197] D. M. Rocke and D. L. Woodruff, "Identification of outliers in multivariate data," *Journal of the American Statistical Association*, vol. 91, no. 435, pp. 1047–1061, 1996.

[198] D. Yu, G. Sheikholeslami, and A. Zhang, "Findout: finding outliers in very large datasets," *Knowledge and Information Systems*, vol. 4, no. 4, pp. 387–412, 2002.

[199] W. Eberle and L. Holder, "Discovering structural anomalies in graph-based data," in *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pp. 393–398, IEEE, 2007.

[200] S. Shekhar, C.-T. Lu, and P. Zhang, "Detecting graph-based spatial outliers," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 451–468, 2002.

[201] H. B. Ahmed, D. Dare, and A.-O. Boudraa, "Graph signals classification using total variation and graph energy informations," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 667–671, IEEE, 2017.

[202] R. S. Tsay, D. Pena, and A. E. Pankratz, "Outliers in multivariate time series," *Biometrika*, vol. 87, no. 4, pp. 789–804, 2000.

[203] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[204] L. Isserlis, "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables," *Biometrika*, vol. 12, no. 1/2, pp. 134–139, 1918.

[205] R. Kan, "From moments of sum to moments of product," *Journal of Multivariate Analysis*, vol. 99, no. 3, pp. 542–554, 2008.

[206] T. Schürmann and I. Hoffmann, "On biased correlation estimation," *arXiv preprint arXiv:1707.09037*, 2017.

[207] R. A. Fisher, "Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population," *Biometrika*, vol. 10, no. 4, pp. 507–521, 1915.

[208] I. Olkin, J. W. Pratt, *et al.*, "Unbiased estimation of certain correlation coefficients," *The Annals of Mathematical Statistics*, vol. 29, no. 1, pp. 201–211, 1958.

[209] G. Brys, M. Hubert, and A. Struyf, "A robust measure of skewness," *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 996–1017, 2004.

[210] M. Hubert and E. Vandervieren, "An adjusted boxplot for skewed distributions," *Computational Statistics & Data Analysis*, vol. 52, no. 12, pp. 5186 – 5201, 2008.

[211] M. Z. Li, K. Gopalakrishnan, K. Pantoja, and H. Balakrishnan, "Graph signal processing techniques for analyzing aviation disruptions," *Transportation Science*, 2021.

[212] M. Z. Li, K. Gopalakrishnan, Y. Wang, and H. Balakrishnan, "Outlier analysis of airport delay distributions in us and china," in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–12, IEEE, 2020.

[213] M. Z. Li, K. Gopalakrishnan, H. Balakrishnan, S. Shin, D. Jalan, A. Nandi, and L. Marla, "Dynamics of disruption and recovery in air transportation networks," in *International Conference on Research in Air Transportation (ICRAT)*, 2020.

[214] M. Z. Li, K. Gopalakrishnan, and H. Balakrishnan, "Approximate projection-based control of networks," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 5573–5579, IEEE, 2020.

[215] H. Balakrishnan and Y. Jung, "A framework for coordinated surface operations planning at dallas-fort worth international airport," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 6553, 2007.

[216] H. Balakrishnan and B. G. Chandran, "Algorithms for scheduling runway operations under constrained position shifting," *Operations Research*, vol. 58, no. 6, pp. 1650–1665, 2010.

[217] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations Research*, vol. 46, no. 3, pp. 406–422, 1998.

[218] H. Balakrishnan and B. G. Chandran, "A distributed framework for traffic flow management in the presence of unmanned aircraft," in *USA/Europe Air Traffic Management R&D Seminar*, June 2017.

[219] S. Cafieri and N. Durand, "Aircraft deconfliction with speed regulation: new models from mixed-integer optimization," *Journal of Global Optimization*, vol. 58, no. 4, pp. 613–629, 2014.

[220] T. Vossen and M. Ball, "Optimization and mediated bartering models for ground delay programs," *Naval Research Logistics (NRL)*, vol. 53, no. 1, pp. 75–90, 2006.

[221] M. Ball, G. Donohue, and K. Hoffman, "Auctions for the safe, efficient, and equitable allocation of airspace system resources," *Combinatorial auctions*, vol. 1, 2006.

[222] C. Barnhart, P. Belobaba, and A. R. Odoni, "Applications of operations research in the air transport industry," *Transportation science*, vol. 37, no. 4, pp. 368–391, 2003.

[223] A. Balasingam, K. Gopalakrishnan, R. Mittal, V. Arun, A. Saeed, M. Alizadeh, H. Balakrishnan, and H. Balakrishnan, "Throughput-fairness tradeoffs in mobility platforms," in *Mobisys*, 2020.

[224] C. Chin, K. Gopalakrishnan, M. Egorov, A. Evans, and H. Balakrishnan, "Efficiency and fairness in unmanned air traffic flow management," *IEEE Transactions on Intelligent Transportation Systems*, 2021.