Advances in Data-Driven Models for Transportation

by

Yee Sian Ng

Submitted to the Sloan School of Management
on May 17, 2019, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

With the rising popularity of ride-sharing and alternative modes of transportation, there has been a renewed interest in transit planning to improve service quality and stem declining ridership. However, it often takes months of manual planning for operators to redesign and reschedule services in response to changing needs. To this end, we provide four models of transportation planning that are based on data and driven by optimization. A key aspect is the ability to provide certificates of optimality, while being practical in generating high-quality solutions in a short amount of time.

We provide approaches to combinatorial problems in transit planning that scales up to city-sized networks. In transit network design, current tractable approaches only consider edges that exist, resulting in proposals that are closely tethered to the original network. We allow new transit links to be proposed and account for commuters transferring between different services. In integrated transit scheduling, we provide a way for transit providers to synchronize the timing of services in multi-modal networks while ensuring regularity in the timetables of the individual services. This is made possible by taking the characteristics of transit demand patterns into account when designing tractable formulations.

We also advance the state of the art in demand models for transportation optimization. In emergency medical services, we provide data-driven formulations that outperforms their probabilistic counterparts in ensuring coverage. This is achieved by replacing independence assumptions in probabilistic models and capturing the interactions of services in overlapping regions. In transit planning, we provide a unified framework that allows us to optimize frequencies and prices jointly in transit networks for minimizing total waiting time.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Professor of Operations Research
Co-Director, Operations Research Center
Acknowledgments

I wish to thank my thesis committee, Dimitris Bertsimas, Patrick Jaillet and Carolina Osorio for their time, suggestions and valuable comments. This thesis also benefitted from the support of our collaborators from Mitsubishi and MELCO, Yoneta, Ryuya, Arvind and Daniel.

This thesis was written over five years of guidance under Dimitris, who has an unerring taste for practical insights from our research together. He has an inspiring amount of enthusiasm for and optimism in the value of research, and taught me that our work is not complete until it can be understood and used by others.

My collaboration with Julia Yan has been crucial to the success of this thesis. Her use of simple examples has saved us from the blind spots of my algorithmic tendencies on multiple occasions, and her creativeness in solution approaches is inspiring.

I am deeply grateful to Melvyn and Patrick who supported and believed in me when I began research in the Future Urban Mobility Lab under the Singapore-MIT Alliance. I spent a summer under the tutelage of Melvyn, who sat me down to work through basic proofs on the board in his office together – I can only hope to pay his generosity and patience forward.

My time at the Operations Research Center (ORC) is a memorable one. I wish to thank Miles, Iain and Joey for inspiring me to contribute and be part of open-source communities. Alex, Angie, Chongyang, Daniel, Mila, Nataly, Nikita, Joel, Will and many wonderful seniors have made the ORC warm and welcoming. My batch mates are a great source of help and companionship: Max Biggs, Max Burg, Rim, Ilias, Lauren, Kevin, Konstantina, Michael, Lennart, Jing, Matthieu. Many thanks to Bart, Bartolomeo, Chris, Daniel, Hari, Martin and Sebastien for their time in discussing research ideas with me. I also wish to thank Agni, Andrew, Arthur, Chris Coey, Deeksha, Elisabeth, Emily, Emma, Galit, Holly, Jackie, Jess, Joshua, Jourdain, Julia, Kayla, Lea, Matthew, Or Dan, Peter, Qingyang, Rebecca, Ryan, Sam, Sean, Susan, Tamar, Ted, Vasileios, Yuchen and Zach for their great company.

One of the most important periods for me was doing an internship in the T-4 divi-
sion with the Center for Nonlinear Studies in Los Alamos. Line is an amazing source of support and encouragement who helped me through some of my most difficult periods. Sidhant is a wonderful mentor who pushed me to think beyond “linear decision rules” for optimization. I also had the good company of Andrey, Arun, Carleton, Deep, Harsha, Kaarthik, Kalina, Marc, Misha, Russell, Scott, Tillmann Mühlpfordt and Tillmann Weisser over a summer of food and sun and sand.

I spent an equally amazing summer in Google with Ross as my host. Apart from being great company, he provided the guidance I needed while entrusting me with autonomy. Which I spent on rock-climbing and doing dubious things with Jonathan and Michelle, in the company of fellow interns Andrew Ross, Felix, Harini, Jisoo, Kevin, Samantha, Zack and Zelda. Together with Miles, Ondrej, Christian, Jon and the rest of the O.R. team, they have made my decision to join Google an easy one.

Special thanks goes to Jack and Jenny for making Cambridge feel like home, and reminding me that how we spend our days is how we spend our lives. They have filled mine with music and food and obscene amounts of television and movies. Thanks to Daisy and Chester and Marcos for being like family – their sense of fun and humour is expansive and infectious, and their company is deep and nourishing. Thanks to Colin for being a wonderful drinking buddy and to Sarah for her brownies.

Many other people have been important to me at various points, and I wish to acknowledge them: Yossiri, Saurabh, Dax, Gladia, Grace Fong, Grace Goon, Grace Yeo, Bingshao, Sanqian, Haiwen, Charlotte, Delph, Danny, Jaren, Alvin, Mingfang, Gargoei, Gao Yuan, Rachel, Sau Ling, Li Cui, Cedric, Shawn, Beng, Kelly, Kenneth, Krishnan, Cynthia, Weiyi, Christine, Qinying, Yangshun, Isabelle, Chloe, Shujing, Ally, Louise, Jiahao, Martijn, Carolyn, Frans and Frank.

Last but not least, I wish to thank my family for their love and support. My parents Thomas and Mary have deeply imprinted on me the value of a good education and the importance of being a good person. My siblings are a great source of counsel, and I look forward to experience and celebrate life in all its stages with them.
## Contents

1 Introduction  
1.1 Overview .................................................. 24

2 Robust and Stochastic Formulations for Ambulance Deployment and Dispatch  
2.1 Introduction .................................................. 28  
2.1.1 Previous Work ............................................. 28  
2.1.2 Our Contribution ........................................... 31  
2.2 From Online Dispatch to Ambulance Deployment ......................... 32  
2.2.1 Ambulance Scheduling and Dispatch .......................... 33  
2.2.2 Gradual Coverage .......................................... 34  
2.3 Ambulance Deployment with Recourse ................................ 35  
2.3.1 Structured Uncertainty Set ................................. 37  
2.3.2 The Recourse Function ...................................... 38  
2.3.3 Column and Constraint Algorithm .......................... 41  
2.4 Computational Results ......................................... 43  
2.4.1 Experimental Setup ........................................ 44  
2.4.2 Discussion of Results ....................................... 48  
2.5 Conclusion .................................................... 51

3 Frequency-Setting and Pricing on Multi-Modal Transit Networks  
3.1 Introduction .................................................. 53  
3.1.1 Literature Review .......................................... 54
3.1.2 Our Contribution .................................................. 57
3.2 Preliminaries .......................................................... 57
3.3 Optimization with Known Choices ................................. 61
3.4 Optimization with Design-Dependent Choices .................... 64
  3.4.1 Discrete Choice Models in Transit Assignment ............... 66
  3.4.2 Solution Algorithm .............................................. 68
3.5 Computational Experiments ........................................... 70
  3.5.1 Case Study: Tokyo .............................................. 71
  3.5.2 Case Study: Boston .............................................. 78
3.6 Conclusion ............................................................ 81

4 Transit Network Design at Scale ................................. 83
  4.1 Introduction .......................................................... 83
  4.2 Literature Review .................................................... 85
  4.3 Methods ............................................................... 87
    4.3.1 Serving direct passengers .................................... 88
    4.3.2 Serving passengers with transfers .......................... 93
    4.3.3 Considering travel times ..................................... 100
    4.3.4 Speeding up the subproblem ................................. 104
  4.4 Computational Results ............................................. 106
    4.4.1 A small synthetic network .................................... 107
    4.4.2 Solving the subproblem at scale ............................ 111
    4.4.3 A large-scale case study from Boston ....................... 113
  4.5 Conclusion .......................................................... 119

5 Transit Integration with Multiple Providers ..................... 121
  5.1 Introduction .......................................................... 121
  5.2 Literature Review .................................................... 123
  5.3 An Integrated Approach to Transit Scheduling .................. 125
    5.3.1 Regularity in Timetable Scheduling ........................ 125
    5.3.2 Sharing Segments across Multiple Lines .................... 128
5.3.3 Coordinating Timetables for Transferring Commuters  
5.3.4 Mitigating Congestion in Transit Services  
5.3.5 A Unified Perspective for Data-Driven Scheduling  
5.3.6 A Scalable Approach for Transit Passenger Assignment  
5.4 Computational Experiments  
5.4.1 Model Comparisons on the MBTA Subway  
5.4.2 A Multi-Modal Large Scale Case Study  
5.5 Conclusion  

A Appendix to Chapter 2  
A.1 Node-Arc Adjacency Matrix Example  
A.2 Proofs of Propositions  
A.3 Scenario Generation in the C&CG algorithm  
A.4 Deployment Plans Generated  
A.5 Model Performance Comparisons  

B Appendix to Chapter 3  
B.1 Input Parameter Estimation  

C Appendix to Chapter 5
List of Figures

2-1 Different hierarchies of coverage for Washington D.C. Illustrates the geographical regions corresponding to (a) individual regions, (b) each region with those adjacent to it, (c) drivetime coverages from each station, and (d) the whole of Washington D.C. ........................................ 39

2-2 Average hourly number of emergency calls per region. This is based on data of emergency calls requiring ambulatory care for DCFEMS, from Jan 2012 to Mar 2013. ................................................................. 43

2-3 Sample drivetime region. The orange dots corresponds to road nodes reached within 10minutes, based on drivetimes using OpenStreetMap data, starting from the fire station (red dot). The grey areas indicate the coverage regions, corresponding to those that contains at least 1 orange dot. ................................................................. 44

2-4 Deployment Coverages for the Robust and Stochastic deployment models with 35 ambulances. The regions are shaded by counting the number of ambulances available within 10minutes. The size of the orange circles corresponds to the number of ambulances allocated to that location. ................................................................. 49

3-1 A screenshot of Google Maps illustrating route options in Japanese transit. (Accessed 2017-11-03) ................................................................. 59

3-2 A map of the Kanagawa trains network. ........................................... 72
3-3 Optimization progress for multiple different random starting points and two example budgets, on the full Kanagawa network and using frequency-setting and line pricing. Each line corresponds to the objective function value from a different random starting point. 76

3-4 Optimal objective values under frequency-setting without pricing, coordinated frequency-setting and distance pricing, coordinated frequency-setting and line pricing, and the system optimum. The network of study was the full Kanagawa network. 77

3-5 Distribution of utilities among passengers for two different frequency-setting and pricing policies on the full Kanagawa network. Note that because passengers lose time and money while commuting, all utilities are negative. 78

3-6 Proportion of Kanagawa commuters who would choose line-pricing instead of distance-pricing, by travel distance. 79

3-7 A subset of the MBTA network used in Section 3.5.2, showing parts of the Red, Green, Orange, and 1 bus (gray) lines. Stations further outside of the metropolitan core of the network, and some 1 bus stations, are omitted for clarity. To generate more route options for commuters, some stations that were within a 0.5 mile walk of each other were assumed to be substitutable for each other. Stations that were viewed as substitutable are surrounded with dotted boxes. 80

4-1 An illustration of transfers excessive travel times due to transfers. 103

4-2 Synthetic bus network generated under the direct-route model without travel time constraints. The network above was produced using the direct-route column generation procedure without edge preprocessing, with budget $B = 24$. A single line is produced connecting all of the stops. 108
4-3 Synthetic bus network generated under the direct-route model with travel time constraints. The network above was produced using the direct-route column generation procedure without edge preprocessing, with budget $B = 24$.

4-4 Synthetic bus network with single-transfer column-and-constraint generation and travel time constraints. We illustrate the network produced in each iteration of the single-transfer column-and-constraint generation procedure. The budget was set to $B = 1.5$, which is exactly the budget needed to sustain a grid network.

4-5 Optimization lower and upper bounds for an iteration of the full subproblem. The bounds are based on a $N = 40$ random network. Because the subproblem is a maximization problem, the lower bound represents the actual solution values.

4-6 Boston-area bus networks generated from the MBTA demand matrix. Blue lines represent the original network, red lines represent bus lines that were generated by the algorithm, and green lines represent the final optimized bus network.

4-7 Boston-area bus networks generated from the Blue Bikes demand matrix. Blue lines represent the original network, red lines represent bus lines that were generated by the algorithm, and green lines represent the final optimized bus network.

5-1 The MBTA subway network.

5-2 Coordination of Timetables. We built visualizations of Marey diagrams [116] to illustrate the schedules and occupancies of the different services. The lines are colored based on the services that they correspond to, and their width corresponds to the occupancy along each service run. This is an illustration of the best found schedule for the MBTA in-bound red and orange subway network.
5-3 Comparison of the Model Runtimes. We ran the Network, Connect, Segment and Arrival formulations on two different networks with a time limit of 10,000 seconds and MIPFocus=3. For each, we plot the incumbent solutions based on their objective value under the network formulation and the corresponding time they were generated. We also plot lower bounds (in orange) on the optimal objective value as time proceeds.

5-4 Computational runtimes with regularity constraints. We ran the Network, Connect, Segment and Arrival Model with regularity constraints on inbound red and orange lines with a time limit of 10,000s and MIPFocus=3. For each, we plot the incumbent solutions based on their objective value under the network formulation and the corresponding time they were generated. We also plot lower bounds (in orange) on the optimal objective value as time proceeds. Finally, we added points to show the MIP start solutions of the Network Model before it started the branch-and-bound process.

5-5 Computational runtimes for the Late Night T network. We ran the Connect Model (in blue) and Arrival Model (in purple) models on the full MBTA network including both buses and trains. For each, we plot the incumbent solutions based on their objective value under the connect formulation and the corresponding time they were generated. We also plot the best bounds (in orange) on the optimal objective value as time proceeds.

A-1 Scenarios generated by the C&CG algorithm. This is based on the robust deployment model for 35 ambulances, with parameter $\alpha = 0.01$. We illustrate the first 16 scenarios generated as chloropleths, and corresponding deployment plans as orange circles, from left to right, top to bottom.
C-1  Some suggestions for the MBTA Late Night T. Both images taken from 

C-2  Recommended Late Night Services for the MBTA Network. A map 
of the recommended MBTA services based on optimization using the 
Late Night T data. All the subway lines (red, blue, orange and green) 
are maintained. Among the bus services, the ones that are maintained 
are plotted in purple, and the ones to be left out of late night services 
are plotted in grey. Map tiles by Stamen Design, under CC By 3.0. 
Data by OpenStreetMap, under CC BY SA. . . . . . . . . . . . . . . 162
List of Tables

2.1 Summary of the Notation ........................................... 32

3.1 Summary of the Notation ........................................... 58

3.2 Travel times for various origin-destination pairs on the Kanagawa network 72

3.3 Frequency-setting on a small network under varying budgets and commuter sensitivities to congestion. ........................................... 74

3.4 Price-setting on a small network under varying budgets and commuter sensitivities to congestion ........................................... 75

3.5 Median run time over thirty iterations for various budgets. ........... 76

3.6 Optimal objective values under the system optimum, frequency-setting without pricing, coordinated frequency-setting and line pricing, and coordinated frequency-setting and distance pricing. The network of study was the Boston network. ........................................... 81

4.1 Summary of the Notation ........................................... 87

4.2 Performance of preprocessed subproblem as compared to the full subproblem. We compared both approaches in performance and computation time (the full subproblem was capped at 48 hours). Each row of the table represents 25 different simulations. .................. 112

4.3 Objectives of the original MBTA network and the optimized network 115

4.4 Characteristics of the original MBTA network and the optimized network116

4.5 Running time for Algorithm 2 on the Boston dataset .................. 118

5.1 Summary of the Notation. ........................................... 126
5.2 Overview of the four scheduling formulations. ........................................ 135

5.3 Computational Performance on the Red and Orange Lines. We ran all
the models on the inbound red and orange lines with a time limit of
10,000 seconds, and keep track of the number of the top three classes
of cuts generated. All but the Network Model terminated at optimality.
We report the MIP Gap for each of them, defined as the lower
and upper objective bound divided by the absolute value of the upper
bound. We also report the time when the Network Model found a
better schedule (if applicable). ............................................................ 141

5.4 Computational Performance on the Full Subway Network. We ran the
models on the full MBTA subway network with a time limit of 10,000
seconds, and keep track of the number of the top three classes of cuts
generated. All but the Connect Model terminated at optimality. ... 141

A.1 Robust Deployment Plans. Generated for different parameter values
\( \alpha \), with varying numbers of ambulances. For values of \( \alpha \) above 0.01,
the model saturated quickly which resulted in deployment plans that
were not as competitive. ................................................................. 153

A.2 Deployment Plans. Generated for each model with varying numbers
of ambulances. Both the Stochastic and Robust formulations evenly
distribute the allocation of ambulances, but the MEXCLP and MALP
clusters them in a few central locations. .......................................... 154

A.3 Coverage Peak Steady. Simulated coverages based on 12 replications,
each for 360 hours of continuous peak-hour ambulance operations, with
steady turnaround times. Shaded cells are results that should be ignored.155

A.4 Coverage Peak Volatile. Simulated coverages based on 12 replications,
each for 360 hours of continuous peak-hour ambulance operations, with
volatile turnaround times. Shaded cells are results that should be ignored.155
A.5 Coverage Off-Peak Steady. Simulated coverages based on 12 replications, each for 360 hours of continuous off-peak ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored. ................................................................. 156

A.6 Coverage Off-Peak Volatile. Simulated coverages based on 12 replications, each for 360 hours of continuous off-peak ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored. ................................................................. 156

A.7 Response Peak Steady. Simulated response times based on 12 replications, each for 360 hours of continuous peak-hour ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored. ................................................................. 156

A.8 Response Peak Volatile. Simulated response times based on 12 replications, each for 360 hours of continuous peak-hour ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored. ................................................................. 157

A.9 Response Off-Peak Steady. Simulated response times based on 12 replications, each for 360 hours of continuous off-peak ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored. ................................................................. 157

A.10 Response Off-Peak Volatile. Simulated response times based on 12 replications, each for 360 hours of continuous off-peak ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored. ................................................................. 157
Chapter 1

Introduction

In a report by the United Nations [164], urban populations are predicted to increase from 54 percent of the global population in 2014 to 66 percent in 2050. Together with overall population growth, this represents an increase of 2.5 billion people. With more of the world’s population living in cities, it is increasingly important to provide transit systems that can efficiently serve a densely-settled population. But public transit systems face significant operating challenges, and the question of how to efficiently manage such systems is of crucial importance. In recent years, increasing urban populations and shrinking operating budgets have both contributed to significant overcrowding and delays during peak hours [68]. Many cities such as Philadelphia [104], Los Angeles [130] and Washington D.C. [152] are seeing declining bus ridership, prompting transit authorities to consider what can be done to halt this decline.

Meanwhile, transit networks are growing and diversifying rapidly, making it increasingly important to understand how operating decisions influence ridership and demand between different route options. In this regard, we are motivated by modern developments in both private ride-sharing services such as Uber and Lyft, and public infrastructure such as electronic road pricing schemes in London, Singapore, and Stockholm. A recent bus network re-design in Houston led to a 6.8% increase in ridership across the bus and light rail networks [27], inspiring other cities to also consider re-designing their bus networks. Examples include Philadelphia [104], Boston [166], St. Louis [149], and Edmonton [157].
This thesis grew out of the desire to employ optimization in the process of transit planning. The use of optimization for operational planning is not a new development: reflecting on his time as an advisor to the US Air Force Comptroller during World War II, Dantzig [54] observed that

Initially there was no objective function; broad goals were never stated explicitly in those days because practical planners simply had no way to implement such a concept. [...] In place of an explicit goal or objective function, there were a large number of ad hoc ground rules issues by those in authority to guide the selection. Without such rules, there would have been, in most cases, an astronomical number of feasible solutions to choose from. [...] Thus the means to attain the objective function becomes an objective in itself, which in turn spawns new ground rules as to how to go about attaining the means such as how best to go about building bombers or space shuttles. These means in turn become confused with goals, etc., down the line.

Given the numerous advances in optimization technology [29, 30, 52], we might imagine the situation to be different today. I have the privilege of observing my colleagues Sebastien and Arthur work closely with officials from Boston Public Schools to replace manual planning with optimization software in designing school bus routes, resulting in millions of dollars in annual savings [19]. But anecdotally, it still appears to be the exception rather than the norm and I will state two cases as examples.

- In 2012, The Central Transportation Planning Staff (CTPS) published the Bus Walking Radius Study [44]. This was directed by the Boston Metropolitan Planning organization (MPO) to understand the implications of eliminating duplicative or poor services and redirecting the resources to focus on the remaining transit routes. In the report, they did not use any optimization techniques, and described their search for a good set of services as a manual procedure:

  [...] the selection of one route for elimination affects the overlap percentages of all other routes with overlapping coverages. Therefore,
the selection of routes for elimination was done in an iterative fashion: first one route was selected; next, the resulting impact on other routes’ overlap percentages was assessed; then, based on this assessment, a second route was selected for elimination, and the iterative process began again. Using this methodology, there were times when the elimination of one particular route led to a series of eliminations that was ultimately determined to be undesirable. In these cases, the methodology required returning to the original elimination and recommencing the iterative process.

- Remix Software Inc. was founded in 2014, and has gone on to partner with over 225 agencies across 10 countries. It provides a platform for planning public transit, automating the process of route and schedule scenario testing, letting planners draw routes onto a map and immediately see a potential schedule and fleet requirements. However, they’ve found that “some places will actively not change their service, even if they are suffering from critical issues like poor on-time-performance – simply because there is not enough staff time to redo the schedule.” [144]. They initially tried integer programming, but switched to use local search techniques instead, reporting that “Even for small instances of the problem, it could take hours. We tried using heuristics to speed things up by sacrificing provable optimality, but small problems still took 5 to 10 minutes. Our scheduling solver typically produced a solution in less than 30 seconds for problems of similar size.” [143]

Given ongoing work by colleagues in bringing optimization technology to the awareness of planning authorities and incorporating public feedback in [19], I am driven towards progress in providing tractable procedures with optimality guarantees that can generate high quality solutions within practical amounts of time.
1.1 Overview

This section summarizes the contributions in each chapter. The models developed in each chapter are individually different, but they all share a few properties: they are transparent in describing their objective and constraints, provide optimality guarantees, and can be solved with the solver technologies of today.

Chapter 2

Efforts at incorporating data into planning often involves the careful modelling of probabilistic assumptions to arrive at closed form expressions that are amenable to optimization. In earlier models such as the MEXCLP [56] and MALP [145], the requirement of tractability often restricts practitioners to probabilistic assumptions that do not fully reflect the data. Subsequent probabilistic models (such as the AMEXCLP [10]) that attempt to remedy the assumptions of earlier ones require solution algorithms that lack optimality guarantees.

In Chapter 2, we formulate data-driven approaches for ambulance deployment based on algorithmic advances in stochastic and robust optimization. Our formulations are solved to exact optimality within minutes, and outperforms previous approaches, requiring only 70% of the total number of ambulances required in probabilistic models to attain comparable out-of-sample performance. This will translate into millions of dollars in yearly savings without any changes in operational procedure for emergency medical services.

Chapter 3

There are a variety of existing studies on the problem of transit management subject to route choice. The general route choice setting we consider is one where each route may vary in its appeal, and commuters select the routes depending on each route’s relative appeal. One real-world drawback of timetabling is that complications such as traffic and variable dwell times make it difficult to adhere to a fixed timetable in congested urban settings. Another important component to the appeal of a route is
the price charged, and a variety of pricing policies have been implemented in transit systems across the world.

In Chapter 3, the consideration of frequency-setting and pricing in coordination motivates our use of choice modeling in representing commuters’ responses to transit operating controls. To our knowledge, ours is the first paper that addresses joint frequency-setting and pricing optimization for public transit. To demonstrate the flexibility of our framework, we ran it on two different networks: one from Kanagawa prefecture, Tokyo, Japan, and the other from Boston, Massachusetts. In both cases, we show that schedule frequencies and prices may be found that bring the system close to its optimal potential performance, as measured by comparison to the system optimum.

Chapter 4

The problem of designing a set of services to meet the commuting needs of a population is called the Transit Network Design Problem (TNDP). The TNDP is a well-studied challenging combinatorial problem, but advanced techniques have not been used to a significant extent in the planning process for actual network design. One of the main barriers to leveraging advanced techniques is scalability; many algorithms have not been proven on the scale that real transit networks require, which can be up to hundreds or thousands of stops.

In Chapter 4, we present a model that addresses the issues of interest to transit authorities, which are principally ridership, connectivity, and budget. We demonstrate our model’s scalability using real data from Boston, which has a network of hundreds of stops. Our algorithm converges to the optimal solution and produces high quality solutions within hours, making them practical for transit planners. In computational experiments, we show that optimization can help design transit networks that improves ridership by 6 to 13%.
Chapter 5

In many regions, individual travel needs often extend beyond the service area of a single public transportation agency. As a result, a high percentage of commuters need to make transfers between different services, making the synchronization of transit schedules of interest to operators and metropolitan planning organizations involved in the integration of transportation providers. However, most optimization models are unable to scale to transit networks of larger sizes and it remains unclear how to develop heuristics that scale up to solve for transit networks of practical interest, while providing optimality guarantees.

In Chapter 5, we develop scheduling models that provide high quality solutions that are close to optimal within minutes for smaller networks and within hours for larger networks. We use real transit networks and realistic origin-destination demand as our starting point, and provide data-driven formulations for transit networks with hundreds of services over thousands of locations. We provide a way for transit operators to balance in maintaining regularity in service timetables, while synchronizing schedules across different services. Finally, we perform a challenging case study on late night services for the Massachusetts Bay Transportation Authority (MBTA).
Chapter 2

Robust and Stochastic Formulations for Ambulance Deployment and Dispatch

In Emergency Medical Systems (EMS), operators deploy a fleet of ambulances to a set of locations before dispatching them in response to emergency calls, with the goal of minimizing the fraction of calls with late response times. We propose stochastic and robust formulations for the ambulance deployment problem that uses data on emergency calls to model uncertainty. By incorporating advances in column and constraint generation, our formulations are solved to exact optimality within minutes. In extensive computational experiments on Washington DC, our approach outperform previous approaches (i.e. the MEXCLP and MALP) that relied on probabilistic assumptions about the availability of ambulances. Our formulations achieve a reduction of 19 to 28% in number of shortfalls, requiring only 70% of the total number of ambulances required in probabilistic models to attain comparable out-of-sample performance.
2.1 Introduction

EMS systems have drawn a great deal of attention from researchers. While the public expects the availability of EMS facilities to provide timely services, this expectation is hard to realize due to limited available resources and stringent governmental budgets. Rising costs of medical equipment, increasing call volumes, and worsening traffic conditions have placed EMS providers under increasing pressure to meet performance goals set by regulators. A key indicator of these performance goals is medical response time, due to its relationship to specific time sensitive conditions such as out-of-hospital cardiac arrest, stroke and severe trauma cases.

Prior to receiving any emergency calls, ambulances are usually positioned within a set of pre-determined locations such as parking lots, hospitals, fire stations, or on the move when returning from servicing a call. When emergency calls arrive, ambulance operators might have to elicit the location of the call from the caller through landmarks or street descriptions. Therefore, the emergency calls are often modeled as arising from a fixed set of demand regions, after which the ambulances might be required to re-stock on emergency supplies.

Operational planning by EMS providers considers short-term decisions such as for ambulance dispatch and dynamic ambulance relocations. Tactical planning involves medium term decision horizons which typically establish baseline deployment plans and manpower shift schedules. Strategic planning involves longer term decision horizons such as the location of ambulance stations and ambulance fleet dimensioning. In essence, baseline tactical plans to guide operational decisions should be robust against short term uncertainties, whereas strategic plans should be robust against longer-term uncertainties. In this chapter, we focus on ambulance deployment at the tactical level and guide operational decisions to be robust against short term uncertainties.

2.1.1 Previous Work

Early ambulance deployment models focus on static policies for tactical planning. [162] formulated the set covering location problem (SCLP), which aims to minimize
the number of ambulances needed to cover a given region. [45] formulated the maximal covering location problem (MCLP), which aims to maximize the demand that can be covered given a fixed number of ambulances. Both the SCLP and MCLP consider single coverage in which a given point is covered if it can be reached within a response time threshold by an ambulance. However, both the SCLP and MCLP do not account for the possibility that a particular ambulance will be “busy” in the event of concurrent emergency calls. Therefore, subsequent formulations such as backup coverage models by [89] and double coverage models by [89] and [74], were introduced. There are also formulations that model ambulance availability from a probabilistic perspective. [56] approximate the expected value of coverage by introducing a “busy fraction” as a proxy for the probability that a given ambulance will be unavailable, and formulated the maximum expected covering location problem (MEXCLP). However, the “busy fraction” was assumed to be constant across all sites in the MEXCLP, which is not a realistic assumption. Therefore, it was extended by [10], [145], and [5] to account for site-specific probabilities. This resulted in nonconvex formulations based on steady-state probabilities of queuing systems which are heuristically solved through approximations. For an extensive review of models in the ambulance deployment literature, we refer the interested reader to [38], [106] and [12].

A related question is on the choice of dispatch rules: when an incident occurs, should the ambulance that is closest be dispatched? Although the “closest-idle policy” is suboptimal [41], there are few alternatives that have been suggested. One exception is the notion of a regionalized response by [159], where ambulances serve their allocated region first and the closest-idle ambulance is sent only if none in the region are unavailable. [6, 126] developed dispatch policies with prioritized patients to reduce response times for urgent patients at the expense of longer times for non-urgent requests. Nonetheless, it does not fully address how to dispatch vehicles to minimize late arrivals when all patients have high priority. Recent models by [75, 73, 123, 129, 2] focus on operational planning that repositions “idle” ambulances in real-time to better respond to future calls. This led [122] to establish bounds on the performance of an optimal ambulance redeployment policy. However, none of them
address the question of whether repositioning was required because of a suboptimal initial allocation of ambulances: it could be that an “optimal” static allocation of ambulances might attain similar benefits, obviating the need to reposition ambulances in real-time.

More recently, [16, 15, 181] revisited the static ambulance location problem and modelled the assignment of vehicles to emergency demands using chance constraints or robust optimization. However, since the models rely on strong restrictions on the dispatch policy for tractability, and often result in deployment plans that are overly conservative. On the other hand, fully adaptive models of dispatch often suffer from the “curse of dimensionality” and are typically computationally intractable [150, 60], and rely on methods that exploit problem structure for tractability [26, 86, 20, 141, 21].

With recent advances in exact solution techniques using column and constraint generation [180], large instances of fully adaptive robust formulations can now be solved within minutes. Given improvements in the quality of EMS data available, we are interested whether stochastic and robust formulations of deploy-and-dispatch models might outperform previous formulations (such as the MEXCLP and MALP) that are based on probabilistic assumptions about ambulance availability. In contrast to the literature on ambulance redeployment, we are interested in minimizing the fraction of late-arrivals, based on the closest-idle dispatch policy without requiring ambulances to be repositioned. Given the observation by [95] of a strong tradeoff between minimizing the fraction of late-arrivals versus response times, we demonstrate improvements in both measures via an exact mathematical optimization approach that we have not seen in the literature so far.

Although there is a connection in the two-stage integer formulation between our models, and the stochastic models by [16, 15], their models are based on probabilistic constraints, whereas our models are based on adaptive recourse functions. This makes our approach amenable to a robust formulation different from the approach by [181], that allows for integer uncertainty and recourse. [71] studied a similar recourse function in the setting of location-transportation problems, and took the same approach
of linearizing the inner bilevel maximization problem. Our work differs from theirs in the choice of application, uncertainty sets, and solution algorithm.

### 2.1.2 Our Contribution

This chapter makes the following contributions:

1. We propose tractable formulations for static ambulance deployment, namely: stochastic and robust two-stage planning models with fully adaptive recourse.

2. We adopt a data-driven approach to construct structured uncertainty sets that takes into account demand interactions across multiple local and regional levels jointly.

3. We provide extensions to our deployment models, that departs from traditional models of threshold coverage, to include notions of partial coverage based on response times.

4. Through realistic computational experiments, we demonstrate improvements in model performance over competitive models in the literature across multiple regimes of ambulance demand and availability, and provide reasons for the observed improvements.

The rest of the chapter is organized as follows. We review the literature in Section 2.1.1 on both ambulance deployment models, and multi-stage optimization. Next, we introduce the notation and preliminary concepts that are used throughout the paper in Section 2.2. We describe the models of ambulance deployment with recourse, and provide an algorithm for solving it in Section 2.3. We perform a realistic case study on the EMS system in Washington DC, and present an assessment of the proposed models through computational results in Section 2.4. We provide an explanation of the observed improvements in Section 2.4.2, and conclude in Section 2.5.
Table 2.1: Summary of the Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>Set of ambulance deployment locations</td>
<td></td>
</tr>
<tr>
<td>$J$</td>
<td>Set of ambulance demand regions</td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>Number of scenarios/time-periods</td>
<td></td>
</tr>
<tr>
<td>$t_{ij}$</td>
<td>Time (minutes) to travel from $i$ to $j$ $i \in I, j \in J$</td>
<td></td>
</tr>
<tr>
<td>$\bar{t}$</td>
<td>Response time threshold (minutes)</td>
<td></td>
</tr>
<tr>
<td>$E$</td>
<td>Set of directed edges $(i, j)$ satisfying $t_{ij} \leq \bar{t}$</td>
<td></td>
</tr>
<tr>
<td>$I_j$</td>
<td>Set of locations $i \in I$ connected to region $j$ $j \in J$</td>
<td></td>
</tr>
<tr>
<td>$J_i$</td>
<td>Set of regions connected to location $i$ $i \in I$</td>
<td></td>
</tr>
<tr>
<td>$\delta_j$</td>
<td>Set of regions geographically adjacent to $j$ $j \in J$</td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>Maximum number of ambulances</td>
<td></td>
</tr>
<tr>
<td>$\tau$</td>
<td>Maximum time (minutes) to complete a run</td>
<td></td>
</tr>
<tr>
<td>$d_j$</td>
<td>The number of calls from region $j$ $j \in J$</td>
<td></td>
</tr>
<tr>
<td>$\gamma_i$</td>
<td>Bound on total calls reachable from $i$ $i \in I$</td>
<td></td>
</tr>
</tbody>
</table>

2.2 From Online Dispatch to Ambulance Deployment

We use boldface letters for vectors and matrices; e.g. $\mathbf{1}$ and $\mathbf{0}$ to denote the vector of 1’s and 0’s accordingly. We use $\mathbf{x}^\top$ to refer to the transpose of $\mathbf{x}$. In addition, we define $(\cdot)^+ := \max\{\cdot, 0\}$, $[n] := \{1, \ldots, n\}$, $|S|$ as the cardinality of set $S$, and the $\alpha$-quantile $\mathbb{V@R}_\alpha(x) := \inf\{\ell \in \mathbb{R} : \Pr(x > \ell) \leq 1 - \alpha\}$.

In this section, we introduce some notation (see Table 2.1) and review the basic setup of an EMS system. We define ambulance deployment models on directed graphs $G = (V, E)$, where the set of vertices $V$ can be partitioned $V = (I, J)$ into the set of deployment locations $I$, and city regions $J$. We represent the network structure of regional accessibility through the node-arc adjacency matrix $B \in \{-1, 0, 1\}^{|V| \times |E|}$, where

$$b_{ik} = \begin{cases} 
1, & \text{if } i \text{ is the start of the } k\text{-th edge}, \\
-1, & \text{if } i \text{ is the end of the } k\text{-th edge}, \\
0, & \text{otherwise}.
\end{cases}$$

Given the partition $V = (I, J)$, we can decompose $B$ into sub-matrices $B_I$ and $B_J$ (See A.1 for an example). Our models aim to find the optimal way to locate $n$ ambulances at a set of potential locations $I$, to minimize the expected shortfall, by
making here-and-now decisions \( x = (x_i)_{i \in I}, \) where \( x_i \) is the number of ambulances to be deployed at node \( i \in I, \) on the basis of wait-and-see variables \( y = (y_{ij})_{(i,j) \in E}; \) where \( y_{ij} \) is the number of ambulances to dispatch from location \( i \in I \) to region \( j \in J. \)

### 2.2.1 Ambulance Scheduling and Dispatch

We consider an arrival process \( A : [0, t_{\text{max}}] \rightarrow \mathbb{Z}_{+}^{\lvert J \rvert}, \) where \( t_{\text{max}} = m \tau \) is the duration of the entire time period, and \( A(t) \) denotes the cumulative number of emergency calls received by time \( t. \) All ambulances take \( \bar{t} \) minutes to reach the site of each call, and \( \tau \) minutes to become available after being dispatched. We are interested in the performance of the static deployment policy \( x \) over the entire time period \( [0, t_{\text{max}}], \) and proceed by partitioning it into smaller time periods

\[
(0, \tau], (\tau, 2\tau], \ldots, (t_{\text{max}} - \tau, t_{\text{max}}],
\]

before optimizing for the demand \( A(i \cdot \tau) - A((i - 1) \cdot \tau) \) in each time period \( i = 1, 2, \ldots, m. \)

Focusing on a single time period \( [t(0), t(0) + \tau] \subset [0, t_{\text{max}}], \) let \( x^{(0)} \in \mathbb{Z}_{+}^{\lvert J \rvert} \) be the initial assignment of ambulance availability, and \( d^{(1)}, \ldots, d^{(k)} \in \mathbb{Z}_{+}^{\lvert J \rvert} \) be a sequence of requests at times \( t^{(1)}, \ldots, t^{(k)} \) such that \( t^{(0)} \leq t^{(1)} < t^{(2)} < \cdots < t^{(k)} \leq t^{(0)} + \tau. \) If an ambulance is dispatched at any time \( t \) in \( [t^{(0)}, t^{(0)} + \tau], \) it will remain unavailable for the rest of the time period \( [t, t^{(0)} + \tau]. \) Therefore, if we run out of ambulances and “queue” a call for the next available ambulance, it will take more than \( \tau \) minutes to respond to the call. Correspondingly, for each emergency request \( d^{(i)}, \) with a response of \( y^{(i)} \) (satisfying \( B_J y^{(i)} \leq x^{(i-1)} \)) ambulances dispatched, there will be a non-negative **shortfall** of \( d^{(i)} + B_J y^{(i)} \) incurred, and \( x^{(i)} \) remaining ambulances for the time period \( (t^{(i)}, t^{(0)} + \tau], \) where
\[ x^{(i)} = x^{(i-1)} - B_I y^{(i)} = (x^{(i-2)} - B_I y^{(i-1)}) - B_I y^{(i)} = \ldots = x^{(0)} - B_I (y^{(1)} + \ldots + y^{(i)}). \]

Defining \( d = \sum_{i=1}^k d^{(i)} \), and \( y = \sum_{i=1}^k y^{(i)} \), we have \( B_I y = B_I (y^{(1)} + \ldots + y^{(k)}) = x^{(0)} - x^{(k)} \leq x^{(0)} \). Therefore, to find the sequence of ambulance dispatches \( y^{(1)} \ldots y^{(k)} \) that minimizes the total shortfall, we solve the following problem

\[
Q(x^{(0)}, d) = \min_{y \in \mathbb{Z}^{|E|}: B_I y \leq x^{(0)}} \mathbf{1}^\top (d + B_I y)^+. \tag{2.1}
\]

Namely, for any sequence of dispatch decisions \( y^{(1)} \ldots y^{(k)} \), we have

\[
\sum_{i=1}^k (d^{(i)} + B_I y^{(i)})^+ \geq Q(x^{(0)}, \sum_{i=1}^k d^{(i)}).
\]

since \( y = \sum_{i=1}^k y^{(i)} \) is a feasible solution to (2.1).

### 2.2.2 Gradual Coverage

Most of the ambulance deployment models make a distinction between demands that are “covered”, and those that are not. [97] developed a partial coverage version of MCLP (MCLP-P) by using a sigmoid function to model the gradual decline of coverage along with the distance increase. [59] proposed a gradual coverage model with a stochastic distance threshold, using probabilistic analysis to calculate the expected coverage. See [62] for a review on other coverage decay functions.

To model gradual coverage, we introduce a dummy location \( i_0 \) to the set of locations \( I \), add the constraint \( x_0 \leq n \) to \( X \), and introduce dummy edges \((0, j)\) with
response times $t_{0,j} > \bar{t}$ for all $j \in J$. Then (2.1) can be re-written as

$$Q_\phi(x, d) = \min_{y \in \mathbb{Z}_+^{[E]}} \phi^T y$$  \hspace{1cm} (2.2)

subject to

$$d + B_J y \geq 0$$ \hspace{1cm} (2.3)

$$B_I y \leq x,$$ \hspace{1cm} (2.4)

with $\phi \in \mathbb{Z}_+^{[E]}$ defined as $\phi = (\phi_{ij})_{(i,j) \in E}$, where $\phi_{ij} := \mathbb{I}(t_{ij} > \bar{t})$.

By modifying the cost vector $\phi$, we can account for different notions of partial coverage based on the estimated travel time $t_{ij}$ from each location $i$ to each region $j$. For example, instead of the $0-1$ coverage, we can consider a cost function that reports the travel time in seconds by setting $\phi_{ij} = \lfloor 60t_{ij} \rfloor$. For consistency with established models of coverage (e.g. MEXCLP and MALP) in the literature, we perform a comparison for the formulations based on $0-1$ coverage, but report the performance of the models based on both response times and coverage (see Section 2.4).

### 2.3 Ambulance Deployment with Recourse

In this section, we are interested in the tactical decision of deploying ambulances to a fixed set of locations. Following the setup in Section 2.2.1, we have samples $(d^i)_{i=1}^m$ of emergency calls corresponding to aggregated demands for time periods $i = 1, \ldots, m$. A natural way of modelling the problem is to consider the following formulation

$$\min_{x \in \mathbb{X}} \mathbf{c}^T x + Q_\phi(x),$$ \hspace{1cm} (2.5)

where $\mathbf{c}^T x$ is the cost of deployment, $\mathbb{X}$ is the set of feasible ambulance deployments, and $Q_\phi(\cdot)$ is the recourse function that we use to measure the performance of any given deployment $x$ using the data $(d^i)_{i=1}^m$.

The setup is fairly general, and we provide some examples of $\mathbb{X}$ below:

1. $\mathbb{X}^{(1)} := \{ x \in \mathbb{Z}^{[J]}_+ \mid 1^T x \leq n \}$ could incorporate a constraint on the total number
of ambulances and drivers available.

2. \( \mathbf{X}^{(2)} := \{ x \in \mathbb{Z}_+^{|H|} \mid x \leq u \} \) could incorporate upper bounds \( u \) on the number of ambulances that can be deployed at each location.

3. \( \mathbf{X}^{(3)} := \{ x \in \mathbb{Z}_+^{|H|} \mid c^\top x \leq b \} \) could incorporate an operating budget \( b \) on the cost \( c^\top x \) of deploying the ambulances.

4. \( \mathbf{X}^{(4)} := \mathbf{X}^{(1)} \cap \mathbf{X}^{(2)} \cap \mathbf{X}^{(3)} \) could incorporate multiple considerations in the set of feasible ambulance deployments.

In the paradigm of two-stage optimization, we compare stochastic and robust models, by modelling \( Q_\phi(\cdot) \) as either

\[
Q^{\text{stochastic}}_\phi(x) = \mathbb{E}_{\hat{P}(d)} \left[ Q_\phi(x, d) \right],
\]

where \( \hat{P}(d) \) is a sample distribution over the possible scenarios, or

\[
Q^{\text{robust}(\alpha)}_\phi(x) = \max_{d \in \mathbb{D}(\alpha)} \left[ Q_\phi(x, d) \right],
\]

where \( \mathbb{D}(\alpha) \) is an appropriately chosen “uncertainty set” parameterized by \( \alpha \in (0, 1) \).

For the stochastic approach in (2.5) and (2.6), we construct the discrete distribution \( \hat{P}(d = d^i) = 1/m \) for all \( i = 1, \ldots, m \). Then a deterministic equivalent can be formulated and solved over both first stage decisions \( x \in \mathbb{Z}_+^{|H|} \) and second stage decision variables \( (y^i)_{i=1}^m \) [1, 28].

For the robust approach in (2.5) and (2.7), there are two issues. First, the demand for ambulances is often sparse (i.e. predominantly zero for any region in any given hour), and merits discussion to motivate its construction in (2.8). Second, conventional methods of solving robust optimization problems through a robust counterpart [14, 18], or cutting-plane approach [98, 161] does not immediately apply, since the recourse function is an integer optimization problem. Therefore, we develop an appropriate uncertainty set in Section 2.3.1, and describe a method of linearization in Section 2.3.2, before providing an algorithm for solving it in Section 2.3.3.
2.3.1 Structured Uncertainty Set

In this section, we develop uncertainty sets that can jointly model the interactions in emergency call demand across both local and regional levels. Existing models of uncertainty sets do not perform the desired function. The polyhedral uncertainty set in [23] is overly-conservative for column-wise uncertainty, but recent models of moment-driven uncertainty sets [57] and data-driven uncertainty sets [22] do not deal with sparse integer uncertainty. If we only provide bounds on groups of regions, adversarial clusters of demand tend to concentrate in local regions. On the other hand, if we only provide bounds on local regions, the uncertainty set will be overly conservative in its estimation of the overall demand. By incorporating both types of bounds (see Figure 2-1), we can construct an uncertainty set that is representative of the scenarios of interest.

To construct our uncertainty sets, we observe that emergency calls tend to follow a Poisson process that is inhomogeneous in both time and space. Therefore, sums of regional aggregated demands can be well-approximated by Poisson distributions with parameters $\hat{\gamma}$ estimated from data $d_1, \ldots, d^m \in \mathbb{Z}_+^{|J|}$, where

$$
\hat{\gamma}_{\text{single}}^j = \frac{1}{m} \sum_{l=1}^m \left[ \sum_{k \in \delta_j} d_{l}^j \right] \quad \forall j \in J,
$$

$$
\hat{\gamma}_{\text{local}}^j = \frac{1}{m} \sum_{l=1}^m \left[ \sum_{j \in J_\ell} d_{l}^j \right] \quad \forall j \in J,
$$

$$
\hat{\gamma}_{\text{regional}}^i = \frac{1}{m} \sum_{l=1}^m \left[ \sum_{j \in J_\ell} d_{l}^j \right] \quad \forall i \in I,
$$

$$
\hat{\gamma}_{\text{global}} = \frac{1}{m} \sum_{l=1}^m \left[ \sum_{j \in J} d_{l}^j \right].
$$

For ambulance operators to control the degree of risk aversion, we introduce a parameter $\alpha \in (0, 1)$ for scaling the uncertainty set, and define the bounds

$$
\gamma_j^{\text{single}}(\alpha) = \text{V@R}_\alpha(-\text{Poisson}(\hat{\gamma}_j^{\text{single}} + \epsilon)) \quad \forall j \in J,
$$

$$
\gamma_j^{\text{local}}(\alpha) = \text{V@R}_\alpha(-\text{Poisson}(\hat{\gamma}_j^{\text{local}} + \epsilon)) \quad \forall j \in J,
$$

37
\[\gamma_i^{\text{regional}}(\alpha) = \text{VaR}_\alpha(-\text{Poisson}(\gamma_i^{\text{regional}} + \epsilon)) \quad \forall i \in I,\]
\[\gamma^{\text{global}}(\alpha) = \text{VaR}_\alpha(-\text{Poisson}(\gamma^{\text{global}} + \epsilon)).\]

where the inclusion of the \(\epsilon > 0\) is introduced to deal with cases where the estimated demand is 0. We let \(\epsilon = 10^{-6}\) in our computational experiments. The resulting uncertainty set is as follows:

\[\mathbb{D}(\alpha) = \left\{ d \in \mathbb{Z}_+^{|J|} \mid\begin{aligned} d_j &\leq \gamma_j^{\text{single}}(\alpha) \quad \forall j \in J, \\ \sum_{k \in j} d_k &\leq \gamma_j^{\text{local}}(\alpha) \quad \forall j \in J, \\ \sum_{j \in I_i} d_j &\leq \gamma_i^{\text{regional}}(\alpha) \quad \forall i \in I, \\ \sum_{j \in J} d_j &\leq \gamma^{\text{global}}(\alpha) \end{aligned}\right\}. \quad (2.8)\]

As the polyhedral nature of the uncertainty set does not depend on the functional form of \(\gamma(\cdot)\), the choice of the Poisson model here is not fundamental. In practice, it can be replaced by the negative binomial distribution to deal with overdispersion in the count data, or with any other distributions that describes the dataset. The approach is data-driven, and does not rely on model-driven assumptions to provide any probabilistic guarantees on the resulting coverage provided by the model. In practice, ambulance operators should evaluate model performance through realistic simulations, and use cross validation to choose an appropriate \(\alpha\) (see Section 2.4.1). The following result shows that it is possible to include any scenario inside the uncertainty set (2.8) (at the expense of conservativeness).

**Proposition 1.** For any given scenario \(d \in \mathbb{Z}_+^{|J|}\), there exists a sufficiently small \(\alpha > 0\) such that \(d \in \mathbb{D}(\alpha)\).

### 2.3.2 The Recourse Function

In this section, we provide a tractable formulation for the recourse function under the robust formulation (2.7). Observe that \(y = \mathbf{0}\) is always a feasible solution to (2.1)
Figure 2-1: Different hierarchies of coverage for Washington D.C. Illustrates the geographical regions corresponding to (a) individual regions, (b) each region with those adjacent to it, (c) drivetime coverages from each station, and (d) the whole of Washington D.C.

for all $x \in Z_+^{|I|}$ and $d \in Z_+^{|J|}$, the recourse function $Q(x, d)$ is finite for all $x \in X$ and $d \in D(\alpha)$, and can be reformulated into a maximization problem for any given deployment $x \in Z_+^{|I|}$ and demand $d \in Z_+^{|J|}$.
**Proposition 2** (Recourse Duality). For a given $x \in \mathbb{Z}_+^{|I|}$ and $d \in \mathbb{Z}_+^{|J|}$, we have

$$Q_\phi(x, d) = \min_y \phi^\top y$$

s.t.  
\[
\begin{bmatrix} B_I \\ B_J \end{bmatrix} y \leq \begin{bmatrix} x \\ -d \end{bmatrix} \\
y \in \mathbb{Z}_+^{|E|}
\]

= $\max_{p,q} x^\top q - d^\top p$

s.t. $q^\top B_I + p^\top B_J \leq \phi^\top$

$p \leq 0, q \leq 0$.

This results in a mixed integer quadratic maximization problem

$$Q_{\phi}^{\text{robust}(\alpha)}(x) = \max_{d \in \mathcal{D}(\alpha)} Q_\phi(x, d)$$

= $\max_{d,p,q} x^\top q - d^\top p$

s.t. $q^\top B_I + p^\top B_J \leq \phi^\top$

$p \leq 0, q \leq 0, d \in \mathcal{D}(\alpha)$,

which is hard to solve in general. When $D(\alpha)$ is bounded, we can represent the integer vectors $d$ as the sum of a small number of binary vectors $d^{(1)}, \ldots, d^{(k)} \in \{0, 1\}^{|J|}$, and linearize the quadratic terms

$$-d^\top p = -p^\top (d^{(1)} + \cdots + d^{(k)}) = -1^\top (r^{(1)} + \cdots + r^{(k)})$$

where the logical constraints

$$r^{(i)}_{j} = \begin{cases} 
0 & \text{if } d^{(i)}_{j} = 0 \\
p_{j} & \text{otherwise.}
\end{cases}$$

are enforced at optimality for $i = 1, \ldots, k$. We arrive at the following integer linear
optimization model:

\[
Q_{\phi}^{\text{robust}(\alpha)}(x) = \max_{d^{(i)}, \ldots, d^{(k)}, p, q} x^\top q - 1^\top (\sum_{i=1}^{k} r^{(i)})
\]

\[
\text{s.t. } q^\top B_J + p^\top B_J \leq \phi^\top
\]

\[
r^{(i)} \geq p \quad \forall i = 1, \ldots, k
\]

\[
r^{(i)} \geq -M d^i \quad \forall i = 1, \ldots, k
\]

\[
r^{(i)} \leq 0 \quad \forall i = 1, \ldots, k
\]

\[
q^\top B_I + p^\top B_J \leq \phi^\top
\]

\[
p \leq 0, q \leq 0
\]

\[
\sum_{i=1}^{k} d^{(i)} \in \mathbb{D}(\alpha)
\]

\[
d^{(i)} \in \{0, 1\}^{|J|}.
\]

In practice, \(\mathbb{D}(\alpha)\) is bounded and we determine a small value for \(k\) from data.

2.3.3 Column and Constraint Algorithm

We now state a procedure for solving the robust formulation (2.5) with (2.7). It begins with an approximation \(\hat{Q}_{\phi}^{\text{robust}(\alpha)}(\cdot)\) of the recourse function, and generates a deployment plan \(x^*\) by solving

\[
x^* \in \arg\min_{x \in \mathbb{X}} c^\top x + \hat{Q}_{\phi}^{\text{robust}(\alpha)}(x),
\]

before evaluating \(x^*\) on the worst case scenario \(d^*\) over \(\mathbb{D}(\alpha)\) by solving

\[
d^* \in \arg\max_{d \in \mathbb{D}(\alpha)} Q(x^*, d).
\]

The algorithm terminates with the deployment plan \(x^*\) if \(\hat{Q}_{\phi}^{\text{robust}(\alpha)}(x^*) = Q_{\phi}^{\text{robust}(\alpha)}(x^*)\). Otherwise, it uses the generated scenario \(d^*\) to improve the approximation function \(\hat{Q}_{\phi}^{\text{robust}(\alpha)}(\cdot)\), and repeats the process. The algorithm is as follows:
1. Set $LB = -\infty$, $UB = \infty$ and $i = 0$.

2. Solve the following (restricted) master problem:

3. Update $LB = \eta^*$. If $UB - LB \leq \epsilon$, return $x^*$.

4. Solve

$$Q_\phi^{\text{robust}}(x^*) = \max_{d \in D(\alpha)} Q_\phi(x^*, d),$$

to obtain

$$d^* \in \arg\max_{d \in D(\alpha)} Q_\phi(x^*, d).$$

5. Update $UB = \min\{UB, Q_\phi^{\text{robust}}(x^*)\}$.

6. Create variables $y^i+1 \in \mathbb{Z}_+^{[E]}$ and add the following constraints

$$\eta \geq \phi^\top y^i+1$$

$$\begin{bmatrix} B_I \\ B_J \end{bmatrix} y^i+1 \leq \begin{bmatrix} x \\ -d^* \end{bmatrix}$$

$$y^i+1 \geq 0,$$

7. Update $i = i + 1$ and go to Step 2.

At each iteration of the algorithm, either the model and the oracle “converge” in their evaluation of the recourse function (when $UB - LB \leq \epsilon$), or the oracle generates a scenario $d$ from the uncertainty set $D(\alpha)$ to be added the model. When $D(\alpha)$ is finite in size (e.g. bounded and integer), the algorithm is guaranteed to converge:

**Proposition 3** (Finite Convergence ([180])). *The C&CG algorithm for the robust problem converges in a finite number of iterations.*
2.4 Computational Results

In this section, we perform computational experiments on a realistic case study of the national EMS system for Washington D.C.. The experiments are based on data released through a Freedom of Information Act (FOIA), geo-coded by CodeForDC, and made available at https://github.com/codefordc/ERDA. The dataset includes emergency calls for both fire trucks and ambulances, and contains 179,160 relevant EMS records over the period 1 Jan 2012 to 31 March 2013. We use data from the first three months of 2012 for generating the models, and data from the remaining twelve months for evaluating the model performance (see Section 2.4.1). As commuters from the surrounding Maryland and Virginia suburbs increase the city’s population during the workweek (see Figure 2-2), we only consider emergency calls for the weekdays, as the same procedure can be done separately for the weekend deployment schedules.

![Figure 2-2](image)

Figure 2-2: Average hourly number of emergency calls per region. This is based on data of emergency calls requiring ambulatory care for DCFEMS, from Jan 2012 to Mar 2013.

We partition the city into 217 regions based on neighborhood zones from the Washington Post\(^1\), and obtain geographical locations of the fire stations from http://opendata.dc.gov/. We estimate the coverage of each neighborhood from the respective fire stations, through a shortest path drivetime analysis that filters out regions that took more

\(^1\)The Washington Post derived the neighborhood boundaries by reviewing original subdivision data, and consulting community sources.
than 10 minutes of drivetime (see Figure 2-3 for an example), using data from OpenStreetMap [83]. The traveling time takes into account the heterogeneity of different road types.

### 2.4.1 Experimental Setup

To reflect model performance under different ambulance fleet sizes, we vary the number of ambulances from 10 to 50 in increments of 5. To evaluate the models’ performances, we ran a discrete-event simulation with interarrival timings between emergency activations based on historical data. Upon activation for medical and trauma emergencies, the closest available ambulance from one of the covering locations will be dispatched to service the call\(^2\). In the event none of the ambulances are available, the call will be queued for the next returning ambulance.

---

\(^2\)In regimes of high ambulance availability, it has been shown to be close to optimal in many studies [109, 126, 163].
estimated drivetimes, and a residual turnover time\textsuperscript{3} that is lognormally distributed. To compare model performance, we track both the simulated response time and whether it was within 10 minutes, for each simulated emergency call. Based on a simulation spanning 12 months, we compute the monthly average and variance in both the hourly shortfall and individual response time.

**Homogeneity of Emergency Calls**

We generate deployment schedules for both the “peak period” from 8am to 8pm, as well as the “off-peak period” from 8pm to 8am, to pragmatically reflect the performance of the models under different demand profiles. From observation (see Figure 2-2), the demand during the peak period tends to be more homogeneous, whereas the demand during the off-peak period is time-inhomogeneous, so the experiments reflect a variety of realistic situations in actual operations.

We generate the demands for both peak and off-peak periods in the following way: for the emergency calls from April 2012 to March 2013, we calculate the interarrival timings for the remaining records by taking its difference from the emergency call preceding it. For the first call of each peak (off-peak) period, we take the difference of its timing from the last call of the previous period, after subtracting the 12 hours of time difference between the two periods. This will generate a sequence of emergency calls that maintains the characteristics of both peak (off-peak) periods in our simulations.

**Choice of Uncertainty Set**

In our experiments, we report the performance of the robust model under different levels of $\alpha$ in the test set: smaller values of $\alpha$ corresponds to larger uncertainty sets, and correspond to a higher degree of risk aversion. The model tends to perform poorly when it does not consider enough scenarios (if $\alpha$ is too big), or gets biased by “extreme” scenarios (if $\alpha$ is too small). See Table A.1 for the deployment plans

\textsuperscript{3}The turnover time includes the time spent at scene, conveyance to hospital, and return to base, after which it will be made available for serving the next emergency call.
generated at different levels of $\alpha$: as $\alpha$ becomes smaller, the size of the uncertainty set $D(\alpha)$ increases, and the more the number of ambulances the model can accommodate before it gets saturated (indicated by the shaded cells).

If the recourse function $Q^{\text{robust}}(\alpha)(x)$ is 0 for some $x \in \mathbb{X}$, then we have some degree of freedom in deciding between different optimal deployment plans in the set $\{x : Q^{\text{robust}}(\alpha)(x) = 0\}$. In such cases, we shade the corresponding values to indicate that it is an invalid result. In practice, it might be appropriate to account for other considerations (such as cost or response time) as a secondary measure. Nonetheless, the issue is handled when determining the value of $\alpha$ through cross-validation for the robust formulation. For that reason, the optimal choice of $\alpha$ is data-driven and specific to each city, and we do not provide a general prescription here. Instead, we recommend that practitioners perform a cross-validation procedure on a holdout set for the best choice of $\alpha$ to get competitive results in practice.

**Volatility of Turnaround Times**

To examine the robustness of model performance to the uncertainty in the turnaround time of the ambulances, we add a lognormally distributed turnaround time $\tilde{t}_{\text{turnaround}}$ to the response times $t_{ij}$ for each edge $(i, j) \in E$, such that $t_{ij} + \tilde{t}_{\text{turnaround}}$ is the overall time it takes for an ambulance to become available after it was dispatched for service from node $i \in I$ to node $j \in J$. By varying the parameters of the lognormal distribution for $\tilde{t}_{\text{turnaround}}$, we simulate the following three environments:

(i) volatile: $\tilde{t}_{\text{turnaround}} \sim \text{lognormal}(\mu = 3.57, \sigma = 0.5)$,

(ii) normal: $\tilde{t}_{\text{turnaround}} \sim \text{lognormal}(\mu = 3.65, \sigma = 0.3)$, and

(iii) stable: $\tilde{t}_{\text{turnaround}} \sim \text{lognormal}(\mu = 3.69, \sigma = 0.1)$.

The choice of a lognormal distribution was based on support from empirical data in studies [101] and its use in optimization models [2, 126]. The three environments correspond to turnaround distributions with the same mean of 40.25 (minutes) but different standard deviation (4.0 for stable, 12.3 for normal, and 21.3 for volatile; all units in minutes).
Models under Comparison

We make comparisons of both the stochastic and robust formulations (equation (2.5) with (2.6) and (2.7) respectively) with their probabilistic counterparts MEXCLP and MALP. For a fair comparison of the models, we set the first stage deployment costs $c^\top x$ to 0, and used the $0 - 1$ coverage loss (described in Section 2.2.2). We hereby describe the formulations.

The MEXCLP by [56] introduces the notion of a “busy fraction” $q \in (0, 1)$, as the probability that any given ambulance will be unavailable to respond to an incoming emergency call. Assuming the independence of ambulance availabilities, the resulting objective function is given by

$$\max \sum_{k=1}^{n} (1 - q)q^{-1}d^\top z^{(k)},$$  \hspace{1cm} (2.9)

where $n$ is the maximum number of ambulances, and $z^{(k)} \in \mathbb{R}^{|J|}$ is such that $z^{(k)} = [z^{(k)}_1, \ldots, z^{(k)}_J]$ with $z^{(k)}_j$ equal to 1 if and only if region $j \in J$ is covered by at least $k$ ambulances. As the objective function (2.9) is convex in $k$, the resulting model can be written as

$$\max \sum_{k=1}^{n} (1 - q)q^{-1}d^\top z^{(k)}$$

s.t. $A^\top x \geq z^{(1)} + \cdots + z^{(n)}$  \hspace{1cm} (MEXCLP)

$$1^\top x \leq n$$

$$z^{(1)}, \ldots, z^{(n)} \in \{0, 1\}^{|J|}, \quad x \in \mathbb{Z}^{|J|}$$

[145] subsequently formulated the Maximum Availability Location Problem (MALP), which introduced the notion of a reliability level $\alpha$, and sought to maximise the demand covered with probability $\alpha$ through the use of chance constraints. By linearizing the expression $1 - q\sum_{i \in I_j} x_i \geq \alpha$ into $\sum_{i \in I_j} x_i \geq \lceil \log(1 - \alpha) / \log q \rceil =: b$ for all $j \in J$, and defining binary vectors $z^{(1)}, \ldots, z^{(b)}$ as in MEXCLP, they arrive at
\[
\max \ d^T z^{(b)} \\
s.t. \ A^T x \geq z^{(1)} + \cdots + z^{(b)} \\
z^{(k)} \leq z^{(k-1)} \quad (k = 2, \ldots, b) \\
1^T x \leq n \\
z^{(1)}, \ldots, z^{(b)} \in \{0, 1\}^{|J|}, \quad x \in Z^{|I|} 
\]

(MALP)

In our experiments, we ran the MEXCLP and MALP with different values of \(q\), and used cross-validation to pick the one with the best performance in out-of-sample scenarios. This came up to the value 0.654.

Computational Setup

We implement the models in the Julia programming language [25] using the package “Julia For Mathematical Programming” (JuMP) developed by [113]. Gurobi 6.0 was used as a solver for all the models, and experiments were run on a MacBook Pro with a 2.6 GHz Intel Core i5 processor, with 16GB DDR3 RAM.

2.4.2 Discussion of Results

For both the MEXCLP and MALP formulations, the models were each solved in less than a minute for all instances. For the stochastic formulation, we formulated its deterministic equivalent with 500 scenarios, which took 1 to 2 GB of RAM, and roughly a minute to solve. For the robust formulation, we begin with a naïve initial deployment plan, and trace the sequence of scenarios generated by the C&CG procedure (see Figure A-1). In the running of the algorithm (see Section 2.3.3), the upper and lower bounds converges in less than 20 iterations. For example, with 35 ambulances and \(\alpha = 0.01\), it took 17 iterations for the algorithm to converge to the optimal solution. The total solve time for all values of \(\alpha\) and ambulances has been observed to take less than 10 minutes.

Although they provide a similar level of coverage for each location, the robust formulation tend to concentrate the ambulances in a few “hubs”, whereas the stochas-
tic formulation distributes the ambulances more evenly distributed over the locations (see Figure 2-4). In contrast, both the MEXCLP and MALP formulations tend over-concentrate their ambulances within a few “hubs” when there is an abundance of ambulances (see Table A.2), with a few locations seeing more than 10 ambulances and most locations being assigned none.

![Deployment Coverages](image)

(a) Robust Deployment Coverage
(b) Stochastic Deployment Coverage

Figure 2-4: Deployment Coverages for the Robust and Stochastic deployment models with 35 ambulances. The regions are shaded by counting the number of ambulances available within 10 minutes. The size of the orange circles corresponds to the number of ambulances allocated to that location.

From the results (see Tables A.4 and A.3 to A.10), the largest gains in performance comes from increasing the number of ambulances available, and the differences in performance between the stochastic and robust formulations are small. On the whole, both the stochastic and robust formulations provide competitive performance with different numbers of ambulances, and demonstrate clear performance gains over their probabilistic counterparts MEXCLP and MALP as the number of ambulances increases. This is expected as the stochastic and robust formulations consider the dispatch decisions, where the MEXCLP and MALP do not.

In particular, at $n = 50$ ambulances, both the stochastic and robust deployments have an average shortfall of less than 3.7 and 3.53 emergency calls for 360 hours of peak-hour ambulance operations, while the MEXCLP and MALP deployments have a significantly higher shortfall of 4.36 and 4.89 respectively (see Table A.4). This corresponds to an improvement (decrease) of approximately 22% of shortfalls after switching from probabilistic (MEXCLP and MALP) models to data-driven (stochas-
tic and robust) models. Translated into operational terms, this means an average of 1 fewer incident experiencing a “late arrival” a month, which is a significant improvement given that these are rare events that should only occur for 2% of incidents during the same period.

This improvement does not come at the expense of response times. In fact, there is a corresponding improvement of approximately 34.6% in response times with 50 ambulances under peak-hour conditions. This is a surprising result, as the deployment models are concerned only with coverage based on the response time threshold and do not explicitly optimize for response times, even though the dispatch policy (for all deployment models) is based on the closest-idle ambulance. Similar results are observed for off-peak hour response times, with an improvement of approximately 23% in response times at 50 ambulances.

To explain the differences in performance, we look at the proportion of time an ambulance is busy:

$$f = \frac{(\bar{t}_{\text{response}} + \mathbb{E}[\bar{t}_{\text{turnaround}}]) \cdot \bar{n}_{\text{call}}}{60 \cdot n},$$

where \(\bar{n}_{\text{call}}\) is the number of calls in an hour. We call \(f\) the busy fraction. In our experiments, \(f\) is less than one when \(n \geq 25\) during peak periods and \(n \geq 15\) during off-peak periods. As the data-driven models perform better than the probabilistic models when \(n \geq 25\) during peak periods and \(n \geq 15\) during off-peak periods, the busy fraction indicates that the data-driven models perform better when there is some slack in the system. The results are similar across both steady and volatile environments, suggesting that the volume of emergency calls matter more than service time volatility for performance comparison.

To put the improvements in perspective, it takes approximately 35 ambulances for the data-driven models to have comparable performance with the probabilistic models at 50 ambulances. This allows EMS operators to save on operating costs for up to 15 ambulances while maintaining guarantees of the same service quality. According to [87], it costs close to half a million dollars per year to staff an advanced life support (ALS) ambulance on 24/7 basis. Therefore, this corresponds to approximately 7 million dollars in savings per year. Moreover, all of these is made possible by better deployment plans that does not require complex technology, and does not preclude
the possibility of further improvements through the redeployment and redispachtch of ambulances in real-time.

2.5 Conclusion

To conclude, recent economic developments have placed EMS providers under increasing pressure to meet performance goals set by regulators, often with a limited governmental budget. We provide a data-driven approach to ambulance deployment that takes into account demand interactions across regions, and improves upon probabilistic models through adaptive recourse functions that capture the essential dynamics of modelling both sparse demands and concurrent emergency calls. We demonstrate the practical tractability of the formulations, and their competitive performance across a broad range of environments in a realistic setting of Washington DC.
Chapter 3

Frequency-Setting and Pricing on Multi-Modal Transit Networks

Modern public transportation systems are increasingly complex: they are operated at a large scale, must support booming urban populations, and run under tight budget constraints. Additionally, passengers are able to make choices between a variety of commuting options. We develop formulations for minimizing system wait time in multi-modal networks, while accounting for operator budget constraints, capacity constraints, and passenger preferences. Furthermore, our algorithms run to near-optimality in minutes for city-sized networks. We demonstrate the benefit of setting schedule frequencies and prices jointly through case studies on real data from Boston and Tokyo. To our knowledge, ours is the first paper that addresses joint frequency-setting and pricing optimization for public transit networks and at scale.

3.1 Introduction

Public transit systems face significant operating challenges, and the question of how to efficiently manage such systems is of crucial importance. In recent years, increasing urban populations and shrinking operating budgets have both contributed to significant overcrowding and delays during peak hours [68]. Meanwhile, transit networks are growing and diversifying rapidly, making it increasingly important to understand how
operating decisions influence ridership and demand between different route options.

The controls available to improve transit operations are numerous. Although an optimized schedule might improve transit operations across multiple services, further coordination can be attained in capacity-constrained networks through the effectiveness of pricing schemes. In this regard, we are motivated by modern developments in both private ride-sharing services such as Uber and Lyft, and public infrastructure such as electronic road pricing schemes in London, Singapore, and Stockholm. In this paper, we will focus on coordinated frequency-setting and pricing, while accounting for commuter preferences between multiple route choices.

3.1.1 Literature Review

Optimal scheduling of transit services has been well-studied in the literature. The train timetabling problem in particular has been studied under numerous variants. [160] first proposed an integer optimization formulation to minimize total travel time subject to coordination constraints. Since then, objectives such as profit [37, 177] or closeness to an ideal timetable [40] have also been studied. The timetabling setting that is closest to our commuter-centric focus was studied by [134], who formulated an integer optimization model that sought to minimize total commuter waiting time on a single congested urban rail line, and solved it using a genetic algorithm. [42] similarly sought to minimize total commuter waiting time, but in a multi-line setting, and solved the optimization problem using a Lagrangian-based heuristic.

One real-world drawback of timetabling is that complications such as traffic and variable dwell times make it difficult to adhere to a fixed timetable in congested urban settings. Frequency-setting addresses this challenge by determining departure rates over longer time horizons; the key difference here is that the timetabling problem is a discrete optimization problem, whereas in frequency-setting, the departures of buses and trains are modeled as continuous or fluid flows. Early work focused on analytical results on idealized single-origin-destination, infinite-capacity routes: [131] computed dispatch rates to minimize passenger waiting time, while [91] extended these results to a multiobjective framework that also incorporates operating costs. To account for
multiple origins and destinations and capacity constraints, [148] formulated a more general nonlinear optimization problem that set bus schedule frequencies to minimize total commuter waiting time.

The above timetabling and frequency-setting examples make demand to be fixed and known. A natural extension to these transit management problems arises when demand is not a fixed input, but rather is influenced by the controls that a transit operator applies. An early paper by [70], which set headways to maximize social benefit, addressed the elasticity of demand by taking ridership to be a function of bus headways. However, the major simplifying assumption they take, that demand on each bus line is independent from that on other lines, is not entirely realistic. In modern transit settings, commuters are faced with a variety of substitutable route choices, and demand spilled by one route may be recaptured by another.

The general route choice setting we consider is one where each route may vary in its appeal, and commuters select the routes depending on each route’s relative appeal. Common assumptions in modeling route choice are that (i) vehicles arrive randomly, and have headways following an exponential distribution with mean equal to the inverse of schedule frequency, and (ii) when presented with multiple attractive route options, commuters will choose that which has the first arriving vehicle [154, 132].

Another important component to the appeal of a route, in addition to the schedule frequencies, is the price charged. A variety of pricing policies have been implemented in transit systems across the world. In New York City, the Metropolitan Transit Authority charges a flat fare for all subway and bus trips, while the Long Island Rail Road employs both zone-based fare and peak-hour congestion pricing. In traffic settings, congestion pricing has also been employed in cities such as London [11], Stockholm [63], and Singapore [76].

The consideration of frequency-setting and pricing in coordination motivates our use of choice modeling in representing commuters’ responses to transit operating controls. The classical choice model is the multinomial logit (MNL) model, which was first proposed in [125] in the context of choosing residential locations. Under the MNL model, commuters randomly select a route from all available route options;
typically, options with higher utilities are selected more frequently. The problem of fitting transit logit models that account for features such as travel time and cost is well-studied [135, 36]. For a broad overview of choice modeling as it applies to route choice in transit, see [13, 142, 34].

There are a variety of existing studies on the problem of transit management subject to route choice. [85] studied a simplified model of a congested bus network where route choices are proportional to the relative frequencies of the bus routes, and demonstrated their results on a toy six-station network. [151] assumed that passengers respond to the changes in the vehicle frequencies according to a slightly modified version of the passenger assignment model of [85], and used an iterative approach to set vehicle sizes and schedule frequencies. [51] followed the commuter route choice strategy outlined in [154] to address commuters’ responses to transit schedules, and developed a bilevel optimization framework to set schedule frequencies accordingly. [179] modeled a similar problem, but additionally accounted for the fact that for buses arriving less frequently, passengers are likely to synchronize their arrivals with a given timetable. [139] used an all-or-nothing route assignment model that assigned passengers to the route of highest utility, where utility was based on total travel time, and developed timetables that would synchronize transfers between bus lines. [167] use bilevel optimization to set schedule frequencies that minimize waiting time in the upper level while modeling passenger responses to the frequencies in the lower level. [146] use integer optimization to set timetables that maximize operating profit while ensuring that passengers maintain a minimum level of utility.

The price elasticity of demand has been studied in several separate contexts from scheduling. For congestion pricing in transit, [90] assumed logit preferences that respond to price, while taking the schedule as fixed, and derives equilibrium conditions on a toy two-node system with two route options. [174] and [175] set tolls and public transit prices in a coordinated, multi-modal network equilibrium model to reduce total travel time in a combined traffic-transit network. In the field of airline operations, [3] also assumed logit preferences that respond to price, and then performed fleet assignment and pricing in coordination.
3.1.2 Our Contribution

We provide a unified framework that allows us to:

1. Jointly optimize schedule frequencies and prices together,

2. Represent dynamic demand that responds to transit operating decisions through commuter route choice modeling, and

3. Tractably model city-scale networks and quantify the impact of policy changes to congested transit systems.

To our knowledge, ours is the first paper that addresses joint frequency-setting and pricing optimization for public transit networks and at scale. To demonstrate the flexibility of our framework, we ran it on two different networks: one from Kanagawa prefecture, Tokyo, Japan, and the other from Boston, Massachusetts. We observe that coordinated frequency-setting and pricing and the detailed modeling of commuter choices can allow for transit operations that incentivize commuters into route choices that benefit system performance, and our model provides a way of quantifying this benefit. We show in two computational case studies that schedule frequencies and prices may be found that bring the system close to its optimal potential performance, as measured by comparison to the system optimum.

The rest of this paper is organized as follows. Section 3.2 defines notation used throughout the paper. In Section 3.3, we elaborate on the recourse function that we use for evaluating total waiting time. In Section 3.4, we consider choice models for commuter preferences. In Section 3.5, we perform computational experiments on two real-world transit networks. Finally, we offer concluding remarks in Section 3.6.

3.2 Preliminaries

We consider the problem of scheduling transit services, such as buses, subways, or trains, on a network when commuters have multiple options available to them. As
each service is scheduled to travel along a fixed sequence of stops, we will call these sequences of stops lines.

Table 3.1: Summary of the Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>budget available for scheduling transit services (buses, subways, trains)</td>
</tr>
<tr>
<td>$c^\ell$</td>
<td>cost to deploy a transit service along line $\ell$, which can vary depending on characteristics of line $\ell$ such as distance or mode</td>
</tr>
<tr>
<td>$K^\ell$</td>
<td>capacity of transit service running on line $\ell$</td>
</tr>
<tr>
<td>$L$</td>
<td>number of transit lines</td>
</tr>
<tr>
<td>$T$</td>
<td>number of time periods</td>
</tr>
<tr>
<td>$\Delta(u, v, r)$</td>
<td>time periods it takes to travel from stop $u$ to stop $v$ using route $r$, while in-vehicle</td>
</tr>
<tr>
<td>$\text{legs}(u, v, r)$</td>
<td>the sequence of lines that a commuter would take from stop $u$ to stop $v$ while using route $r$</td>
</tr>
<tr>
<td>$\text{routes}(u, v)$</td>
<td>the set of all routes that can take a commuter from origin $u$ to destination $v$</td>
</tr>
<tr>
<td>$\text{stops}(\ell)$</td>
<td>the set of all stops on line $\ell$</td>
</tr>
</tbody>
</table>

Within the transit network, we refer to each origin-destination pair $(u, v)$ as a commute. Each commute may be associated with a number of different route options, each of which is a different sequence of lines that can take a commuter from origin $u$ to destination $v$. The travel on each individual line in the sequence is called a leg of the commute. The problem parameters are described in Table 3.1, and definitions of routes and legs are illustrated in Figure 3-1, a screenshot taken from Google Maps. In this example, the commute of interest is from Yokohama Station to Ofuna Station. Two route options are offered: the first route consists of two legs (Yokosuka and Tokaido), and the second route consists of a single leg (Negishi).

Given data on $d = (d_t^{u,v})$, where $d_t^{u,v} =$ demand for commute $(u, v)$ that arrives at station $u$ at time $t$, we wish to construct a schedule $x$ that minimizes total commuter waiting time. In practice, demand would also depend upon the choice (if any) of routes that the commuters might take through the network. We use $\theta = (\theta_t^{u,v,r})$ to represent commuter choices, where each element $\theta_t^{u,v,r}$ represents the proportion of commuters for commute $(u, v)$ at time $t$ who chose to take route $r$.

For transit scheduling, we discretize the full schedule into time periods $t =$
1, \ldots, T, and are interested in providing decisions on $x = (x^\ell_t)_{t=1,\ldots,T}$, where

$$x^\ell_t = \begin{cases} 1 & \text{if a transit service is scheduled for departure from the start of line } \ell \text{ at time } t, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the set of feasible schedules is denoted by

$$\mathcal{X}_B = \left\{ x \in \{0, 1\}^{T \times L} : \sum_{t=1}^T \sum_{\ell=1}^L c^\ell x^\ell_t \leq B \right\}. \tag{3.1}$$

The constraint in (3.1) enforces the requirement that the total number of services cannot exceed that allowed by the budget $B$. The cost $c^\ell$ is allowed to vary by each line $\ell$, to account for the fact that different modes of transportation are associated with different costs. The linear optimization relaxation

$$\bar{\mathcal{X}}_B = \left\{ x \in [0, 1]^{T \times L} : \sum_{t=1}^T \sum_{\ell=1}^L c^\ell x^\ell_t \leq B \right\}. \tag{3.2}$$

of the budget constraint (3.1) represents the set of fluid rates of departure for the transit services, rather than a concrete timetable for operators to follow. For example, a value of $x^\ell_t = 0.5$ means that a train would depart once every two time periods.
The practical interpretation of the fluid model is that it primarily models waiting time due to capacity constraints, while ignoring waiting time incurred if commuters arrive between service arrivals. This simplification is useful, for example, in peak traffic hours when there is high congestion and trains arrive at a high frequency; here, the capacity restrictions, not the train interarrival wait time, are of primary concern. Another motivator of frequency-setting as opposed to timetabling is that during congested peak-hours, frequency guidelines are more implementable than exact timetables. Hereafter, we focus on the relaxed constraint (3.2), which only requires solving a linear optimization problem.

The above formulation contains some basic operational considerations to a transit authority, but can easily be expanded to incorporate other specific constraints that might be of interest. We provide some examples below.

1. **Lower and upper bounds on schedule frequencies.** It may be desirable to ensure a minimal level of service on a particular line, or it may be impossible to operate services above a particular level of frequency without collisions. If $B$ and $\overline{B}$ are the service lower and upper bounds, then these requirements can be modeled with the following constraint:

$$B \leq x \leq \overline{B}.$$

2. **Service coordination.** Certain network structures may require that transit services on separate lines share common resources, such as a lane in the train tracks. Suppose that lines $\ell_1$ and $\ell_2$ are two such lines, and that it takes $\delta_1$ and $\delta_2$ time periods for trains on lines $\ell_1$ and $\ell_2$ to reach the point of conflict, respectively. To ensure that the shared area can only accommodate a single service for any given period, we can introduce the constraint

$$x_{\ell_1}^{t+\delta_1} + x_{\ell_2}^{t+\delta_2} \leq 1$$

for all relevant time periods $t$.  

60
3.3 Optimization with Known Choices

To model the progress of the commuters through the network, we introduce boarding variables $z = (z_{u,v,r,i}^t)$, where $z_{u,v,r,i}^t$ is the number of commuters traveling from origin $u$ to destination $v$, taking route option $r$, on the $i$-th leg of their itinerary, boarding a service that had started at time $t$. For clarity of presentation, we write our formulation ignoring vehicle travel time between stations; accounting for vehicle travel time is ultimately a straightforward procedure of adjusting the time indices, but comes at the expense of significantly heftier notation. In this section, we also make the assumption that the commuter choice probabilities $\theta$ are known, and satisfy $0 \leq \theta_{u,v,r}^t \leq 1$ and $\sum_{r \in \text{routes}(u,v)} \theta_{u,v,r}^t = 1$. This assumption is only for simplicity of exposition and will be revisited in Section 3.4. Then the transit frequency-setting problem can be stated as:

$$\min_{x \in \mathcal{X}_B} Q(x, \theta),$$

where

$$Q(x, \theta) = \min_{z \geq 0} J(z, \theta)$$

s.t. $O_{\ell,u,t}(z) \leq K_{x,\ell}^t$

$\forall \ell = 1, \ldots, L,$

$\forall u = \text{stops}(\ell),$

$\forall t = 1, \ldots, T$ (3.4a)

$$\sum_{t' = 1}^t z_{u,v,r,1}^{t'} \leq \sum_{t' = 1}^t d_{u,v}^{r} \theta_{t'}^{u,v,r}$$

$\forall u, v = 1, \ldots, N,$

$\forall r = \text{routes}(u,v),$

$\forall t = 1, \ldots, T$ (3.4b)

$$\sum_{t' = 1}^t z_{u,v,r,i}^{t'} \leq \sum_{t' = 1}^{i-1} z_{u,v,r,i-1}^{t'}$$

$\forall u, v = 1, \ldots, N$

$\forall r = \text{routes}(u,v),$

$\forall i = 2, \ldots, |\text{legs}(u,v)|,$

$\forall t = 1, \ldots, T.$ (3.4c)
The function $J(z, \theta)$ in (3.4a) computes the total waiting time, by adding the number of commuters who have arrived or transferred to each station and subtracting the commuters who have boarded their transit. More specifically, the total number of passengers waiting at a station $u$ on line $\ell$ at time $t$ is given by

$$AD_{\ell,u,t}(\theta) + XD_{\ell,u,t}(z) - BD_{\ell,u,t}(z),$$

(3.5)

where $AD_{\ell,u,t}(\theta)$ represents “arriving” demand, $XD_{\ell,u,t}(z)$ represents “transferring demand”, and $BD_{\ell,u,t}(z)$ represents “boarding demand”, all cumulative up until time $t$. The total time passengers spend waiting across all lines, stations, and time periods is then obtained by aggregating equation (3.5) as follows:

$$J(z, \theta) := \sum_{\ell=1}^{L} \sum_{u \in \text{stops}(\ell)} \sum_{t=1}^{T} AD_{\ell,u,t}(\theta) + XD_{\ell,u,t}(z) - BD_{\ell,u,t}(z).$$

(3.6)

In equations (3.5) and (3.6), the arrivals quantity $AD_{\ell,u,t}(\theta)$ represents the total demand that has arrived to station $u$ on line $\ell$ by time period $t$, and is computed as follows:

$$AD_{\ell,u,t}(\theta) := \sum_{\{(v,r) : \text{legs}(u,v,r)_1 = \ell\}} \sum_{t'=1}^{t} d_{u,v,t'} \theta_{u,v,r}^{u,v,r},$$

(3.7)

where the condition $\{(v, r) : \text{legs}(u, v, r)_1 = \ell\}$ indicates that the first leg of the commute must be on line $\ell$.

The transferring demand quantity $XD_{\ell,u,t}(z)$ represents the total number of passengers who have arrived to station $u$ on line $\ell$ by time period $t$, having transferred over from another line. It is computed as follows:

$$XD_{\ell,u,t}(z) := \sum_{\{(w,v,i) \in \text{xfrthru}(\ell, u)\}} \sum_{t'=1}^{t} z_{w,v,i-1}^{w,v,i},$$

(3.8)

where the set $\text{xfrthru}(\ell, u)$ represents all of the commute-route-legs that make a transfer through station $u$ on line $\ell$. Specifically, when considering a commute $(w, v)$, route option $r \in \text{routes}(w, v)$, on the $i$th leg of the commute, $(w, v, r, i)$ must satisfy
the criteria that (i) \( i \geq 2 \), (ii) the transfer station connecting the \((i - 1)\)th and \(i\)th legs of the itinerary is station \(w\) on line \(\ell\).

The boarding demand quantity \( BD_{\ell,u,t}(z) \) represents the total number of passengers who have managed to board a train at station \(u\) on line \(\ell\) by time period \(t\). It is computed as follows:

\[
BD_{\ell,u,t}(z) := \sum_{\{(w,v,r,i) \in \text{brdat}(\ell,u)\}} \sum_{t'=1}^{t} z_{t'}^{w,v,r,i},
\]

(3.9)

where the set \( \text{brdat}(\ell,u) \) represents all of the commute-route-legs that require boarding a vehicle at station \(u\) on line \(\ell\). This includes commutes where the origin station is \(u\) and the origin line is \(\ell\), as well as commutes where station \(u\) and line \(\ell\) are the location of some later transfer.

The function \( O_{\ell,u,t}(z) \) in (3.4b) computes the occupancy of a service as it passes through stop \(u\) on line \(\ell\), having started its run at time \(t\). It is computed as the following summation:

\[
O_{\ell,u,t}(z) := \sum_{\{(w,v,r,i) \in \text{passthru}(\ell,u)\}} z_{t}^{w,v,r,i},
\]

(3.10)

where the set \( \text{passthru}(\ell,u) \) represents all of the commute-route-legs that pass through station \(u\) on line \(\ell\). Specifically, when considering a commute \((w,v)\), route option \(r \in \text{routes}(w,v)\), and on the \(i\)th leg of the commute, \((w,v,r,i)\) must satisfy the criteria that (i) the \(i\)th element of \(\text{legs}(w,v,r)\) is \(\ell\), (ii) the transfer station connecting the \((i - 1)\)th and \(i\)th legs of the itinerary is \textit{at or before} station \(w\) on line \(\ell\), and (iii) the transfer station connecting the \(i\)th and \((i+1)\)th legs of the itinerary is \textit{after} station \(w\) on line \(\ell\). With this definition of \(O_{\ell,u,t}(z)\), constraint (3.4b) then ensures that the number of commuters on board a transit vehicle must be within the vehicle capacity \(K^\ell\).

Finally, the boarding and transfer constraints (3.4c) and (3.4d) enforce the requirement that commuters cannot embark on the \(i\)th leg of their commute until they have completed all previous legs.
Both $J(z, \theta)$ and $O_{t,u,l}(z)$ are linear functions of the decision variables $z$ in the case where the commuter choice probabilities $\theta$ are known. Therefore, equation (3.3) is a linear optimization problem when the commuter choice probabilities are known.

### 3.4 Optimization with Design-Dependent Choices

In this section, we examine a framework where addition to setting the schedule frequencies, the service operator can set prices $p = (p_{u,v,r})$, where $p_{u,v,r}$ represents the price charged for commute $(u,v)$ and route $r$. The indices for $p$ are intentionally detailed to accommodate the variety of pricing policies that could be implemented. Such policies can be described with the addition of the appropriate constraints and auxiliary variables, some examples of which are given below.

1. **Flat fare.** If the transit operator wants to charge a flat fare for all commuters entering the system, this can be accomplished by adding the constraint

   $$p_{u,v,r} = f$$  \hspace{1cm} (3.11)

   for all commutes $(u,v)$ and routes $r$, where $f$ represents the value of the flat fare.

2. **Line-based fare.** If the commuters are charged for each service they use, then auxiliary variables $f^\ell$ are introduced to represent the fare charged for each service. In addition, the constraints

   $$p_{u,v,r} = \sum_{\ell \in \text{legs}(u,v,r)} f^\ell$$  \hspace{1cm} (3.12)

   should be added for every commute $(u,v)$ and route $r$.

3. **Distance-based fare.** A more equitable pricing policy than the line-based pricing would be to weight the fare charged on each line by the distance that the commuter traveled along that line. Denoting that distance by the constant
\[ h^{u,v,r,\ell} \] for commute \((u, v)\), route \(r\), and line \(\ell\), the constraints

\[
p^{u,v,r} = \sum_{\ell \in \text{legs}(u,v,r)} h^{u,v,r,\ell} f^\ell \tag{3.13}
\]

then represent this distance-based pricing policy.

In a further extension to the framework in Section 3.3, we also allow that the route choices are not fixed, but can depend on the decisions made by the service operator. We write the choice probabilities as \(\theta(x, p)\) to show that they can depend on the schedule frequencies \(x\) and prices \(p\). The schedule frequencies and prices can then be set through the objective function

\[
\min_{x \in \mathbb{X}_B, p \geq 0} Q(x, \theta(x, p)). \tag{3.14}
\]

The choice probabilities \(\theta(x, p)\) are influenced by the utility that a commuter would gain from taking each route option, which we express as \(\mu^{u,v,r}_t(x, p)\) for commute \((u, v)\) and route choice \(r\) under schedule frequencies \(x\) and prices \(p\). To specify \(\mu^{u,v,r}_t(x, p)\), we assume that the crucial attributes in a commuter’s utility function are time, cost, and comfort. The time component can be described as time spent waiting to board a service, plus time spent on the service. The time spent waiting to board a service can often include not just time spent waiting for the next service arrival, but also any waiting time that is incurred because the service cannot fit any more commuters. We quantify comfort using the percent occupancy of the transit service, since that is the primary feature of commuter comfort that is endogenous to the model.

Concretely, assuming that services at time \(t\) on line \(\ell\) are arriving uniformly at rate \(x^\ell_t\), an arriving commuter should expect on average to wait \(\frac{1}{2x^\ell_t}\) for a service to arrive. This waiting time is incurred at every leg \(\ell \in \text{legs}(u,v,r)\). We represent the remaining time of the commute, which is spent on the service, with the term \(\Delta(u,v,r)\). For the cost attribute, we have previously defined \(p^{u,v,r}\) as the price charged to this commuter. Assuming that the commuter utility function is linear in the time, cost,
and comfort attributes, the form of \( \mu_{t}^{u,v,r}(x,p) \) is then given by:

\[
\mu_{t}^{u,v,r}(x,p) := -\beta_1 \sum_{\ell \in \text{legs}(u,v,r)} \left( \frac{1}{2x^\ell_t} + \Delta(u,v,r) \right) - \beta_2 p^{u,v,r} - \beta_3 \sum_{\ell \in \text{legs}(u,v,r)} \psi \left( \frac{\eta_{\ell,u,t}}{K^\ell x^\ell_t} \right),
\]

where \( \beta_1 \) represents marginal utility of time, \( \beta_2 \) represents the marginal utility of money, and \( \beta_3 \) represents the marginal utility of comfort. It is of course possible to weight the different time components of the utility function separately if commuters treat waiting time differently from in-vehicle time. The last term \( \sum_{\ell \in \text{legs}(u,v,r)} \psi \left( \frac{\eta_{\ell,u,t}}{K^\ell x^\ell_t} \right) \) represents our quantification of comfort from the percent occupancies of the relevant transit services, where \( K^\ell \leq K^\ell \) is a threshold “soft” capacity. The discomfort function \( \psi : \mathbb{R} \to \mathbb{R} \) largely follows the definition in [93], and has been modified to be convex and differentiable:

\[
\psi(\kappa) = \begin{cases} 
\kappa & \text{if } \kappa \leq 1 \\
\kappa \left( e^{\kappa} - 1 \right) & \text{if } \kappa > 1.
\end{cases}
\]

Intuitively, the discomfort increases linearly with the occupancy \( \eta_{\ell,u,t} \) up until the threshold \( K^\ell x^\ell_t \) and exponentially thereafter. This comfort term poses some particular challenges which we will address in Section 3.4.2.

### 3.4.1 Discrete Choice Models in Transit Assignment

With the routes and the associated utilities in hand, one possible choice model is the first-choice model [124], which is also called the pure characteristics model in the economics literature [17, 58] or the all-or-nothing model in the traffic literature. Under this model, the commuter chooses the route that provides her with highest utility, so that the choice probability is given by

\[
\theta_{t}^{u,v,r}(x,p) := \mathbb{1} \left( \mu_{t}^{u,v,r}(x,p) = \max_{r' \in \text{routes}(u,v)} \mu_{t}^{u,v,r'}(x,p) \right).
\]

However, this assumption that commuters behave exactly according to the specified utility function as a monolith is not entirely realistic, and is particularly unstable.
to problem parameters. We therefore turn to the multinomial logit model [125], where the setup is similar to that in the first-choice model, but the utility function $\mu_t^{u,v,r}(x,p)$ is also associated with a Gumbel-distributed noise parameter. Due to the noise parameter, commuters will probabilistically pick routes with the choice probabilities given by:

$$\theta_t^{u,v,r}(x,p) := \frac{\exp(\mu_t^{u,v,r}(x,p))}{\sum_{r' \in \text{routes}(u,v)} \exp(\mu_t^{u,v,r'}(x,p))}.$$  \hfill (3.18)

Finally, for purposes of comparison, we introduce the concept of the system optimum. In this model, we allow the choice probabilities $\theta$ to be any valid probability distribution over the route options. In this case, prices are irrelevant, and schedule frequencies are set according to the formulation

$$\min_{x \in \bar{X}, \theta} Q(x, \theta) \hfill (3.19a)$$

s.t. \[ \sum_{r \in \text{routes}(u,v)} \theta_t^{u,v,r} = 1 \quad \forall u, v = 1, \ldots, N \]

\[ \forall t = 1, \ldots, T \]  \hfill (3.19b)

$$\theta \geq 0. \hfill (3.19c)$$

The interpretation of (3.19) is that the system operator is able to direct passengers to take whatever route would benefit the system most. For example, if a particular route is especially congested, the system operator could direct passengers to alternative routes. Such a dictatorial policy would be impossible to implement, but the objective value of (3.19) serves as a useful lower bound for the best waiting time that any transit policy could achieve.

The multinomial logit model (3.18) suffers from a property known as the independence of irrelevant alternatives (IIA). An alternative choice model is the nested logit model, which is a generalization of the multinomial logit that has been shown by [171] to be consistent if the IIA assumption does hold. Following the example in [111], one approach is to classify all the route choices into a small set of “travel modes” based on
the combination of services used by that route. Under this model, commuters make route choices based on a two-level choice model: the first level decides the mode, and the second level decides the route options available under that mode. The resulting choice probabilities will then look like

$$\theta_{t,u,v,r}(x,p) := \frac{\exp(\alpha_1\mu_{t,u,v,m}(x,p))}{\sum_{m' \in \text{modes}(u,v)} \exp(\alpha_1\mu_{t,u,v,m'}(x,p))},$$

(3.20)

where $\text{modes}(u,v)$ is the set of travel modes available to go from $u$ to $v$, $\alpha_1$ is the coefficient of perceptional variation between different travel modes, and $\mu_{t,u,v,m}$ determines the utility for commuters to choose travel mode $m$ as

$$\mu_{t,u,v,m}(x,p) = \frac{1}{\alpha_2} \ln\left( \sum_{r \in \text{routes}(u,v;m)} \exp(\alpha_2\mu_{t,u,v,r}(x,p)) \right),$$

(3.21)

where $\text{routes}(u,v;m)$ is the set of routes under travel mode $m$, and $\alpha_2$ is the corresponding perception variation parameter. In practice, both $\alpha_1$ and $\alpha_2$ will have to be estimated from commuter survey data. The gradient $\nabla \theta_{t,u,v,r}(x,p)$ for both the multinomial logit (3.18) and the nested logit (3.20) can be easily derived using a recursive application of the chain rule, or computed using automatic differentiation [9].

### 3.4.2 Solution Algorithm

The transit frequency-setting problem with known choices (3.3) is a linear optimization problem, and can be solved efficiently. Similarly, the calculation of the system optimum (3.19) is a linear optimization problem. However, the transit frequency-setting and pricing problem (3.14), with the choice probabilities given according to the multinomial logit model (3.18) or the nested logit model (3.20), is a nonconvex optimization problem. To solve this problem, we use a first-order method which replaces the nonlinear function with a series of locally linear approximations. Around a particular point $(\bar{x}, \bar{p})$, the choice probability can be approximated as

$$\hat{\theta}_{t,u,v,r}(x,p; \bar{x}, \bar{p}) \approx \theta_{t,u,v,r}(\bar{x}, \bar{p}) + \nabla \theta_{t,u,v,r}(\bar{x}, \bar{p})' (x - \bar{x}, p - \bar{p}),$$

(3.22)
provided the point \((x, p)\) is close to the original point \((\bar{x}, \bar{p})\). Equation (3.22) is linear in the decision variables \(x\) and \(p\). Substituting the linearized choice probability approximations \(\hat{\theta}\) in (3.22) for the multinomial logit probabilities (3.18) in formulation (3.14) turns the problem into a linear optimization problem. Then, beginning with feasible starting schedule frequencies and prices \((x^{(0)}, p^{(0)})\), a new \((x^{(i)}, p^{(i)})\) can be produced for each iteration \(i = 1, \ldots\), by solving the problem

\[
\begin{align*}
\min_{x, p} & \quad Q(x, \hat{\theta}(x, p; x^{(i-1)}, p^{(i-1)})) \\
\text{s.t.} & \quad x \in \bar{X}_B \\
& \quad p \geq 0 \\
& \quad x^{(i-1)} - \eta \leq x \leq x^{(i-1)} + \eta \\
& \quad p^{(i-1)} - \gamma \leq p \leq p^{(i-1)} + \gamma,
\end{align*}
\]

where \(\eta\) and \(\gamma\) are constants referring to the step-sizes for schedule frequencies and prices, respectively. Formulation (3.23) solves the optimization problem with design-dependent choices when the frequencies \(x\) and prices \(p\) lie within the small intervals \([x^{(i-1)} - \eta, x^{(i-1)} + \eta]\) and \([p^{(i-1)} - \gamma, p^{(i-1)} + \gamma]\), in which the multinomial logit choice probabilities can be linearly approximated. In this way, we solve the nonconvex optimization problem using a series of linear optimization problems. The constants \(\eta\) and \(\gamma\) should be chosen to be small enough that the approximation (3.22) is reasonably accurate within the frequency and price intervals, but large enough that progress can be made quickly. Advantages of such first-order methods include their speed and simplicity of implementation. However, it is possible in nonconvex problems such as ours to converge to suboptimal local extrema. To guard against this, we repeat our procedure at multiple random starting points and select the frequencies and prices producing the best objective value. For more on first-order methods, see [35].

As previously mentioned, the term \(\eta_{t,u,t}\) in the utility function (3.15) poses a particular challenge. Taking \(\eta_{t,u,t}\) to be the occupancy expression \(O_{t,u,t}(z)\) is the most natural modeling choice, but would make the utility functions extremely dense

69
functions of the model decision variables. Instead, we specify $\eta_{\ell,u,t}$ at iteration $i$ to be the value of the occupancy calculated from step $i - 1$. Since the region in constraints (3.23d) and (3.23e) is small, the previous values are good estimates of the values in the current iteration.

To summarize, our model optimizes frequencies and prices in order to minimize waiting time due to congestion. Commuters respond to the frequencies and prices with choice probabilities that are determined by the utility they might derive from each route option. These choice probabilities are also decision variables in the optimization formulation, allowing commuters to respond dynamically to the decisions made by the transit operator. The full formulation is a nonconvex optimization model, but is solved efficiently using first-order methods.

### 3.5 Computational Experiments

We now turn to computational experiments on two real-world transit networks for which we obtained data. The first is a subset of the train network in Kanagawa prefecture, which is part of the metropolitan area surrounding Tokyo, Japan. The second is the subway and bus systems in Boston, Massachusetts, which are run by the Massachusetts Bay Transportation Authority (MBTA). These two transit systems are of particular interest due to their high utilization by a booming urban population, as well as the multiplicity of route choices available to commuters.

All methods were implemented using the optimization package JuMP [114] in the Julia programming language [25], and solved using Gurobi 7.5 [82]. Computational experiments for the Kanagawa network were run on eight cores of a computer with a 16-core Intel Xeon E5-2650 CPU, 3.40 GHz processor, and 64GB of memory. Computational experiments for the Boston network were scheduled as jobs on a cluster: each one, corresponding to multiple random starting points, used one core with 2GB of memory and took less than 30 minutes to complete on Dell C6300 machines with Intel E5-2690 v4 2.6 GHz/35M Cache processors.
Before discussing our computational experiments in detail, we note that the purpose of the experiments is not to provide a policy prescription, but to show that our methods can be applied to realistic data. Due to the limited data we had available, we use the multinomial logit choice model (3.18), and describe our method for generating utility function coefficients in Appendix B.1. We do not claim that parameters such as relative costs, capacities, or coefficients in the utility function, are accurate, merely that they are reasonable enough to draw insight.

3.5.1 Case Study: Tokyo

Through [61], [137], and [47], we obtained transit demand data for a subset of the train system in Kanagawa Prefecture, which is part of the Greater Tokyo Area in Japan. The transit network, displayed in Figure 3-2, is comprised of 57 stations along five lines: Blue, Tokaido, Negishi, Odakyu, and Yokohama. The demand data includes the precise time, origin, and destination of each commute for a typical evening, from 5-9pm. We divided time periods to be of length 15 minutes. Travel times along the network were obtained through queries to Google Maps. Our method for generating utility function coefficients is described in Appendix B.1. Route options were precomputed for the network using shortest-path-based methods. In general, it appeared that route options after the second-shortest path tended to involve much higher in-vehicle times and numbers of transfers, so these were discarded.

As shown in Figure 3-2, the Blue and Negishi (cyan) lines are local trains with closely-spaced stops, while the Tokaido (orange) line is an express line that connects some of the stations on the Blue and Negishi lines. Some example commutes and associated travel times are shown in Table 3.2, illustrating that the Tokaido line is significantly faster than both the Blue and Negishi lines, and the Blue line is slightly faster than the Negishi line.

In this case study, we first focus on an illustrative subset of our Kanagawa data that comprises the Blue, Negishi, and Tokaido lines during the rush hour from 5 to 6pm. This allows us to examine the decision variables in great detail and interpret how transit dynamics relate to the optimal frequencies and prices. We then move to
Figure 3-2: A map of the Kanagawa trains network.

Table 3.2: Travel times for various origin-destination pairs on the Kanagawa network

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Line</th>
<th>Travel Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ofuna</td>
<td>Yokohama</td>
<td>Tokaido</td>
<td>16 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Negishi</td>
<td>34 min</td>
</tr>
<tr>
<td>Totsuka</td>
<td>Yokohama</td>
<td>Tokaido</td>
<td>10 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blue</td>
<td>27 min</td>
</tr>
<tr>
<td>Kannai</td>
<td>Yokohama</td>
<td>Blue</td>
<td>5 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Negishi</td>
<td>7 min</td>
</tr>
</tbody>
</table>

the full network of all five lines for the full 5 to 9pm time range in our dataset to show tractability at scale. Our findings are outlined as follows:

- In Section 3.5.1, we examine optimal frequency-setting on a subset of the Kanagawa network and show that it can have a significant impact on reducing total waiting time. We also show that the impact of frequency-setting on reducing waiting times becomes more pronounced as commuters’ sensitivity to congestion increases.

- In Section 3.5.1, we also examine optimal pricing on a subset of the Kanagawa network and show that pricing is more effective in reducing waiting times when commuters are sensitive to congestion. Nonetheless, it is less effective than frequency-setting in reducing total waiting time.
In Section 3.5.1, we perform coordinated frequency-setting and pricing and demonstrate that our methods are practically useful. We compared system performance for the full Kanagawa network under a variety of pricing policies and evaluate their social impact, and show that our methods are tractable for a city-size network with quality solutions obtained within minutes. We also show that coordinated frequency-setting and pricing significantly outperforms frequency-setting alone.

The effect of congestion on a small network

In this section, we focus on a subnetwork comprising the Blue, Negishi, and Tokaido lines, where the dynamics are relatively simple. Of the 70,599 commuters on this subnetwork, 24,226 (34.3%) have multiple commuting options available to them. We refer to them as multiple-option commuters. The remaining 46,373 (65.7%) passengers only have one route option available to them, and we refer to them as single-option commuters. Among single-option commuters, the Blue line is the most highly-sought after line (22,206 passengers), followed by the Negishi line (15,162 passengers), and finally by the Tokaido line (14,021 passengers). Note that these numbers do not add up to the 46,373 passenger total because some passengers must transfer between multiple lines to get from their origin to their destination. Although single-option commuters do not have preferences to be modeled, their presence on the trains impacts the congestion of each line and the decisions that are made to alleviate this congestion. The parameter $\bar{K}_\ell$ in the utility function (3.15) was set to 0.8$K$, meaning that commuters begin to experience heightened disutility once the occupancy of the vehicle grows past 80% its total capacity.

In our first experiment, we examine the impact of commuters’ sensitivity to congestion on optimal frequency-setting without pricing. For a range of budgets and $\beta_3$ parameters, we ran our solution algorithm from thirty initial starting points where the frequencies were set randomly and the prices were set to be identical for each line. Each instance terminated in under a minute, with many as quickly as several seconds. The optimal waiting times averaged across the total commuter demand are
reported in the third column of Table 3.3, and the next three columns break down the
waiting time experienced across the three different lines. As commuters become more
sensitive to discomfort due to congestion, the average waiting time per commuter
drops. This drop is driven largely by the commuters on the Blue line, which is the
most highly sought-after line.

Table 3.3: Frequency-setting on a small network under varying budgets and commuter
sensitivities to congestion.

<table>
<thead>
<tr>
<th>Budget (trains)</th>
<th>$\beta_3$</th>
<th>Waiting Time / Commuter (min)</th>
<th>Commuters / Train</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall</td>
<td>Blue</td>
<td>Tokaido</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>5.84</td>
<td>2.66</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5.79</td>
<td>2.53</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5.76</td>
<td>2.39</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>3.65</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3.34</td>
<td>1.60</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2.04</td>
<td>1.28</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>2.04</td>
<td>0.77</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1.87</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.52</td>
<td>0.49</td>
</tr>
</tbody>
</table>

This phenomenon illustrates that optimal frequency-setting is more effective when
the commuters are more sensitive to congestion. Intuitively, as service frequency in-
creases on a line, all else being equal, more commuters will opt to use that service.
A line that improves its service frequency must then be prepared to serve not only
its existing commuters, but also the others who switch to its improved service. How-
ever, this effect is controlled by the commuters’ disutility for congestion, since as the
vehicles become more crowded, commuters will ultimately choose alternatives that
are less crowded even if they are slower or less frequent. This effect is seen in the
final three columns of Table 3.3, which shows that as the commuters become more
sensitive to congestion, relatively fewer commuters choose to take the Blue line and
instead move to the Tokaido or Negishi lines.

In our next experiment, we illustrate the impact of commuters’ sensitivity to
congestion on pricing. The setup was identical as before, except that frequencies
were fixed, roughly in proportion to the demand for each line, and pricing was the
Table 3.4: Price-setting on a small network under varying budgets and commuter sensitivities to congestion

<table>
<thead>
<tr>
<th>Budget (ntrains)</th>
<th>$\beta_3$</th>
<th>Wait/Com. (min)</th>
<th>Pricing Premium</th>
<th>Blue</th>
<th>Negishi</th>
<th>Tokaido</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>6.38</td>
<td>2.05</td>
<td>0.00</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>6.37</td>
<td>1.77</td>
<td>0.00</td>
<td>1.20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6.37</td>
<td>1.44</td>
<td>0.00</td>
<td>1.15</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>4.86</td>
<td>1.00</td>
<td>0.00</td>
<td>1.50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4.86</td>
<td>1.05</td>
<td>0.00</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4.85</td>
<td>1.05</td>
<td>0.00</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>0</td>
<td>3.83</td>
<td>1.00</td>
<td>0.00</td>
<td>0.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3.82</td>
<td>1.05</td>
<td>0.00</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.81</td>
<td>1.05</td>
<td>0.00</td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>

only lever to manage congestion. The optimal waiting times averaged across the total commuter demand are reported in the third column of Table 3.4, and the next three columns show the pricing premium set on each line relative to the Negishi line, which is consistently the cheapest line. As in the frequency-setting case before, the waiting time declines as commuters become more sensitive to congestion although the effect is substantially less pronounced than before. Furthermore, for the lower budgets $B = 30$ and $B = 60$, the prices generally decline with increased $\beta_3$, an effect that is seen most clearly at the lowest budget. This illustrates that when commuters are sensitive to congestion, their natural tendency to avoid congestion allows the transit operator to achieve lower waiting times at lower prices.

**Practical Policy Evaluation on the Full Network**

Returning to the full Kanagawa network and the full 5-9pm time range, we now demonstrate the tractability of our methods for even larger network sizes and longer time scales. Of the 362,477 commuters on this subnetwork, 183,681 (50.7%) are multiple-option commuters. Table 3.5 shows the median run time for each budget, over 30 different random starting positions ($x^{(0)}, p^{(0)}$). Unsurprisingly, frequency-setting alone appears to be the easiest problem; however, even for distance- and line-pricing, typical run times are within several minutes.
Table 3.5: Median run time over thirty iterations for various budgets.

<table>
<thead>
<tr>
<th>Budget of Trains</th>
<th>Median Run Time (s)</th>
<th>Scheduling</th>
<th>Scheduling, Line Pricing</th>
<th>Scheduling, Dist. Pricing</th>
</tr>
</thead>
<tbody>
<tr>
<td>280</td>
<td>255</td>
<td>546</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>240</td>
<td>503</td>
<td>289</td>
<td></td>
</tr>
<tr>
<td>320</td>
<td>220</td>
<td>601</td>
<td>344</td>
<td></td>
</tr>
<tr>
<td>340</td>
<td>224</td>
<td>585</td>
<td>277</td>
<td></td>
</tr>
<tr>
<td>360</td>
<td>219</td>
<td>532</td>
<td>468</td>
<td></td>
</tr>
<tr>
<td>380</td>
<td>225</td>
<td>497</td>
<td>380</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>214</td>
<td>221</td>
<td>123</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-3 shows the progress of the objective function over time for a selection of budgets $B$ when optimizing for both schedule frequencies and prices jointly, under a distance-based pricing policy. Each line on the plot corresponds to one of the 30 different random starting positions. Some of the runs converge to suboptimal extrema, clearly visible in the low-budget $B = 280$ case. However, in the majority of cases, high-quality solutions are obtained within approximately 200 seconds, and the remainder of the time is spent on small refinements with incremental improvement to the objective. This indicates that quality solutions should be obtainable with a relatively modest number of random starts, and significant time can be saved by terminating immediately after the objective plateaus.

Figure 3-3: Optimization progress for multiple different random starting points and two example budgets, on the full Kanagawa network and using frequency-setting and line pricing. Each line corresponds to the objective function value from a different random starting point.
The objective value for various budget levels $B$ and pricing policies are shown in Figure 3-4. For comparison, the distance-pricing (cf. equation (3.13)) policy and line-pricing (cf. equation (3.12)) are displayed with a pure frequency-setting policy and the system optimum (cf. equation (3.19)), which are the respective upper and lower bounds on optimal policy performance. There appears to be significant benefit to implement any sort of optimal pricing policy: the distance-pricing policy is able to reduce the waiting time per person by approximately a minute, and the line-pricing policy is able to lower this further by another minute, for a total improvement of approximately two minutes per commuter. Particularly at higher budgets $B \geq 360$, the pricing policies markedly close the gap between frequency-setting without pricing and the system optimum.

![Figure 3-4: Optimal objective values under frequency-setting without pricing, co-ordinated frequency-setting and distance pricing, coordinated frequency-setting and line pricing, and the system optimum. The network of study was the full Kanagawa network.](image)

Beyond pure system efficiency, a transit operator may be interested in equity as an auxiliary objective. For a low budget $B = 280$, we examine the effects that the optimal decisions will have on the commuters who take public transit. Figure 3-5 displays the distribution of the utility values amongst commuters under the distance- and line-pricing policies. The line-pricing policy shows a slightly heavier left tail than the distance-pricing policy, which reflects the higher prices paid under line-pricing.
Among the commuters for whom there are differences in the fare policies (i.e., those who take some line other than the Negishi or Odakyu lines), the proportion of people who would prefer line-pricing over distance-pricing is shown in Figure 3-6. In general, we observe the intuitive result that the longer the distance traveled, the more likely a commuter is to prefer line-pricing, since under such a policy, their fare will not increase with the trip length. Such a result could be useful for transit authorities in deciding which pricing policy to impose: for transit networks where distance traveled correlates positively with income (for example, if lower-income communities move farther away from city centers), the more efficient line-pricing policy would not put an unfair burden on lower-income commuters.

3.5.2 Case Study: Boston

We now turn to a transit setting in Boston, Massachusetts. The motivation of this second study is to show applicability for different transit networks and to contrast with the Kanagawa results. In Boston, there are fewer route choices available than in Kanagawa; the potential gains are therefore smaller, but still substantial. Transit
Figure 3-6: Proportion of Kanagawa commuters who would choose line-pricing instead of distance-pricing, by travel distance.

in the greater Boston area consists of both buses and the subway, run by the MBTA. Buses are typically numbered, and the subway system consists of five intersecting lines called the Red, Orange, Green, Blue, and Silver lines. With this network, we move beyond networks with trains as the single mode of transit, to a fully multi-modal setting.

The MBTA provided us with private data on the number of entries and exits hourly at every station, for both the buses and the subway system. We ran a generative model based on the approach in [24] to produce peak-hour origin-destination demand matrices on weekdays. Again, we focus on the rush-hour from 5-6pm when the system is at capacity, where there is the potential for pricing mechanisms to influence commuting behavior and reduce network congestion.

We focus on the core of the metropolitan area, which accounts for the main volume of commuters and where multiple alternatives in route choices exist. To this end, we include bus service 1, as well as the Red, Orange, and Green subway services. This corresponds to a transit network that comprises 74 stops along 4 services. A map showing the core component of our subnetwork is shown in Figure 3-7. Since the subway and bus lines in Boston interact across a smaller area than in the Kanagawa network, there are fewer multiple-option commuters. In order to model more route choices for the commuters, we assumed that certain stations that were within a 0.5 mile walk of each other were substitutable for each other. For example, much of
the Green and Orange lines lie close together, so commuters might have the option of taking either line. These substitutable stations are shown in dotted boxes in Figure 3-7. After considering these substitutable stations, we found that of the 95,760 commuters in this network during the 5 to 6pm rush hour period, 28,020 (29.2%) of them are multiple-option commuters, and of the 362 origin-destination pairs with nonzero demand, 100 (27.6%) of them are multi-option commutes.

![Figure 3-7: A subset of the MBTA network used in Section 3.5.2, showing parts of the Red, Green, Orange, and 1 bus (gray) lines. Stations further outside of the metropolitan core of the network, and some 1 bus stations, are omitted for clarity. To generate more route options for commuters, some stations that were within a 0.5 mile walk of each other were assumed to be substitutable for each other. Stations that were viewed as substitutable are surrounded with dotted boxes.

The optimization formulation (3.3) largely treats buses and trains the same; the main difference is that the capacity $K_\ell$ tends to be smaller for buses, and that buses tend to be operated at lower costs. In the computational experiments that follow, we determine pricing and frequency-setting for both trains and buses jointly, assuming that buses operate at 30% of the capacity and at 30% of the cost of trains.

As before, we consider the pricing policies corresponding to line-based fare (3.12) and distance-based fare (3.13), and compared them to a frequency-setting policy with a fixed flat fare. Again, all policies were compared to the lower bound of the
system optimum (3.19). For each model, we ran the first-order method (3.23) with 30 different random starting points and chose the schedule frequencies and prices with the lowest waiting time. To compare the models with different operating budgets, we ran it across a range of budgets from the equivalent of 45 to 60 trains, and present the results in Table 3.6.

Comparing the performance of the models, there is a consistent gap between the frequency-setting model alone and the system optimum. As was the case with the Kanagawa network, both line pricing and distance pricing improve upon the model with only frequency-setting, although this time, the line-pricing and distance-policies are comparable in performance. The gains for joint frequency-setting and pricing are less substantial than they were in the case of the Kanagawa network, which is likely due to the fact that there are both relatively fewer route options and fewer multi-option commuters in Boston. Nevertheless, the addition of distance pricing decreases congestion by 5.5% in the case of $B = 55$, showing that coordinated frequency-setting and pricing can provide gains even on simpler networks.

Table 3.6: Optimal objective values under the system optimum, frequency-setting without pricing, coordinated frequency-setting and line pricing, and coordinated frequency-setting and distance pricing. The network of study was the Boston network.

<table>
<thead>
<tr>
<th>Budget (ntrains)</th>
<th>Waiting Time / Commuter (min)</th>
<th>System Optimum</th>
<th>Frequencies</th>
<th>Frequencies, Line Prices</th>
<th>Frequencies, Distance Prices</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>7.49</td>
<td>9.86</td>
<td>9.48</td>
<td>9.54</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>4.24</td>
<td>6.08</td>
<td>5.80</td>
<td>5.79</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>2.28</td>
<td>3.85</td>
<td>3.67</td>
<td>3.64</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0.82</td>
<td>2.00</td>
<td>1.73</td>
<td>1.82</td>
<td></td>
</tr>
</tbody>
</table>

### 3.6 Conclusion

In this paper, we have presented a general framework for jointly optimizing schedule frequencies and prices on multi-modal transit networks. With modern developments in transit services, it has grown of increasing importance to other cities to use whatever
means available to them to operate under tight constraints. Our framework accounts for the feedback loop that occurs when passengers respond to frequency-setting and pricing decisions made by the transit operator through the multinomial logit model. The passenger choices induce demand for transit services, and the transit operator seeks to efficiently service this demand by minimizing congestion in the transit system.

We solve our formulation using first-order methods, and demonstrate its tractability on two real-world transit networks in Tokyo, Japan and Boston, Massachusetts. To our knowledge, ours is the first paper that addresses joint frequency-setting and pricing optimization for public transit networks and at scale. As the computational experiments are performed on a mixture of synthetic and real data, it is difficult to make direct comparisons with the current system. Nonetheless, we are able to evaluate the performance of multiple frequency-setting and pricing policies against a lower bound given by the system optimum, illustrating the potential benefits that joint frequency-setting and pricing can have in inducing passengers to make decisions that benefit the system as a whole.
Chapter 4

Transit Network Design at Scale

Public transit remains the most efficient way to service a densely-packed commuter population. However, reliability issues and increasing competition in the transportation space have led to declining ridership, and transit agencies must also operate under tight budget constraints. Recent attempts at using bus network re-design to improve ridership have attracted attention from transit authorities around the country. However, the analysis seems to rely on ad-hoc methods, for example, considering each bus line in isolation and using manual incremental adjustments with backtracking. We provide a holistic approach to optimally design a transit network using column-and-constraint generation. Our approach is tractable, scaling to hundreds of stops, and we demonstrate its usefulness on a case study with real data from Boston.

4.1 Introduction

The United Nations [164] projects that urban populations will increase from 54 percent of the global population in 2014 to 66 percent in 2050. Together with overall population growth, this represents an increase of 2.5 billion people. With more of the world’s population living in cities, it is increasingly important to provide transit systems that can efficiently serve a densely-settled population.

Public transit is crucial for sustainability, efficiency, and equity in serving urban populations. However, it faces significant outside competition from ride-sharing com-
panies and private bus or shuttle services [33]. Many cities such as Philadelphia [104], Los Angeles [130], Washington D.C. [152] are seeing declining bus ridership, prompting transit authorities to consider what can be done to halt this decline. A recent bus network re-design in Houston led to a 6.8% increase in ridership across the bus and light rail networks [27], inspiring other cities to also consider re-designing their bus networks. Examples include Philadelphia [103], Boston [166], St. Louis [149], and Edmonton [157].

The problem of designing a set of sequences of stops, called *lines* or *routes*, in order to service a commuting population is called the Transit Network Design Problem (TNDP). The TNDP is a challenging combinatorial optimization problem and has been well-studied in the literature, which we survey in Section 4.2. Despite this body of work, advanced techniques are not used to a significant extent in the planning process for actual network design. For instance, the Service Delivery Policy [121] outlined by the Massachusetts Bay Transportation Authority (MBTA) analyzes the transit network by evaluating bus lines individually without considering the network as a whole, and improvements to bus networks were performed as incremental adjustments to individual bus lines. Such a strategy clearly limits the scope of a potential bus network re-design.

One of the main barriers to leveraging advanced techniques is scalability; many algorithms have not been proven on the scale that real transit networks require, which can be up to hundreds or thousands of stops. Due to the combinatorial explosion in the number of possible service routes, most work in this area relies on the use of metaheuristics that do not provide optimality guarantees. Our work makes the following contributions:

1. We present a model that addresses the issues of interest to transit authorities, which are principally *ridership*, *connectivity*, and *budget*. Ridership represents serviced demand, and is also the main driver of revenue to a transit agency. Connectivity ensures that commuters can go from origin to destination in a relatively direct manner. Budget recognizes the labor and financial constraints that agencies operate under.
2. We address two crucial interests from the commuter perspective: number of
transfers, and travel time. If either of them are not adequately serviced, then
transit ridership will decrease. Therefore, properly accounting for them is key
to generate practical transit network designs that addresses commuters’ needs.

3. We demonstrate our model’s scalability using real data from Boston, which has
a network of hundreds of stops. We use a model of commuter routing behavior
that allows for fast precomputation outside of the optimization problem. Using
a column-and-constraint generation algorithm, we allow our models to scale
while providing optimality guarantees. Our algorithm converges to the optimal
solution and produces high quality solutions within a reasonable amount of
time.

The rest of the paper is outlined as follows. In Section 4.2, we provide an overview
of the literature on the TNDP. In Section 4.3, we describe our model and algorithmic
approach. In Section 4.4, we show computational results on a variety of case studies
involving both synthetic and real data. Finally, we offer concluding remarks in Section
4.5.

4.2 Literature Review

The TNDP is well-studied; for a review of material until the 1980s, see [115], and for
more recent reviews, see [80] and [64]. The goal of the TNDP is to design a set of
routes for buses or trains to serve transit demand in a cost-effective way. Auxiliary
objectives important to transit operators such as service area coverage may also be
incorporated. Most methods iteratively generate potential lines before selecting a
final set of routes to operate.

Much of the early work on the TNDP focused on heuristic solution methods. Typi-
cally, the origin-destination demand matrix was sorted from highest to lowest demand,
and bus routes were generated using fast shortest-path computations between high-
demand nodes. [117] generate an initial line set by computing the shortest paths
between terminal nodes, and then uses local search to iteratively improve the total travel time on the network. [43] and [4] extended this work by including additional routes that are no longer than a factor of the length of the shortest paths. In addition, [4] considered criteria for local node insertions to further expand the generated lines. Along a similar vein, metaheuristics such as genetic algorithms [46, 168], simulated annealing [183], and tabu search [112] have also been used to iteratively improve upon initial heuristically-generated route sets. [182] introduced the notion of route-directness and network-directness constraints to capture geometric characteristics of desirable routes, and used a hill-climb search algorithm to solve it. Finally, [178] used ant-colony optimization and considered direct demands and single transfers in route design.

Another area of work has employed mathematical optimization to solve network design problems. The benefit of mathematical optimization is a certificate of optimality; however, many models have had scalability issues at practical network sizes. Many papers restricted their attention to the optimal selection of a subset of bus stops [128] or heuristically-generated bus routes [79, 39], without considering the generation of new routes. Even with these limitations, [79] scaled to a network of only 49 stops, which was preprocessed to reduce the size to nine stops. [39] additionally considered the passenger perspective through a lower-level assignment model, and scaled to a larger network of 84 nodes and 143 edges.

Relatively fewer papers have addressed exact route generation due to further scalability issues. [169] used mixed-integer optimization to create a fixed number of routes between given bus stops that would minimize operating costs subject to capacity constraints. However, the formulation is not practical, scaling only to a network of ten stops. [8] used constraint programming to define service level goals and budget limits, but it was computationally difficult to find a solution even for a small fifteen-stop case. Rather than relying purely on branch-and-bound, [119] used a variant of Benders decomposition to solve a network design problem on 24 stops and 264 edges.

In contrast to these smaller-scale examples, [31] employed column generation to scale up their model to a network of hundreds of stops and one thousand edges, a
truly large-scale application. Their model sought to select a set of lines to operate and set train frequencies that minimizes a combination of total operating costs and travel time. However, they remained closely tethered to the original network design by only considering edges that already existed in the network in their computational study, so that the new lines that were produced were rearrangements of existing lines.

We propose an approach that scales up to a network of hundreds of nodes and thousands of edges. It accommodates a more flexible arrangement where all potential edges, not just the existing ones, may be incorporated into the new design in order to generate new bus lines. Unlike [31] which ignored transfers and allowed for unlimited travel times in their model, we explicitly model both direct and indirect passenger routes and restrict commuters to reasonable travel times.

4.3 Methods

We consider the problem of designing a transit network, and provide a description of the problem parameters in Table 4.1.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>budget available for network design</td>
</tr>
<tr>
<td>$c_\ell$</td>
<td>cost to deploy a transit service along line $\ell$</td>
</tr>
<tr>
<td>$\delta_{u,v}$</td>
<td>direct travel time between stops $u$ and $v$</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td>set of all transit lines</td>
</tr>
<tr>
<td>$N$</td>
<td>number of transit stops</td>
</tr>
<tr>
<td>$\text{stops}(\ell)$</td>
<td>the set of all stops on line $\ell$</td>
</tr>
<tr>
<td>$\mathcal{T}_{u,v}$</td>
<td>the set of all viable transfer stops between stops $u$ and $v$.</td>
</tr>
</tbody>
</table>

Each bus is scheduled to travel along a fixed sequence of stops, which we call a line. In practice, bus lines are typically bidirectional, so we take each bus line to actually represent two bus lines traveling in opposite directions between the terminal stops. Each origin-destination pair $(u, v)$ is called a commute. Each commute may be associated with a number of different route options, each of which is a different
sequence of stops that can take a commuter from origin \( u \) to destination \( v \). We further assume that the cost \( c_\ell \) of a bus line can be computed by summing the travel times \( \delta_{u,v} \) between pairs of consecutive stops \( u \) and \( v \) on the line.

We use boldface letters to denote vectors, and generalized inequalities on them to be element-wise. \( \mathbf{x}^T \) denotes the transpose of \( \mathbf{x} \), \( \mathbf{e} \) is the vector of all ones, and and \( |\mathcal{S}| \) refers to the cardinality of the set \( \mathcal{S} \).

**4.3.1 Serving direct passengers**

For simplicity, we first consider a network where commuters only take the bus if they can get from origin directly to destination without transfers. We assume that there is a set \( \mathcal{L} \) of all feasible bus lines, and introduce the following variables:

- \( x_\ell \): 1 if line \( \ell \in \mathcal{L} \) is operated, 0 otherwise, and
- \( \theta_{u,v} \): 1 if \((u, v)\) commuters take public transit, 0 otherwise.

To design a network that has high ridership, we solve the following integer optimization problem:

\[
\text{TNDP}(\mathcal{L}) = \max_{\mathbf{x}, \theta} \sum_{u=1}^{N} \sum_{v=1}^{N} d_{u,v} \theta_{u,v} \quad (4.1a)
\]

subject to:

\[
\sum_{\{\ell \in \mathcal{L} | u \in \text{stops}(\ell), v \in \text{stops}(\ell)\}} x_\ell \quad \forall u = 1, \ldots, N, v = 1, \ldots, N, \quad (4.1b)
\]

\[
\mathbf{c}^T \mathbf{x} \leq B \quad (4.1c)
\]

\[
\theta \leq \mathbf{e} \quad (4.1d)
\]

\[
\mathbf{x} \in \{0, 1\}^{\mathcal{L}}. \quad (4.1e)
\]

The objective (4.1a) maximizes the total demand that is served by the transit network. Constraint (4.1b) requires that some line that connects both \( u \) and \( v \) should be operated in order for those commuters to be serviced. Constraint (4.1c) is the budget constraint on the lines to be operated. Our model focuses on the operator
perspective and seeks to maximize ridership, which is the main metric that many transit authorities use to measure their operating performance [27]. In maximizing ridership, we allow for the case that a transit network is unable to service all demand. This is important because when budgets are tight, or when a transit agency is considering expansion into new areas, the requirement that all demand is served may be onerous. There are other operating constraints such as capacity constraints that a transit operator must consider. However, these can be addressed in the purview of scheduling or frequency-setting, so we do not consider them here.

Problem (4.1) is in general difficult to solve, as the set $\mathcal{L}$ of all feasible bus lines will be extremely large. However, it can be solved efficiently using column generation. For a comprehensive overview of column generation, see [7]. We begin with the following relaxation:

$$\text{MP}(\mathcal{L}) = \max_{x, \theta} \sum_{u=1}^{N} \sum_{v=1}^{N} d_{u,v}^{u,v} \theta_{u,v}$$

s.t. $\theta_{u,v} \leq \sum_{\{\ell \in \mathcal{L} | u \in \text{stops}(\ell), v \in \text{stops}(\ell)\}} x_{\ell} \quad \forall u = 1, \ldots, N, \quad v = 1, \ldots, N,$

$$c^T x \leq B$$

$$\theta \leq e$$

$$x \geq 0,$$

which we call $\text{MP}(\mathcal{L})$ to stand for the master problem. As the constraint $x \leq e$ is omitted, this is not the linear relaxation of Problem (4.1). (Removing the upper bound on $x$ reduces redundancy in the model, which helps in calculating reduced costs for subsequent stages of the column generation.) But it is straightforward to generate a solution to the linear relaxation of Problem (4.1) from the solutions to Problem (4.2): for any solution $(\bar{x}, \bar{\theta})$ to Problem (4.2), the solution $(x', \bar{\theta}) = ((\min(\bar{x}_{\ell}, 1))_{\ell \in \mathcal{L}}, \bar{\theta})$ will remain feasible and have the same objective value as $(\bar{x}, \bar{\theta})$, while also satisfying $x' \leq e$. 

89
The dual of Problem (4.2) is as follows:

\[
D(\mathcal{L}) = \min_{p,q,s} \quad Bq + e^\top s \tag{4.3a}
\]

subject to

\[
- \sum_{u \in \text{stops}(\ell)} \sum_{v \in \text{stops}(\ell)} p_{u,v} + c_{\ell} q \geq 0 \quad \forall \ell \in \mathcal{L}, \tag{4.3b}
\]

\[
p_{u,v} + s_{u,v} = d_{u,v} \quad \forall u = 1, \ldots, N, v = 1, \ldots, N, \tag{4.3c}
\]

\[
p, q, s \geq 0. \tag{4.3d}
\]

Evidently, by solving the dual (4.3), we also solve the primal (4.2).

Our column generation algorithm proceeds iteratively as follows. First, we begin with a restricted set of bus lines \( \bar{\mathcal{L}} \subseteq \mathcal{L} \) and the corresponding subset of variables \( (x_\ell)_{\ell \in \bar{\mathcal{L}}} \). We solve the restricted master problem, which is the primal MP(\( \bar{\mathcal{L}} \)) (see (4.2)), on this restricted set to get a primal solution \( \bar{x} \) and corresponding dual solution \( (\bar{p}, \bar{q}, \bar{s}) \). To see if the solution \( \bar{x} \) is optimal for the full problem, we check whether there exists some line \( \ell \in \mathcal{L} \setminus \bar{\mathcal{L}} \) that violates the constraint (4.3b), i.e.

\[
c_{\ell} \bar{q} < \sum_{u \in \text{stops}(\ell)} \sum_{v \in \text{stops}(\ell)} \bar{p}_{u,v}. \tag{4.4}
\]

The right-hand-side of the violated constraint (4.4) is interpreted as the increase in the primal ridership objective due to servicing commutes \( (u, v) \) on line \( \ell \). For example, a dual variable \( p_{u,v} \) might represent increased ridership from commutes \( (u, v) \) that are not already served by the network, either because no line exists connecting stops \( u \) and \( v \), or because any connecting lines were not included in the network due to the budget constraint. The left-hand-side of the violated constraint (4.4) contains dual variable \( q \), which represents the increase in ridership associated with a unit increase in budget, and is in the appropriate units converting budget to ridership. Condition (4.4) therefore requires that, for a new line to be profitable, the ridership increase should outweigh the associated costs of the line. For example, if the budget constraint
is not tight, then $\bar{q} = 0$ by complementary slackness, and the algorithm searches for any lines with positive ridership.

New profitable lines $\ell$ can be generated by solving the subproblem, which is formulated as an integer optimization problem. When only considering the cost of connecting adjacent stops, the subproblem is a shortest path problem. However, the problem is more complicated with the additional ridership computation. We define a graph $G(\mathcal{V}, \mathcal{E})$ composed of nodes $\mathcal{V} = \{0, \ldots, N, N + 1\}$, where node 0 corresponds to a source, nodes 1, $\ldots$, $N$ correspond to the stops on the transit network, and node $N + 1$ corresponds to a sink. The edge set $\mathcal{E}$ consists of the following types of directed edges:

- edges from the source (0) to all stop nodes (1, $\ldots$, $N$),
- edges between the stop nodes, and
- edges from the stop nodes (1, $\ldots$, $N$) to the sink $N + 1$.

The subproblem uses the following decision variables:

- $f_{i,j}$: 1 if edge $(i, j) \in \mathcal{E}$ is used, 0 otherwise, and
- $g_{u,v}$: 1 if commute $(u, v)$ can be served, 0 otherwise.

A bus line corresponds to a simple path from source to sink in $G(\mathcal{V}, \mathcal{E})$. In practice, bus lines are typically bidirectional, so we take each bus line to actually represent two bus lines traveling in opposite directions between the terminal stops. Therefore, a bus line can serve any commuters whose origin and destination are both on the path from source to sink regardless of ordering, and the relationship between the $f$ and $g$ variables is enforced by the following constraints:

$$g_{u,v} \leq \sum_{v' \in \text{In}(u)} f_{v',u} \quad \forall u = 1, \ldots, N, v = 1, \ldots, N,$$  \hspace{1cm} (4.5a)

$$g_{u,v} \leq \sum_{u' \in \text{In}(v)} f_{u',v} \quad \forall u = 1, \ldots, N, v = 1, \ldots, N,$$  \hspace{1cm} (4.5b)
where the right-hand-sides of constraints (4.5) indicate that nodes $u$ and $v$ respectively are present in the generated line. We use $\text{In}(u)$ ($\text{Out}(u)$) to refer to the set of nodes, including the source (sink), that have edges incoming to (outgoing from) node $u$.

Recalling that the cost $c_\ell$ of a bus line can be computed by summing the travel times between consecutive stops on the line, we search for a profitable line by solving the following integer optimization problem:

$$\text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E})) = \max_{f, g} \sum_{u=1}^{N} \sum_{v=1}^{N} (\bar{p}_{u,v} g_{u,v} - \delta_{u,v} \bar{q} f_{u,v})$$  \hspace{1cm} (4.6a)

s.t. \hspace{1cm} \sum_{u=1}^{N} f_{0,u} = 1  \hspace{1cm} (4.6b)

$$\sum_{u=1}^{N} f_{N+1,u} = 1  \hspace{1cm} (4.6c)$$

$$\sum_{v \in \text{Out}(u)} f_{u,v} - \sum_{v \in \text{In}(u)} f_{v,u} = 0  \hspace{1cm} \forall u = 1, \ldots, N,  \hspace{1cm} (4.6d)$$

$$g_{u,v} \leq \sum_{v' \in \text{In}(u)} f_{v',u}  \hspace{1cm} \forall u = 1, \ldots, N, v = 1, \ldots, N,  \hspace{1cm} (4.6e)$$

$$g_{u,v} \leq \sum_{u' \in \text{In}(v)} f_{u',v}  \hspace{1cm} \forall u = 1, \ldots, N, v = 1, \ldots, N,  \hspace{1cm} (4.6f)$$

$$\sum_{u \in S} \sum_{v \in S} f_{u,v} \leq |S| - 1  \hspace{1cm} \forall S \subset \mathcal{V},  \hspace{1cm} (4.6g)$$

$$f \in \{0, 1\}^{\mathcal{E}}, \ g \in \{0, 1\}^{N \times N}.  \hspace{1cm} (4.6h)$$

In Problem (4.6), the objective (4.6a) checks for the profitability condition (4.4). Constraints (4.6b) through (4.6d) are network flow constraints, and constraints (4.6e) and (4.6f) correspond to (4.5). Constraints (4.6g) are subtour elimination constraints that are required to prevent the formation of edge-disjoint cycles; otherwise the objective could gain credit for connecting stops belonging to separate cycles. Since the
number of potential cycles in the graph is large, these constraints are added lazily using branch-and-cut: violated constraints are identified and added to the problem at incumbent integral solutions.

Although the subproblem has been formulated in a basic way, a variety of conditions of interest to transit planners can be modeled using additional constraints. We provide some illustrative examples here.

- **Bus Depots**: In some cities, bus lines must begin and end at certain predesignated depot stops. This can be modeled in the subproblem by eliminating all edges between the source and non-depot nodes, and similarly for the sink.

- **Edge Lengths**: It may not be desirable for buses to make stops that are too close or far apart. This can be modeled in the subproblem by eliminating all edges between stops that are either too short or too long, which has the additional benefit of significantly sparsifying the graph $G(V, E)$.

- **Line Length**: It may not be desirable for a bus line to contain too many stops. This can be modeled in the subproblem by restricting the number of edges that are used.

- **Obstacles**: If it is impossible to travel from stop $u$ to stop $v$, then that edge can be eliminated.

Algorithm 1 describes the full column generation scheme.

### 4.3.2 Serving passengers with transfers

The assumption of Problem (4.1) that commuters will only take the bus if it can get them directly from origin to destination is a restrictive one. In ignoring transferring commuters, bus networks are forced to be excessively connected to gain ridership. However, this could lead to significant redundancies and therefore cost inefficiencies in the bus network. We therefore generalize to the case with two lines, where commuters are willing to make a single transfer. We call the model in Section 4.3.1 the *direct-route* model, and we call the model in this section the *single-transfer* model. It
Algorithm 1 Column generation algorithm for solving TNDP (4.1)

**Require:** Initial set of bus lines $\mathcal{L}$, tolerance $\epsilon > 0$

1: Solve $\text{MP}(\mathcal{L})$, get primal solution $(\bar{x}, \bar{\theta})$ and dual solution $(\bar{p}, \bar{q}, \bar{s})$

2: $J \leftarrow$ objective value of $\text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}))$

3: $\ell \leftarrow$ bus line solution to $\text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}))$

4: while $J > \epsilon$ do

5: $\hat{\mathcal{L}} \leftarrow \mathcal{L} \cup \ell$

6: Solve $\text{MP}(\hat{\mathcal{L}})$, get primal solution $(\bar{x}, \bar{\theta})$ and dual solution $(\bar{p}, \bar{q}, \bar{s})$

7: $J \leftarrow$ objective value of $\text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}))$

8: $\ell \leftarrow$ bus line solution to $\text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}))$

9: Solve $\text{TNDP}(\hat{\mathcal{L}})$, get integral solution $(x^*, \theta^*)$

10: return $x^*$

is possible to generalize our approach to an arbitrary number of transfers, but we restrict our attention to a single transfer for two reasons. The main reason is that more than one transfer tends to deter most of the population from taking the bus, with the exception of lower-income, transit-dependent riders. The time lost due to an inefficient sequence of transfers tends to fall disproportionately on these passengers who cannot afford alternatives, so we focus on allowing only one transfer in the interest of equity. In restricting our attention to only a single transfer, we also follow the example of papers such as [4].

Problem (4.1) can be extended to incorporate single transfers by modifying constraint (4.1b) into:

$$\theta_{u, v} \leq \sum_{\ell \in \mathcal{L}| \text{u} \in \text{stops}(\ell), v \in \text{stops}(\ell)} x_\ell + \sum_{(\ell_1, \ell_2): \text{u} \in \text{stops}(\ell_1), v \in \text{stops}(\ell_2), \text{stops}(\ell_1) \cap \text{stops}(\ell_2) \cap T_{u,v} \neq \emptyset} x_{\ell_1} x_{\ell_2} \quad (4.7)$$

for all $u = 1, \ldots, N$ and $v = 1, \ldots, N$. The second summation counts the number of single-transfer commute options, the $x_{\ell_1} x_{\ell_2}$ product indicates whether both lines $\ell_1$ and $\ell_2$ are operated, and $T_{u,v}$ represents a set of stops that could serve as viable transfer stops between stops $u$ and $v$. We further discuss $T_{u,v}$ at the end of this section; for now, it suffices to say that $T_{u,v}$ can be easily precomputed outside of the master problem.

Constraints (4.7) pose an issue in that they are nonlinear in the $x$ variables.
They can be reformulated into linear constraints by introducing auxiliary variables \( y \in \mathbb{R}^{|\mathcal{L}| \times |\mathcal{L}|} \) with the following:

\[
\theta_{u,v} \leq \sum_{\{\ell \in \mathcal{L} \mid u \in \text{stops}(\ell), v \in \text{stops}(\ell)\}} x_\ell + \sum_{(\ell_1, \ell_2) \colon u \in \text{stops}(\ell_1), v \in \text{stops}(\ell_2), \text{stops}(\ell_1) \cap \text{stops}(\ell_2) \cap \mathcal{T}_{u,v} \neq \emptyset} y_{\ell_1,\ell_2} \quad \forall u = 1, \ldots, N, \quad (4.8a)
\]

\[
y_{\ell_1,\ell_2} \leq x_{\ell_1} \quad \forall \ell_1, \ell_2 \in \mathcal{L}, \quad (4.8b)
\]

\[
y_{\ell_1,\ell_2} \leq x_{\ell_2} \quad \forall \ell_1, \ell_2 \in \mathcal{L}. \quad (4.8c)
\]

Note that it is not necessary to include a \( y_{\ell_1,\ell_2} \) variable for all pairs of lines \((\ell_1, \ell_2) \in \mathcal{L} \times \mathcal{L}\), just the intersecting pairs that could service some commute \((u, v)\). The irrelevant constraints may be eliminated from the master problem, with the corresponding dual values set to zero in the dual. For notational brevity, we consider the transfer from line \( \ell_1 \) to line \( \ell_2 \) to be distinct from the reverse of that transfer from line \( \ell_2 \) to line \( \ell_1 \). However, the number of variables and constraints can be reduced by replacing all instances of \( y_{\ell_2,\ell_1} \) with \( y_{\ell_1,\ell_2} \) when implementing this model.

Replacing constraint (4.2b) with (4.8), the single-transfer master problem is as follows:

\[
\text{MP2}(\mathcal{L}) = \max_{x,y,\theta} \sum_{u,v=1}^{N} d^u,v \theta_{u,v} \quad (4.9a)
\]

\[
\text{s.t.} \quad \theta_{u,v} \leq \sum_{\{\ell \in \mathcal{L} \mid u \in \text{stops}(\ell), v \in \text{stops}(\ell)\}} x_\ell + \sum_{(\ell_1, \ell_2) \colon u \in \text{stops}(\ell_1), v \in \text{stops}(\ell_2), \text{stops}(\ell_1) \cap \text{stops}(\ell_2) \cap \mathcal{T}_{u,v} \neq \emptyset} y_{\ell_1,\ell_2} \quad \forall u = 1, \ldots, N, \quad (4.9b)
\]

\[
y_{\ell_1,\ell_2} \leq x_{\ell_1} \quad \forall \ell_1, \ell_2 \in \mathcal{L}, \quad (4.9c)
\]

\[
y_{\ell_1,\ell_2} \leq x_{\ell_2} \quad \forall \ell_1, \ell_2 \in \mathcal{L}, \quad (4.9d)
\]

\[
c^\top x \leq B \quad (4.9e)
\]

\[
\theta \leq e \quad (4.9f)
\]

\[
x \geq 0 \quad (4.9g)
\]
The dual for the new master problem (4.9), with dual variables \( \pi^{(1)} \) and \( \pi^{(2)} \) corresponding to constraints (4.9c) and (4.9d), is as follows:

\[
D_2(\mathcal{L}) = \min_{p,q,s} \ Bq + e^T s \\
\text{s.t.} \quad - \sum_{u \in \text{stops}(\ell)} \sum_{v \in \text{stops}(\ell')} p_{u,v} - \sum_{\ell' \in \mathcal{L}} \left( \pi^{(1)}_{\ell,\ell'} + \pi^{(2)}_{\ell,\ell'} \right) + c_{\ell} q \geq 0 \quad \forall \ell \in \mathcal{L},
\]

\[
p_{u,v} + s_{u,v} = d_{u,v} \quad \forall u = 1, \ldots, N, v = 1, \ldots, N,
\]

\[
- \sum_{u \in \text{stops}(\ell_1), v \in \text{stops}(\ell_2)} p_{u,v} + \pi^{(1)}_{\ell_1,\ell_2} + \pi^{(2)}_{\ell_1,\ell_2} = 0 \quad \forall \ell_1 \in \mathcal{L}, \ell_2 \in \mathcal{L},
\]

\[
p, q, s, \pi^{(1)}, \pi^{(2)} \geq 0.
\]

However, due to the auxiliary constraints (4.9c) and (4.9d) linking the \( x \) and \( y \) variables, generating columns based on the dual (4.10) would require knowledge of \( \pi^{(1)} \) and \( \pi^{(2)} \) values that have not yet been defined, since the restricted master problem is only defined on a subset \( \mathcal{L} \subset \mathcal{L} \).

Column-and-constraint generation provides a remedy to this issue. At every iteration, using the primal solution \((\hat{x}, \hat{y}, \hat{\theta})\) and dual solution \((\hat{p}, \hat{q}, \hat{s}, \pi^{(1)}, \pi^{(2)})\) for the restricted problems \( \text{MP2}(\hat{\mathcal{L}}) \) and \( \text{D2}(\hat{\mathcal{L}}) \), we can construct solutions \((\hat{x}, \hat{y}, \hat{\theta})\) and \((\hat{p}, \hat{q}, \hat{s}, \pi^{(1)}, \pi^{(2)})\) for the full problems \( \text{MP2}(\mathcal{L}) \) and \( \text{D2}(\mathcal{L}) \). The full primal solution \((\hat{x}, \hat{y}, \hat{\theta})\) should be primal feasible for the full problem \( \text{MP2}(\mathcal{L}) \) and have the same objective value as the restricted solution. The full dual solution \((\hat{p}, \hat{q}, \hat{s}, \pi^{(1)}, \pi^{(2)})\) should satisfy complementary slackness conditions, and have the same objective values as in the restricted solution. However, this full dual solution need not be feasible for \( \text{D2}(\mathcal{L}) \). If a constraint of \( \text{D2}(\mathcal{L}) \) is violated, then add the corresponding element of \( \mathcal{L} \setminus \hat{\mathcal{L}} \) to \( \hat{\mathcal{L}} \) and proceed to the next iteration. If the new dual solution is feasible, then the optimality criteria are satisfied and the algorithm terminates. We refer the interested reader to [65] for worked examples on a similar approach through
branch-cut-and-price.

We now proceed to outline the construction of the full primal and dual solutions. The construction of the full primal solution \((\hat{x}, \hat{y}, \hat{\theta})\) is straightforward. The only variables that are missing are the \(x\) variables corresponding to lines \(\ell \in \mathcal{L} \setminus \mathcal{L}\), and \(y\) variables where one of the two lines is also in \(\mathcal{L} \setminus \mathcal{L}\). These variables can be set to zero to generate a feasible primal solution that has the same objective function value as the original solution (because \(\bar{\theta} = \hat{\theta}\)). The dual variables \(\hat{p}, \hat{q}, \hat{s}\) are all present from the outset. If both \(\ell_1 \in \mathcal{L}\) and \(\ell_2 \in \mathcal{L}\), then the dual values \(\hat{\pi}_{\ell_1,\ell_2}^{(1)}\) are already defined. However, if one or both of \(\ell_1\) and \(\ell_2\) are missing from \(\mathcal{L}\), then \(\hat{\pi}_{\ell_1,\ell_2}^{(1)}\) and \(\hat{\pi}_{\ell_1,\ell_2}^{(2)}\) have to be constructed from the primal solution \((\hat{x}, \hat{y}, \hat{\theta})\). For reasons that will become clear, we propose the following construction of \(\hat{\pi}_{\ell_1,\ell_2}^{(1)}\) and \(\hat{\pi}_{\ell_1,\ell_2}^{(2)}\):

\[
\hat{\pi}_{\ell_1,\ell_2}^{(1)} = \begin{cases} 
\hat{\pi}_{\ell_1,\ell_2}^{(1)} & \text{if } \hat{x}_{\ell_1} > 0, \hat{x}_{\ell_2} > 0, \\
0 & \text{if } \hat{x}_{\ell_1} > 0, \hat{x}_{\ell_2} = 0, \\
\sum_{u \in \text{steps}(\ell_1) \cap \text{steps}(\ell_2) \cap \mathcal{T}_{u,v} \neq \emptyset} u \cdot P_{u,v} & \text{if } \hat{x}_{\ell_1} = 0, \hat{x}_{\ell_2} > 0, \\
\frac{1}{2} \sum_{u \in \text{steps}(\ell_1) \cap \text{steps}(\ell_2) \cap \mathcal{T}_{u,v} \neq \emptyset} u \cdot P_{u,v} & \text{if } \hat{x}_{\ell_1} = 0, \hat{x}_{\ell_2} = 0, 
\end{cases}
\]

\[
\hat{\pi}_{\ell_1,\ell_2}^{(2)} = \begin{cases} 
\hat{\pi}_{\ell_1,\ell_2}^{(2)} & \text{if } \hat{x}_{\ell_1} > 0, \hat{x}_{\ell_2} > 0, \\
0 & \text{if } \hat{x}_{\ell_1} > 0, \hat{x}_{\ell_2} = 0, \\
\sum_{u \in \text{steps}(\ell_1) \cap \text{steps}(\ell_2) \cap \mathcal{T}_{u,v} \neq \emptyset} u \cdot P_{u,v} & \text{if } \hat{x}_{\ell_1} = 0, \hat{x}_{\ell_2} = 0, \\
\frac{1}{2} \sum_{u \in \text{steps}(\ell_1) \cap \text{steps}(\ell_2) \cap \mathcal{T}_{u,v} \neq \emptyset} u \cdot P_{u,v} & \text{if } \hat{x}_{\ell_1} = 0, \hat{x}_{\ell_2} > 0, 
\end{cases}
\]

It can be shown that the construction in equation (4.11) has the same objective value as the restricted dual, satisfies complementary slackness conditions, is feasible for constraint (4.10d), but is not necessarily feasible for constraint (4.10b). The objective value for the full dual solution is the same as in the restricted dual solution because \(\bar{q} = \hat{q}\) and \(\bar{s} = \hat{s}\). When \(\hat{x}_{\ell_1} > 0\) and \(\hat{x}_{\ell_2} = 0\), then it must be that \(\hat{y}_{\ell_1,\ell_2} = 0 < \hat{x}_{\ell_1}\), so the corresponding dual variable \(\hat{\pi}_{\ell_1,\ell_2}^{(1)}\) must be 0. Then, \(\hat{\pi}_{\ell_1,\ell_2}^{(2)}\) is determined by constraint (4.10d). The logic is analogous for the case where \(\hat{x}_{\ell_1} = 0\) and \(\hat{x}_{\ell_2} > 0\).
For the case where \( \hat{x}_{\ell_1} = \hat{x}_{\ell_2} = 0 \), we have that \( \hat{y}_{\ell_1, \ell_2} = 0 = \hat{x}_{\ell_1} = \hat{x}_{\ell_2} \), so both \( \hat{\pi}^{(1)}_{\ell_1, \ell_2} \) and \( \hat{\pi}^{(2)}_{\ell_1, \ell_2} \) can be nonzero. The cases where \( \hat{x}_{\ell_1} > 0 \) and \( \hat{x}_{\ell_2} > 0 \) were taken from the original restricted problems. Since they satisfy the complementary slackness conditions in the restricted problems, they satisfy complementary slackness here as well.

As the only source of infeasibility in the dual comes from constraint (4.10b), we have a natural extension of the condition for the direct-route model in equation (4.4) by including the set of terms \( \sum_{\ell' \in \mathcal{L}} \left( \hat{\pi}^{(1)}_{\ell, \ell'} + \hat{\pi}^{(2)}_{\ell', \ell} \right) \) to account for single-transfer commuters. Substituting the dual values (4.11) into constraint (4.10b), we obtain that a new bus line \( \ell \in \mathcal{L} \setminus \hat{\mathcal{L}} \) may be added to \( \hat{\mathcal{L}} \) if the following condition is met:

\[
c_{\ell} \hat{q} < \sum_{\nu=1}^{N} \sum_{\nu=1}^{N} \alpha_{\ell, u, v}(\hat{x}) \hat{p}_{u, v},
\]

where \( \alpha(\cdot) \) is defined so that

\[
\alpha_{\ell, u, v}(x) =
\begin{cases}
1 & \text{if } u \in \text{stops}(\ell), v \in \text{stops}(\ell), \\
1 & \text{if } \exists \ell' \in \mathcal{L} \text{ s.t. } x_{\ell'} > 0 \text{ and } u \in \text{stops}(\ell), v \in \text{stops}(\ell') \text{ or } v \in \text{stops}(\ell) \text{ and } u \in \text{stops}(\ell') \text{ and } \text{stops}(\ell) \cap \text{stops}(\ell') \cap \mathcal{T}_{u, v} \neq \emptyset, \\
\frac{1}{2} & \text{if } \forall \ell' \in \mathcal{L} \text{ s.t. } u \in \text{stops}(\ell), v \in \text{stops}(\ell') \text{ or } v \in \text{stops}(\ell) \text{ and } u \in \text{stops}(\ell') \text{ and } \text{stops}(\ell) \cap \text{stops}(\ell') \cap \mathcal{T}_{u, v} \neq \emptyset, \\
0 & \text{otherwise}.
\end{cases}
\]

The \( \alpha \) coefficients have a natural interpretation. Recalling that the dual variables \( p_{u, v} \) correspond to the ridership that stands to be gained by servicing commute \((u, v)\), each case in (4.13) corresponds to a different way of servicing the commute. All of the ridership is accrued if either \( u \) and \( v \) are connected directly, or if they are connected
indirectly through another line \( \ell' \) that was operational in the optimal solution to the restricted master problem. If \( u \) and \( v \) are only connected through lines that were not operational in the optimal solution to the restricted master problem, then only half of the ridership is accrued.

Modifying the subproblem to accommodate the new condition (4.12) is straightforward, but requires more variables to model the various cases in the \( \alpha \) coefficients. For the first case where the commutes are connected directly, we introduce the variables \( g^{(1)} \in [0, 1]^{N \times N} \) and add the following constraints for all commutes \((u, v)\):

\[
g_{u,v}^{(1)} \leq \sum_{v' \in \text{In}(u)} f_{v',u}, \tag{4.14a}
\]

\[
g_{u,v}^{(1)} \leq \sum_{u' \in \text{In}(v)} f_{u',v}, \tag{4.14b}
\]

where the \( f \) variables are defined as in Section 4.3.1.

For the second case, where the commute is serviced indirectly by transferring to or from a line which was operational in the restricted master solution, we introduce the variables \( g^{(2)} \in [0, 1]^{N \times N} \) and \( g^{(3)} \in [0, 1]^{N \times N} \), and add the following constraints for all commutes \((u, v)\):

\[
g_{u,v}^{(2)} \leq \sum_{v' \in \text{In}(u)} f_{v',u}, \tag{4.15a}
\]

\[
g_{u,v}^{(2)} \leq \sum_{\{w \in \text{In}(u) \mid \exists \ell' \in \mathcal{L} \text{ s.t.} v' \in \text{stops}(\ell'), u \in \text{stops}(\ell'), x_{\ell'} > 0\}} \sum_{v' \in \text{In}(w)} f_{v',w}, \tag{4.15b}
\]

\[
g_{u,v}^{(3)} \leq \sum_{u' \in \text{In}(v)} f_{u',v}, \tag{4.15c}
\]

\[
g_{u,v}^{(3)} \leq \sum_{\{w \in \text{In}(u) \mid \exists \ell' \in \mathcal{L} \text{ s.t.} u \in \text{stops}(\ell'), w \in \text{stops}(\ell'), x_{\ell'} > 0\}} \sum_{v' \in \text{In}(w)} f_{v',w}. \tag{4.15d}
\]

Note the dependence on the restricted master solution \( x \) in (4.15b) and (4.15d). The variables \( g_{u,v}^{(2)} \) model the case where the new line connects the origin \( u \) and a transfer
stop \( w \) that is already connected to the destination \( v \), and the variables \( g_{u,v}^{(3)} \) model the case where the new line connects destination \( v \) and a transfer stop \( w \) that is already connected to the origin \( u \).

For the third case, where the commute is serviced indirectly through a transfer but the connecting line is not yet operational, we introduce the variables \( g_{u,v}^{(4)} \in [0, 1]^{N \times N} \) and \( g_{u,v}^{(5)} \in [0, 1]^{N \times N} \), and add the following constraints for all commutes \((u, v)\):

\[
\begin{align*}
    &g_{u,v}^{(4)} \leq \sum_{u' \in \text{In}(u)} f_{u',u}, \quad (4.16a) \\
    &g_{u,v}^{(4)} \leq \sum_{\{w \in T_{u,v} | \forall \ell' \in \mathcal{L} \text{ s.t.} \ v \in \text{stops}(\ell'), w \in \text{stops}(\ell'), x_{u,v} = 0\}} \sum_{v' \in \text{In}(w)} f_{v',w}, \quad (4.16b) \\
    &g_{u,v}^{(5)} \leq \sum_{u' \in \text{In}(u)} f_{u',v}, \quad (4.16c) \\
    &g_{u,v}^{(5)} \leq \sum_{\{w \in T_{u,v} | \forall \ell' \in \mathcal{L} \text{ s.t.} \ u \in \text{stops}(\ell'), w \in \text{stops}(\ell'), x_{u,v} = 0\}} \sum_{v' \in \text{In}(w)} f_{v',w}. \quad (4.16d)
\end{align*}
\]

Again, note the dependence on the restricted master solution \( x \) in (4.16b) and (4.16d).

Finally, we add the following constraint to ensure that only one of the cases is chosen:

\[
g_{u,v}^{(1)} + g_{u,v}^{(2)} + g_{u,v}^{(3)} + g_{u,v}^{(4)} + g_{u,v}^{(5)} \leq 1,
\]

and modify the subproblem objective as follows:

\[
\max_{f, g} \sum_{u=1}^{N} \sum_{v=1}^{N} \bar{p}_{u,v} \left( g_{u,v}^{(1)} + \frac{1}{2} \left( g_{u,v}^{(2)} + g_{u,v}^{(3)} + g_{u,v}^{(4)} + g_{u,v}^{(5)} \right) \right) - \delta_{u,v} \bar{q} f_{u,v}. \quad (4.17)
\]

### 4.3.3 Considering travel times

The task of computing travel times and the resulting route choices directly within the line-generation subproblem is a complex one. Prior works such as [4] and [31] have avoided this complexity by alternating between route generation and assignment.
models. However, both works consider problems of a smaller size than ours. Although [31] present an exact formulation, they allow for arbitrarily long travel times and ignore transfers. This is not entirely realistic, since if the commuters’ only options are routes of long travel time and many transfers, they will likely leave the transit system for other alternatives. By contrast, the assignment model of [4] is detailed, but the route generation is heuristic.

We now turn to the problem of travel time in our own framework, and first address commutes that are confined to a single line. Recalling that the direct-route model assumes that commuters are willing to take the bus as long as there is a route connecting its origin and destination, this implies that a Hamiltonian path through all of the bus stops would adequately service all demand, so long as it is feasible for the budget constraint. However, such a path will likely be inefficient for many commuters, particularly for commuters whose origins and destinations lie close to opposing bus terminals. Such commuters may likely seek alternative transit options. In our model, we will assume that commuters for commute \((u, v)\) where \(u\) and \(v\) are on the same line, will only take the bus if they are able to follow some sequence of edges \(\omega = \{(u, w_2), (w_2, w_3), \ldots, (w_k, v)\}\) that satisfies the following condition:

\[
\sum_{(w_i, w_j) \in \omega} \delta_{w_i, w_j} \leq \Gamma \delta_{u,v},
\]

for some \(\Gamma > 1\). Namely, commuters will only take the bus if the travel time experienced on the bus network is not much greater than the direct travel time between origin and destination.

Incorporating travel time restrictions in the master problem is straightforward. In the direct-route model, we modify constraint (4.2b) so that the right-hand-side omits any \(x_{\ell}\) terms that correspond to lines where the travel time from origin \(u\) to destination \(v\) on line \(\ell\) is excessive, and the necessary modifications to the single-transfer model are similar. However, the subproblem poses a greater challenge. The most natural approach is to ensure that ridership is only accrued along paths that are sufficiently short using cutting planes. At every intermediate solution in the
branch-and-bound tree, we check whether the bus line contains some path \( \omega = \{(w_1, w_2), (w_2, w_3), \ldots, (w_{k-1}, w_k)\} \) with length exceeding \( \Gamma \delta_{w_1, w_k} \). If so, we introduce the following constraint:

\[
g^{(1)}_{w_1, w_k} \leq \sum_{i=1}^{k-1} (1 - f_{w_i, w_{i+1}}). \tag{4.19}
\]

Although the approach (4.19) achieves the goal of counting only those commuters whose travel times are not excessive, this quickly becomes intractable even on small networks. The key difficulty is that many paths may exist that connect each pair of stops, requiring the addition of many cutting planes before the optimizer can prove optimality.

We propose a solution that achieves tractability by enforcing stricter constraints on the bus lines produced by the subproblem. Instead of using condition (4.18) only to calculate the reduced cost of the bus line, we enforce the condition for every pair of stops on the line. Although this may be a little excessive, it seems to hold in practice; on the MBTA’s key network, which are defined by the agency as the high-ridership and high-frequency bus lines, we found that 97.8% of the stop pairs within the bus lines satisfied condition (4.18) for \( \Gamma = 1.5 \), and 99.8% of the stop pairs satisfied it for \( \Gamma = 2.0 \).

Our approach is implemented as follows. At intermediate solutions in the branch-and-bound tree, we check whether the bus line contains some path

\[
\omega = \{(w_1, w_2), \ldots, (w_{k-1}, w_k)\}
\]

with end-to-end travel time exceeding \( \Gamma \delta_{w_1, w_k} \). If such a \( \omega \) is found, we introduce the following constraint:

\[
\sum_{i=1}^{k-1} f_{w_i, w_{i+1}} \leq k - 2. \tag{4.20}
\]

Constraint (4.20) suffers from the same issue as (4.19) where cutting a single path at
(a) A Boston example. The transfer at Park Street makes the travel time from Central Square to Hynes Convention Center significantly longer than the direct path, which is unappealing for commuters.

(b) A cartoon of the Central Square (u) to Hynes Convention Center (v) via Park Street (w) example.

Figure 4-1: An illustration of transfers excessive travel times due to transfers.

e a time will be intractable on large networks. However, if \( \omega \) is also the shortest path on all of the stops \( \sigma = \{w_1, \ldots, w_k\} \), we instead use the following, stronger constraint:

\[
\sum_{w_i \in \sigma \setminus w_j \in \sigma} f_{w_i, w_j} \leq k - 2,
\]

which cuts not just the path \( \omega \) corresponding to the current solution, but also all other paths connecting the stops \( \sigma \). Constraint (4.21) is valid because if the shortest path on stops \( \sigma \) does not satisfy condition (4.18), then no other paths on stops \( \sigma \) will satisfy that condition either. Verifying that \( \omega \) is the shortest path on all of the stops \( \sigma \) could potentially be costly; in the worst case, such an operation would need to be performed on all subpaths of all visited solutions in the branch-and-bound tree. Rather than verifying this exactly, we use insertion-based heuristics to compute the shortest path on \( \sigma \) and compare the cost of that path to that of \( \omega \).

We now turn to travel time for single-transfer commuters who are forced to make a transfer through less direct routes. An illustrative example from the Boston-area
subway and bus network is shown in Figure 4-1. To get from Central Square to Hynes Convention Center via the subway, a commuter must take the Red Line (dashed red) from Central Square to Park Street before transferring to the Green Line (dot-dashed green). This is a significant increase in travel time relative to the direct bus route (solid gray), making the subway unappealing to commuters.

To limit the travel time experienced by single-transfer commuters, we precompute transfer stops for each commute that would not take commuters too far out of their way. In particular, we define a set of transfer stops for commute \((u, v)\) as follows:

\[
T_{u,v} := \{ w \in [N] \mid \delta_{u,w} + \delta_{w,v} \leq \Lambda \delta_{u,v} \},
\]

where \(\Lambda\) is a constant greater than one. In Figure 4-1b, the displayed transfer stop \(w\) would be disqualified from being in \(T_{u,v}\) based on definition (4.22). The restriction of the transfer stops, combined with the travel time restriction of (4.18), has the effect of preventing a commuter’s total travel time for commute \((u, v)\) from being more than \(\Gamma \Lambda\) times the direct travel time \(\delta_{u,v}\).

Restricting the set of transfers for a commute \((u, v)\) to only occur through a stop \(w \in T_{u,v}\) keeps travel times on the network reasonable and prevents generation of excessively long commuter routes. Furthermore, this modeling approach makes the underlying network of edges significantly sparser, resulting in a tractable subproblem.

### 4.3.4 Speeding up the subproblem

Much of the computational expense in the subproblem comes from the addition of the subtour elimination constraints (4.6g) and the travel time constraints (4.21), particularly in dense graphs. We propose a fast preprocessing step that significantly reduces the running time of the subproblem. The intent is to preprocess the edge set \(E\) so that cycles will not appear in the subproblem graph, thus removing the need for subtour elimination constraints. As an auxiliary effect, the preprocessing tends to limit travel times on the network, so fewer travel time constraints will be needed as well. For the remainder of the paper, we will call the subproblem on the full edge set \(E\) the full subproblem.
set with subtour elimination constraints the full subproblem, and the subproblem on the preprocessed edge set the preprocessed subproblem.

The preprocessing is done by choosing a particular geographic direction and removing all edges that are not aligned with that geographic direction, within a certain range of tolerance. For a direction \( \vec{z} \) and a range \( \Delta \), the preprocessed edge set \( \mathcal{E}(\vec{z}, \Delta) \in \mathcal{E} \) is defined as follows:

\[
\mathcal{E}(\vec{z}, \Delta) = \left\{ (u, v) \in \mathcal{E} \left| \frac{\text{vec}(u, v) \cdot \vec{z}}{\|\text{vec}(u, v)\| \|\vec{z}\|} > \Delta \right. \right\}, \tag{4.23}
\]

where we use the notation \( \text{vec}(u, v) \) to indicate the vector pointing from stop \( u \) to stop \( v \). For \( \Delta \geq 0 \), cycles will not appear in the graph, and larger values of \( \Delta \) will tend to make the bus lines more oriented along the direction \( \vec{z} \). Orienting the bus line along the direction \( \vec{z} \) will also keep travel times short, which will reduce the number of travel time constraints that need to be added. In computation, we must choose a set of directions \( \vec{z} \) that is exhaustive, as well as set \( \Delta \) to a value that eliminates cycles while also not overly restricting the edge set so as to eliminate the optimal solution.

Algorithm 1 is modified to accommodate the edge-preprocessing as follows. The subproblem is solved over an exhaustive set of directions, producing a set of different bus lines. If the subproblem terminates with positive objective value for any direction \( \vec{z} \), then the most profitable of these lines are added to \( \bar{\mathcal{L}} \). If all versions of the subproblem terminate with objective value zero, then there are no profitable lines that can be added, and the algorithm terminates with the optimal network design for the linear relaxation (4.2). This process is described in full detail in Algorithm 2. The loop in line 2 is easily parallelizable over all directions, allowing for further computational speedup.

In the next section, we demonstrate the benefit and tractability of Algorithms 1 and 2.
Algorithm 2 Column-and-constraint generation algorithm for solving TNDP (4.1), with edge preprocessing

Require: Initial set of bus lines $\mathcal{L}$, tolerance $\epsilon > 0$, line directedness parameter $\Delta \in [0, 1]$

1: Solve MP($\mathcal{L}$), get primal solution $(\bar{x}, \bar{\theta})$ and dual solution $(\bar{p}, \bar{q}, \bar{s})$

2: for all directions $\bar{z}$ do
3: \ $J_{\bar{z}} \leftarrow \text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}(\bar{z}, \Delta)))$
4: \ $\ell_{\bar{z}} \leftarrow \text{bus line solution to SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}(\bar{z}, \Delta)))$
5: while max$_{\bar{z}} J_{\bar{z}} > \epsilon$ do
6: \ $\mathcal{L} \leftarrow \mathcal{L} \cup \ell_{\arg \max_{\bar{z}} J_{\bar{z}}}$
7: \ Solve MP($\mathcal{L}$), get primal solution $(\bar{x}, \bar{\theta})$ and dual solution $(\bar{p}, \bar{q}, \bar{s})$
8: for all directions $\bar{z}$ do
9: \ $J_{\bar{z}} \leftarrow \text{SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}(\bar{z}, \Delta)))$
10: \ $\ell_{\bar{z}} \leftarrow \text{bus line solution to SP}(\bar{p}, \bar{q}; G(\mathcal{V}, \mathcal{E}(\bar{z}, \Delta)))$
11: Solve TNDP($\mathcal{L}$), get integral solution $(x^*, \theta^*)$
12: return $x^*$

4.4 Computational Results

In this section, we evaluate the performance of our column-and-constraint generation algorithm on synthetic and real-world data. All methods were implemented using the Julia language [25] and the optimization package JuMP [114] using the Gurobi solver v7.0 [82]. Computational experiments in Section 4.4.1 were run on a laptop with an Intel i7-6500U processor and 16GB of RAM. Computational experiments in Sections 4.4.2 and 4.4.3 were scheduled as jobs on a cluster: each job used one core with 16GB of memory on Dell C6300 machines with Intel E5-2690 v4 2.6 GHz/35M Cache processors.

In Section 4.4.1, we show that Algorithm 1 produces intuitive results for a small synthetic grid network. In Section 4.4.2, we compare the performance of the preprocessed subproblem with edge preprocessing to the full subproblem, and show that the preprocessing speeds up solve time by an order of magnitude while also preserving competitive objective values. In Section 4.4.3, we demonstrate that Algorithm 2 is tractable for a large-scale example in Boston with hundreds of stops, and is able to improve ridership over the existing bus network.
4.4.1 A small synthetic network

Our first computational experiment was inspired by the Houston network re-design [27], which dramatically simplified its bus routing from a hub-and-spoke framework to a pure grid network. The appeal of a grid network is that due to its relative simplicity, a commuter can get from any origin to any destination on the grid with at most one transfer. Our computational experiments on the synthetic dataset aim to illustrate the intuitive appeal of the grid and to illuminate the differences between the various models of Section 4.3.

We created a toy sixteen-stop network with coordinates on the integer lattice \( \{1, 2, 3, 4\}^2 \), representing the intersections of cross streets that we label A-D (directed vertically) and W-Z (directed horizontally). To avoid degeneracy and thereby improve computation time, a small amount of noise uniformly distributed between 0 and 0.01 was added to the coordinates of each stop. Edges were allowed between every stop and its horizontal, vertical, and diagonal immediate neighbors. Demand between every pair of stops \((u, v)\) was set to \(d_{u,v} = 100\), for a total of \(\sum_{u,v} d_{u,v} = 24,000\). The budget was sent to \(B = 24\), which is exactly the budget needed to sustain a grid network. For such a small network, we were able to solve the full subproblem with subtour elimination constraints (4.6g).

We first tested the direct-route subproblem without travel time restrictions. On this small network, the subproblem solved nearly instantaneously, and the first bus line that is generated, shown in Figure 4-2, achieves global optimality, connecting every stop possible and therefore servicing all demand. Note that because the budget constraint was not binding in the first iteration of the algorithm, the value of the corresponding dual variable \(q\) was zero, and there was therefore no pressure for the subproblem to find the most cost-effective bus line. As a result, this solution, although feasible for \(B = 24\), is certainly not the most cost-efficient one to connect all stops.

Although this solution is optimal, it is not practically useful. The full subproblem with subtour-elimination constraints allows excessive flexibility and produces a complex winding bus path, whereas real-life bus lines are often simpler and more directed.
Furthermore, ridership here is most likely overestimated; although the terminal stops (B,Z) and (C,Y) are connected by the line, a commuter would likely prefer to seek transit options that involve less travel time.

Figure 4-2: Synthetic bus network generated under the direct-route model without travel time constraints. The network above was produced using the direct-route column generation procedure without edge preprocessing, with budget $B = 24$. A single line is produced connecting all of the stops.

Next, we designed a network using the direct-route model with travel time restrictions. We set $\Gamma = 1.5$, meaning that commuters experience travel times no longer than 50% more than the direct travel time between their origin and destination. The algorithm terminated after eighteen iterations, in 103 seconds. The bus lines that appear in the integral solution are shown in Figure 4-3; each panel shows a bus line labeled by the iteration in which it was produced. Figure 4-3 shows some improvement over Figure 4-2; all of the bus lines adhere to the simple underlying grid structure, as compared to the winding bus path produced without the travel time restrictions. However, under the assumption that passengers will only take the bus if they can go directly from origin to destination, the network is excessively connected and has some redundancy. For example, three of the four bus lines traverse W Street to some extent. This demonstrates that modeling transferring behavior is essential, not just for realism from the passenger perspective, but also for efficiency in the bus networks produced by the model.

Finally, we designed a network using the single-transfer model with travel time restrictions. We set $\Gamma = \Lambda = 1.5$, meaning that commuters along a single line experience travel times no longer than 50% more than the direct travel time between their origin and destination, and commuters who transfer experience travel times no
Figure 4-3: Synthetic bus network generated under the direct-route model with travel time constraints. The network above was produced using the direct-route column generation procedure without edge preprocessing, with budget $B = 24$.

more than 125% more than the direct travel time between their origin and destination. Each panel in Figure 4-4 illustrates bus lines as they are generated, with transparency levels indicating the value of $x_\ell \in [0, 1]$ that entered the basis. For clarity, bus lines that have values $x_\ell = 0.0$ are not shown in any of the panels.

Once again, Figure 4-4 shows that the travel time restrictions force the bus lines that are generated to all adhere to the simple underlying grid structure. The algorithm terminated in eight iterations, which took about one minute. In the first and second iterations, L-shaped bus lines are produced along the perimeter of the network. In the sixth iteration, the bus line along B Street appears with $x_\ell < 0$; this is because the bus line also includes edges along Z and W Streets, having incorporated those before the budget constraint was tight, and so when the budget constraint becomes tight it is no longer possible for that line to be included at $x_\ell = 1$. Subsequent iterations then produce a bus line that traverses only B Street, which can be included in the network at $x_\ell = 1$. The final grid corresponds to an objective of 24,000, which is the total demand on the network. This illustrates that by modeling transferring behavior and travel time preferences, the optimal bus network develops a clear grid structure that is physically simple and actually implementable.
Figure 4-4: Synthetic bus network with single-transfer column-and-constraint generation and travel time constraints. We illustrate the network produced in each iteration of the single-transfer column-and-constraint generation procedure. The budget was set to $B = 1.5$, which is exactly the budget needed to sustain a grid network.
These computational results are not surprising; from a grid-type network of stops one would expect an efficient solution to have a grid structure. However, they provide a useful contrast between the networks generated with and without travel time restrictions and transfers, with the more realistic models producing intuitive and simple results. This is a key step towards implementability.

### 4.4.2 Solving the subproblem at scale

Our next computational experiment aims to compare the performance of the preprocessed subproblem proposed in Section 4.3.4 to the full subproblem with subtour elimination constraints (4.6g). To this end, we generated 25 random networks and demand matrices of varying sizes from $N = 50$ to $N = 70$. The stops were random points generated in the unit square $[0, 1]^2$, and edges were drawn between stops that were no more than 0.2 units apart. Demand matrices were generated to have support over 20% of the origin-destination pairs, with each nonzero element set to ten units. We then solved a single iteration of the single-transfer subproblem with and without preprocessing on each network, recording the running time and objective value each time. The preprocessing was run at a range of $\Delta$ values between 0.4 (most restrictive, fewest edges) to 0.0 (least restrictive, most edges), over a range of seventeen directions spaced $\frac{\pi}{16}$ radians apart between 0 and $\pi$. Like in the previous subsection, we set $\Gamma = \Lambda = 1.5$.

Solving the full subproblem to optimality was challenging. We used the solution from the $\Delta = 0.0$ preprocessed subproblem as a warm start, and let the solver run for up to 48 hours. We also tried solving the preprocessed subproblem without a warm start, but found that adding a warm start generally improved the solution quality. For $N = 50$, only two instances hit the time limits without proving optimality, but for $N = 60$, sixteen cases hit the time limit, and for $N = 70$, all instances hit the time limit.

However, by preprocessing the subproblem, we were able to speed up running time from days to minutes. The running times are shown in Table 4.2, as well as the number of successful iterations (out of 25 total) and the average gap between the
<table>
<thead>
<tr>
<th>$N$</th>
<th>Subproblem</th>
<th>$\Delta$</th>
<th>Run Time (s)</th>
<th>Successes</th>
<th>Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>Preprocessed</td>
<td>0.4</td>
<td>35</td>
<td>5</td>
<td>11.7</td>
</tr>
<tr>
<td>50</td>
<td>Preprocessed</td>
<td>0.2</td>
<td>65</td>
<td>14</td>
<td>2.9</td>
</tr>
<tr>
<td>50</td>
<td>Preprocessed</td>
<td>0.0</td>
<td>74</td>
<td>19</td>
<td>0.4</td>
</tr>
<tr>
<td>60</td>
<td>Preprocessed</td>
<td>0.4</td>
<td>84</td>
<td>1</td>
<td>15.7</td>
</tr>
<tr>
<td>60</td>
<td>Preprocessed</td>
<td>0.2</td>
<td>121</td>
<td>10</td>
<td>5.1</td>
</tr>
<tr>
<td>60</td>
<td>Preprocessed</td>
<td>0.0</td>
<td>168</td>
<td>20</td>
<td>1.2</td>
</tr>
<tr>
<td>70</td>
<td>Preprocessed</td>
<td>0.4</td>
<td>182</td>
<td>1</td>
<td>15.6</td>
</tr>
<tr>
<td>70</td>
<td>Preprocessed</td>
<td>0.2</td>
<td>219</td>
<td>6</td>
<td>5.7</td>
</tr>
<tr>
<td>70</td>
<td>Preprocessed</td>
<td>0.0</td>
<td>670</td>
<td>22</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 4.2: Performance of preprocessed subproblem as compared to the full subproblem. We compared both approaches in performance and computation time (the full subproblem was capped at 48 hours). Each row of the table represents 25 different simulations.

preprocessed and full subproblem objectives. Here, a successful iteration of the preprocessed subproblem was defined as achieving an objective as good as that achieved by the full subproblem. The average gap is reported as

$$\text{Gap} = 1 - \frac{\sum_{i=1}^{25} J_{\text{preprocessed}}^{(i)}}{\sum_{i=1}^{25} J_{\text{full}}^{(i)}},$$

where $J_{\text{preprocessed}}^{(i)}$ ( $J_{\text{full}}^{(i)}$) represents the objective achieved by the preprocessed (full) subproblem in simulation $i$. As expected, the higher $\Delta$ values corresponding to more heavily preprocessed edge sets were less competitive, particularly at large $N$. It could be that at large $N$, the effect of heavy edge preprocessing is exacerbated by the increased density of the graphs. More encouragingly, the performance of the preprocessing for the lowest value $\Delta = 0.0$ remained competitive at larger $N$, with 22 successes out of the 25 preprocessed simulations.

The solver progress in the full subproblem is shown for an illustrative example for $N = 40$ in Figure 4-5. In this case, the optimal solution was found after 49 seconds, and the remainder of the time was spent proving optimality. By contrast, the subproblem was solved to optimality on preprocessed edge sets over seventeen directions for $\Delta = 0.0$ in 46.6 seconds, and achieved the same objective value of 260.
Figure 4-5: Optimization lower and upper bounds for an iteration of the full subproblem. The bounds are based on a $N = 40$ random network. Because the subproblem is a maximization problem, the lower bound represents the actual solution values.

Even for $\Delta = 0.4$, an objective of 230 was reached in 15.4 seconds, which is 88.5% of the optimal objective value.

These results illustrate that for large networks, it is useful to preprocess the subproblem before solving. The first reason is that a flexible enough $\Delta$ parameter keeps enough edges to find quality or even optimal solutions. On these cases where points were uniformly spread through a square, the most flexible $\Delta = 0.0$ seemed to be the best choice; however, for networks where the stops are not spread uniformly, a more restrictive $\Delta$ may still provide good results. The second motivation for preprocessing is that in the full subproblem, significant time is spent on proving optimality on the full edge set after the optimal solution is found, while the preprocessed subproblems are able to terminate quickly due to the relative ease of proving optimality on the preprocessed edge sets. Finally, if further improvements are desired, then the solution from the preprocessed subproblem can be used as a warm start to the full problem.

4.4.3 A large-scale case study from Boston

Our last set of computational experiments focus on the bus network in the greater Boston area, operated by the Massachusetts Bay Transportation Authority (MBTA). Unlike Houston, Boston does not have a grid-based street network, and its bus network is much more complicated. We show that our method is tractable for a real-world
sized network of hundreds of stops, and that real benefits can be realized from taking an optimization-based approach to transit network design.

We obtained bus stop and line information from the General Transit Feed Specification (GTFS) provided by the [120]. We focused on the stops and lines in the “key” network, which is defined by the MBTA as the (sub)network of high-ridership and high-frequency bus lines, comprising \( N = 410 \) stops and \( L = 23 \) lines. In considering new edges to add to the network, we only considered edges of length 1.0 miles or less, since 97.5% of the edges in the existing network fell into this category. This gave us 4,893 edges to consider. We restricted ourselves to bus lines containing at most 40 stops, with 21 of the of the 23 existing bus lines in our dataset satisfying this criterion.

We also obtained automated passenger count data from 2017 through a private data request to the MBTA, which provided the average number of passengers boarding and alighting buses at each stop for each hour. Using techniques from [24], we estimated the hourly origin-destination demand matrices, and aggregated them to produce a daily demand matrix. For computational tractability, we ignored origin-destination pairs that saw fewer than 25 passengers over the course of the entire day. This accounts for a small fraction of the total demand, and the resulting demand matrix still has a total of 108,015 trips, with 270 of the 410 stops seeing activity. Note that it may not be possible to serve all of the demand; if physical restrictions on the edge set prevent an origin and destination from being connected, then the corresponding demand will remain unserved. When accounting for the connectivity of the network, the total volume of the serviceable demand was 104,444 trips.

As in the previous two sets of computational experiments, we set \( \Gamma = \Lambda = 1.5 \). We do not claim that these are the values for the Boston network; the intention is merely to set these parameters to reasonable values on which to test our approach. If our approach were to be implemented, additional surveys should be conducted to determine true values of \( \Gamma \) and \( \Lambda \).

Motivated by the strong computational performance of the preprocessed subproblem using the restricted edge sets in Section 4.4.2, we used the preprocessed subproblem for our large-scale Boston case study. A key factor influencing the tractability of
Table 4.3: Objectives of the original MBTA network and the optimized network

<table>
<thead>
<tr>
<th>Demand Matrix</th>
<th>Budget</th>
<th>MBTA Network</th>
<th></th>
<th>Optimized Network</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Objective</td>
<td>% Ridership</td>
<td>Objective</td>
<td>% Ridership</td>
</tr>
<tr>
<td>MBTA</td>
<td>50</td>
<td>73,575</td>
<td>70.4%</td>
<td>80,267</td>
<td>76.8%</td>
</tr>
<tr>
<td>MBTA</td>
<td>100</td>
<td>88,716</td>
<td>84.9%</td>
<td>98,487</td>
<td>94.3%</td>
</tr>
<tr>
<td>MBTA</td>
<td>150</td>
<td>89,472</td>
<td>85.7%</td>
<td>103,092</td>
<td>98.7%</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>25</td>
<td>41,149</td>
<td>35.9%</td>
<td>97,374</td>
<td>84.9%</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>50</td>
<td>70,557</td>
<td>61.5%</td>
<td>110,220</td>
<td>96.1%</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>100</td>
<td>78,954</td>
<td>68.6%</td>
<td>113,416</td>
<td>98.8%</td>
</tr>
</tbody>
</table>

our approach is the choice of the $\Delta$ parameter. We expect the choice of $\Delta$ to depend on the network: although $\Delta = 0.0$ or $\Delta = 0.2$ seem to be the best choice for the synthetic network in 4.4.2, those might not necessarily be the best choice for this network. We tested $\Delta = 0.4$ and $\Delta = 0.2$, but found the difference in the objective values for the lower parameter to be negligible while significantly increasing the runtime. Therefore we report all our results with $\Delta = 0.4$ and did not further test lower values. The algorithm was capped at a maximum of 30 iterations, which captures most of the gains in a reasonable amount of time.

Since the counts provided by the MBTA are taken entirely from passengers who are already using MBTA services, we expect the existing network to be adequate, and were interested in seeing what the performance gains might be using a demand matrix that did not come from the MBTA. Furthermore, with significant concerns about how the rise of alternatives such as ride-sharing services might impact ridership for public transit [33], it is useful to examine the demand from alternative modes of transport to see how the bus network might better service those who have chosen those who did not elect to take the bus. Blue Bikes (formerly Hubway), a bike-sharing company in the Boston area, made its 2011-13 trip data publicly available through a data challenge [127]; by mapping each Blue Bikes dock to the closest MBTA stop, we obtained an alternative origin-destination matrix. For maximal impact, we focused on the trips from August 2013, which has the highest monthly trip volume of 126,934 trips. For context, the highest achievable trip volume was 114,746 when accounting for the connectivity of the network.

115
Table 4.4: Characteristics of the original MBTA network and the optimized network

<table>
<thead>
<tr>
<th>Demand Matrix</th>
<th>Budget</th>
<th>Bus Lines Used</th>
<th>Edges Used</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Original (%)</td>
<td>New (%)</td>
</tr>
<tr>
<td>MBTA</td>
<td>50</td>
<td>21.7</td>
<td>13.3</td>
</tr>
<tr>
<td>MBTA</td>
<td>100</td>
<td>60.9</td>
<td>10.0</td>
</tr>
<tr>
<td>MBTA</td>
<td>150</td>
<td>69.6</td>
<td>20.0</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>25</td>
<td>0.0</td>
<td>23.3</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>50</td>
<td>4.3</td>
<td>30.0</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>100</td>
<td>30.4</td>
<td>78.6</td>
</tr>
</tbody>
</table>

The results of Algorithm 2 over a range of budgets is shown in Table 4.3 for both datasets. We begin with the MBTA key network as the initial set of lines \( \mathcal{L} \), before improving it using Algorithm 2. This improves ridership for the MBTA network by 6.4% in the low-budget \( B = 50 \) case and 13.0% in the high-budget \( B = 150 \) case. At the higher budgets \( B = 100 \) and \( B = 150 \), the network produced by Algorithm 2 is able to service nearly all of the demand.

Even higher gains in the objective are seen on the Blue Bikes dataset, as significantly less of the demand is served by the existing network. The Blue Bikes experiments were run on a range of smaller budgets from 50 to 100, as the Blue Bikes only saw demand between 72 stops and therefore provided a significantly sparser demand matrix. At \( B = 100 \), the existing MBTA network was only able to serve 68.6% of the demand; this increased to 98.8% after running the algorithm.

Some characteristics of the optimal network designs are summarized in Table 4.4. This includes the percentage of the original MBTA bus lines that appear in the optimal design, the percentage of the generated bus lines that appear in the optimal network design, and the percentages of edges in the optimal network design that did not appear in the original MBTA network. Unsurprisingly, the Blue Bikes demand matrix in general uses significantly fewer bus lines from the original MBTA network in order to service the demand (0 – 30.4% as compared to 21.7 – 69.6%), and a much higher percentage of the edges used in the optimal network design are new (33.2 – 46.2% as compared to 10.4 – 16.9%), showing that the network must change more significantly to service the Blue Bikes demand as opposed to the MBTA.
Figure 4-6: Boston-area bus networks generated from the MBTA demand matrix. Blue lines represent the original network, red lines represent bus lines that were generated by the algorithm, and green lines represent the final optimized bus network.

demand. At lower budgets, the percentage of edges that are new edges tends to be higher, indicating that when the network is under a tighter budget constraint, it must undergo more changes to adequately service demand. At lower budgets, a smaller proportion of the new bus lines tends to appear in the final network design, indicating that there is some spurious bus line generation.

The original and optimized networks for the MBTA and Blue Bikes datasets on a selection of budgets are shown in Figures 4-6 and 4-7, respectively. In these figures, blue lines represent the original network, red lines represent bus lines that were generated by the algorithm, and green lines represent the final optimized network. For the MBTA dataset, the optimized bus lines appear to track the original bus lines fairly closely, but many new connections are seen under the Blue Bikes dataset. The Blue Bikes dataset also covers a smaller geographic region, which reflects the relatively lower coverage area for bike-sharing relative to buses. For both datasets, under the lower budget of $B = 50$, the bus lines are largely concentrated in the downtown area, but with higher budgets, the bus lines can additionally cover the outlying suburban areas.

In addition to enabling significant ridership gains, Algorithm 2 is also tractable. The running times for each of the budgets and demand scenarios are shown in Ta-
Figure 4-7: Boston-area bus networks generated from the Blue Bikes demand matrix. Blue lines represent the original network, red lines represent bus lines that were generated by the algorithm, and green lines represent the final optimized bus network.

Table 4.5: Running time for Algorithm 2 on the Boston dataset

<table>
<thead>
<tr>
<th>Demand Matrix</th>
<th>Budget</th>
<th>Iterations</th>
<th>Running Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>to 50%</td>
</tr>
<tr>
<td>MBTA</td>
<td>50</td>
<td>30</td>
<td>3.1</td>
</tr>
<tr>
<td>MBTA</td>
<td>100</td>
<td>30</td>
<td>0.8</td>
</tr>
<tr>
<td>MBTA</td>
<td>150</td>
<td>30</td>
<td>0.8</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>25</td>
<td>30</td>
<td>1.9</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>50</td>
<td>30</td>
<td>1.8</td>
</tr>
<tr>
<td>Blue Bikes</td>
<td>100</td>
<td>14</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Table 4.5. The “to 50%” column represents the time it took to achieve 50% of the improvement, and similarly for the “to 90%” column. We observe diminishing gains with running time, where it takes only between one and three hours to see 50% of the improvement, but six to thirteen hours to terminate. This effect is particularly pronounced at higher budgets. However, since network design is an offline problem and cities only rarely redesign their transit networks, this an appropriate level of computational effort to be practically useful.
4.5 Conclusion

In this work, we address the problem of designing bus lines for urban transit networks. In particular, we seek to maximize ridership on a bus network, accounting for the fact that passengers will choose to take the bus if one of the possible routes was appealing in travel time and number of transfers. We discuss how to incorporate a variety of physical and operational constraints on bus lines, and present computational experiments to show our method’s tractability and usefulness. In the first experiment, we show that our method produces a simple grid network from a synthetic dataset, which mirrors real-life observations in cities such as Houston on the appeal of grid networks. In the second, we propose a preprocessing step that greatly improves the tractability of our algorithm. In the third, we show that our algorithm can be run on real data from Boston, and also that significant ridership gains can be obtained by considering the demand that is not served by the existing MBTA network. These ideas present opportunities for transit authorities to perform holistic re-design of their transit networks in order to offer a service that is both cost-efficient and appealing to commuters.
Chapter 5

Transit Integration with Multiple Providers

In many regions, individual travel needs often extend beyond the service area of a single public transportation agency. As a result, a high percentage of commuters need to make transfers between different transit services. However, when multiple transit services are taken into account for integration, the scale of the networks makes it difficult to manually synchronize the schedules of the various services. We develop data-driven formulations for coordinating between multiple transit operators in multi-modal networks while synchronizing transfers across connecting services and accounting for operator budget concerns. By building a hierarchy of relaxations for different scenarios that balance between realism and tractability, we provide models that can be solved to near-optimality in minutes for city-sized networks, and we demonstrate the benefit of coordination through large-scale case studies on real data from the Boston MBTA network.

5.1 Introduction

In many regions, individual travel needs often extend beyond the service area of a single public transportation agency. As a result, a high percentage of commuters need to make transfers between different services, making the synchronization of transit
schedules of interest to operators and metropolitan planning organizations involved in the integration of transportation providers [77]. Some possibilities include coordinating schedules and operations along major, often congested, corridors to reduce bus bunching at shared stops. Sharing common technologies also allow for trip scheduling across providers or the commingling of riders from different programs.

A first example is the Northwest Transit Alliance, which comprises five transit agencies in Northwest Oregon. Each agency retains ownership of its assets and operations, but they share transit stops and staff resources, and coordinate transfers across regional transit services. A second example is Metro Transit, which operates the services in and around Minneapolis and Saint Paul, working with six smaller transit providers that serve the region’s vast suburbs and provide links to major destinations. Along the major Downtown Marquette and 2nd Avenue Corridors, both urban and suburban transit operators coordinate schedules and follow the same operating procedures to serve designated multi-agency stops. A last example is the Massachusetts Bay Transportation Authority (MBTA), a public agency responsible for operating most public transportation services in Greater Boston, Massachusetts. It succeeds the Metropolitan Transit Authority, which was formed as the integration of independently owned and operated public transportation services.

The potential advantages afforded by optimization are meaningful when there is scope for the effective sharing of resources and coordination of services. There have been templates for transit planners similar to those in [32], but they are difficult to perform manually and result in suboptimal timetables for large networks. Therefore, a full coordination of operations and services to meet these travel needs and service delivery challenges is often the exception. However, most optimization models are unable to scale to transit networks of larger sizes and it remains unclear how to develop heuristics that scale up to solve for transit networks of practical interest, while providing optimality guarantees.

We use real transit networks and realistic origin-destination demand as our starting point, and provide data-driven formulations for transit networks with hundreds of services over thousands of locations. Our work makes the following contributions:
1. We formulate models that address key issues of *coordination*, *regularity* and *operating budget* in transit scheduling. Coordination involves service timetables that reduce waiting times for commuters who transfer between different services. Regularity ensures that service runs are not spaced too far apart from each other, and budget recognizes the labor and financial constraints that agencies operate under.

2. We provide a way of solving the shortest path assignment problem at scale using contraction hierarchies. This resulted in two orders of magnitude savings in computational runtime and memory requirements, allowing us to scale up the problem to optimize over millions of possible origin-destination pairs.

3. Our models provide high quality solutions that are close to optimal within minutes for smaller networks and within hours for larger networks. To our knowledge, this is the first time we are solving a bi-level optimization scheduling problem accounting for passenger assignment at such scale. We provide some insights through a case study on the MBTA network in the Greater Boston area.

The rest of the chapter is outlined as follows. In Section 5.2, we provide an overview of the literature. In Section 5.3, we provide four data-driven models for handling synchronization and congestion, under a unified framework that balances between accuracy and tractability. In Section 5.4, computational experiments show that our models are able to generate high quality solutions within minutes. In a large-scale case study, we provide insights on generating late night schedules for the MBTA network spanning thousands of locations. Finally, we offer concluding remarks in Section 5.5.

### 5.2 Literature Review

For a general overview of the literature on schedule-based transit modelling, see [172]. [88] provided an exact approach for scheduling trains on a single line. This was gen-
eralized to multiple lines in [55]. Although earlier work have been focused on scheduling for freight transportation on railways and had considerations of minimizing cost [99, 53], there has been an increasing interest in “passenger oriented” transit scheduling tailored towards minimizing waiting times for commuters in metro networks. A major difficulty with such models is their lack of tractability: [48] developed an optimization model and heuristic solutions for designing a bus-bridging system, and provided results for a set of 48 lines operating on 17 stops. [158] uses an exact approach, but were able to run experiments on a single line with 15 stations. [170] scaled it up to larger networks by using genetic algorithms, but are unable to provide any optimality guarantees.

When optimizing over multiple services, transit schedules need to be coordinated. In the deterministic case, [173, 92] looked into models for minimizing “missed trains” through the synchronization of time differences. These models are difficult to scale for larger networks, and so [96] uses local search. In the stochastic case, [100] took a stochastic programming approach for the cyclic timetabling problem in which the waiting times are averaged over multiple scenarios weighted by their probabilities. [108] provided a theoretical explanation of why buffer times should be distributed asymmetrically and proposed ways of incorporating robustness in integer programs. However, their approach was not applicable to large transit networks. The notions of light robustness [67] and recoverable robustness [107] were subsequently introduced to provide tractable ways of modelling slack into timetables. Nonetheless, all of them are applicable only when the nominal problem is tractable.

Transit scheduling optimization is based on models of how commuters travel through the network, and is related to the equilibrium assignment model, which describes how commuters would choose their routes through the transit network under a given schedule. This has been studied by [102, 140, 136], and falls under a broader study of transit user choice [110], and related studies in equilibrium assignment models [176, 84]. The user-equilibrium in traffic assignment problem can be compactly formulated using variational inequalities [153]. More recently, [49] extended this characterization to account for congestion, and provided a heuristic for doing so in [50].
Nonetheless, these equilibrium assignment models remain only as a descriptive tool for understanding the potential impact of proposed changes [155]. Bi-level optimization for transit schedules where commuters behave according to complex equilibrium models have not been shown to scale for networks that span hundreds of services and thousands of locations.

5.3 An Integrated Approach to Transit Scheduling

In this section, we develop a few models for scheduling services by each operator in a coordinated fashion. As the transit network will increase in size with multiple operators, we consider data-driven models that balance between complexity and tractability. A key connection between the models is based on the commuting demand and its interaction with the transit services. If commuting demand is low, then it is less important to account for capacity constraints. If services are frequent, then it is less important to synchronize between services.

We begin with a simple model in Section 5.3.1 that generates regular timetables. We then add the complexities of sharing segments between multiple lines (Section 5.3.2), synchronizing timetables (Section 5.3.3) and accounting for congestion (Section 5.3.4). Finally, we discuss the relationships between the models in Section 5.3.5.

5.3.1 Regularity in Timetable Scheduling

We consider $L$ transit operators who have to schedule transit services for the time period from 1 to $T$. The problem is dynamic as the demand for transit services over different origin-destination pairs is time-varying, and the schedules need not be periodic in nature. These decisions are made on the basis of data $d_t^{u,\ell}$ denoting the number of commuters along line $\ell$ entering stop $u$ at time $t$. We provide the full list of notation in Table 5.1.
Table 5.1: Summary of the Notation.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>The number of lines.</td>
</tr>
<tr>
<td>$S$</td>
<td>The number of segments.</td>
</tr>
<tr>
<td>$T$</td>
<td>The number of time periods (where each time period is one minute).</td>
</tr>
<tr>
<td>$B_\ell$</td>
<td>The number of available vehicles for line $\ell$.</td>
</tr>
<tr>
<td>$K_{u,s}^{\text{platform}}$</td>
<td>The capacity (number of commuters) of stop $u$ along segment/line $s$.</td>
</tr>
<tr>
<td>$K_{s}^{\text{service}}$</td>
<td>The capacity (number of commuters) of vehicles along segment/line $s$.</td>
</tr>
<tr>
<td>$[N]$</td>
<td>The set ${1, \ldots, N}$.</td>
</tr>
<tr>
<td>$u \leq s v$</td>
<td>Stop $u$ precedes stop $v$ on segment $s$.</td>
</tr>
<tr>
<td>$\delta(s,w)$</td>
<td>The time (minutes) it takes to travel to stop $w$ on segment $s$.</td>
</tr>
<tr>
<td>$\text{dests}(s,w)$</td>
<td>The set of all possible destinations from stop $w$ on segment $s$.</td>
</tr>
<tr>
<td>$\text{stops}(s)$</td>
<td>The set of all stops on segment and/or line $s$.</td>
</tr>
<tr>
<td>$\text{offset}(\ell,s)$</td>
<td>The time (minutes) it takes to travel from line $\ell$ to segment $s$.</td>
</tr>
<tr>
<td>$\text{validlines}(\ell,t)$</td>
<td>The set of lines that can reach segment $s$ in $t$ minutes.</td>
</tr>
<tr>
<td>$\text{eta}(s)$</td>
<td>The earliest service time beginning on segment/line $s$.</td>
</tr>
<tr>
<td>$\text{ltd}(s)$</td>
<td>The latest service time beginning on segment/line $s$.</td>
</tr>
<tr>
<td>$\text{trange}(s)$</td>
<td>The set of possible service times for segment/line $s$.</td>
</tr>
<tr>
<td>$\text{seg}(u,v,i)$</td>
<td>The segment used to travel from $\text{xfr}(u,v,i)$ to $\text{xfr}(u,v,i+1)$.</td>
</tr>
<tr>
<td>$\text{legs}(u,v)$</td>
<td>The (ordered) list of segments for the commute from $u$ to $v$.</td>
</tr>
<tr>
<td>$\text{xfr}(u,v,i)$</td>
<td>The $i$-th transfer stop on the commute from $u$ to $v$. That is</td>
</tr>
<tr>
<td>$\text{commuters}(u,v)$</td>
<td>The set of tuples $(u',v',i')$ corresponding to commuters travelling from $u'$ to $v'$ who are travelling from $u$ to $v$ on the $i'$-th stage of their commute.</td>
</tr>
</tbody>
</table>

Transit schedules by the operators are made through decisions $x^\ell_t$ where

$$x^\ell_t = \begin{cases} 
1, & \text{if a service starts on line } \ell \text{ at time } t, \\
0, & \text{otherwise.}
\end{cases}$$

For simplicity, we begin by assuming that the services can accommodate all boarding commuters, before modelling service capacity in Section 5.3.4. If the operators for each line $\ell \in [L]$ operate their services without interactions with the others, this will amount to solving $L$ independent optimization problems where operator $\ell$ minimizes the total waiting time on line $\ell$ by solving

$$\min_{x^\ell, z^\ell} \sum_{u \in \text{stops}(\ell)} \sum_{\tau = \text{eta}(\ell)} \sum_{t = \text{eta}(\ell)} \tau d_{u,t}^{\ell} - z_{u,t}^{\ell}$$

subject to

$$\sum_{t=1}^{T} x^\ell_t \leq B_\ell$$

(5.1)
\[ l \leq \sum_{t=0}^{w} x_{\tau+t}^\ell \leq u \quad \forall \ell \in [L], \forall \tau \in [T-w] \]  
(5.3)

\[ \sum_{t=\text{eta}(\ell)}^{\tau} z_{u,\tau}^{u,\ell} \leq \sum_{t=\text{eta}(\ell)}^{\tau} d_{\ell+\delta(u,\ell)}^{u,\ell} \quad \forall u \in \text{stops}(\ell), \forall \tau \in \text{trange}(\ell) \]  
(5.4)

\[ z_{u,\tau}^{u,\ell} \leq K_{u,\ell}^{\text{platform}} x_{t}^\ell \quad \forall u \in \text{stops}(\ell), \forall t \in \text{trange}(\ell) \]  
(5.5)

\[ x_{t}^\ell \in \{0,1\} \quad \forall t \in [T] \]

\[ z_{u,\tau}^{u,\ell} \geq 0 \quad \forall u \in \text{stops}(\ell), \forall \tau \in \text{trange}(\ell) \]

where \( z_{t}^{u,\ell} \) are auxiliary variables for the number of commuters boarding line \( \ell \) on stop \( u \) at time \( t \).

The objective (5.1) is to minimize the number of people waiting over all time \( \tau \) during the time period from \( \text{eta}(\ell) \) to \( \text{ltd}(\ell) \). The number of commuters arriving at stop \( u \) is given by \( \sum_{t=\text{eta}(\ell)}^{\tau} d_{\ell+\delta(u,\ell)}^{u,\ell} \), and the number of commuters who have boarded is given by \( \sum_{t=\text{eta}(\ell)}^{\tau} z_{u,\tau}^{u,\ell} \). Therefore, summing up their difference over all stops \( u \) in \( \text{stops}(\ell) \) would give us the objective. For each line \( \ell \), the schedule \( x_{1}^\ell, \ldots, x_{T}^\ell \) is constrained by the budget \( B_{\ell} \) in (5.2). Constraint (5.4) ensures that the number of commuters who have boarded a train cannot be larger than the demand that has arrived up to that point. Finally, (5.5) enforces that commuters can only board if a service has arrived at that point in time.

The formulation can result in irregular schedules that do not provide a consistent commuter experience. If the service times are evenly spaced, commuters will not have to wait overly long in the worst case. On the other hand, a strict enforcement of the schedules for different lines to be exactly periodic might be overly restrictive and result in infeasibility. To balance between them, we introduce the \textit{regularity} constraint (5.3) to enforce that the schedules for any given line should have at least \( l \) runs starting in any window of length \( w \) and at most \( u \) runs in any window of the same length. The constraint can be ignored if we set \( w = T \) and \( u = B_{\ell} \). (Although we kept it simple, the parameters \( l \), \( u \) and \( w \) can be different for each line.) Therefore, transit planners can adjust \( l \), \( u \) and \( w \) to allow for varying levels of flexibility in coordinating their schedules.
5.3.2 Sharing Segments across Multiple Lines

If multiple operators might share any segments in the same transit network (see Figure 5-1b), then their schedules have to be coordinated in a single optimization model that sets the schedule for all operators. This is complicated for several reasons. First, there might be common resources that have to be shared between different transit services. For instance, train tracks will have to be shared between merging lines in railway networks, requiring the corresponding headways for the trains to be synchronized. Second, the transit operators have to account not only for the commuters who arrive between time $1$ to $T$, but also those commuters arriving before time $1$ or after time $T$ waiting for transit services scheduled from $1$ to $T$.

For multi-stage commutes $(u, v)$ from origin $u$ to destination $v$, we assume commuters will take a route through the transit network based on a sequence of segments $\text{seg}(u, v, i)$ and transfer stops $\text{xf}r(u, v, i)$ for each leg $i \in \text{legs}(u, v)$ of the commute. We introduce auxiliary variables $y_{s,t}$, which is 1 if a service starts on segment $s$ at time $t$ and 0 otherwise. The model is as follows:

\[
\min_{x, y, z} \sum_{s=1}^{S} \sum_{u \in \text{stops}(s)} \sum_{\tau = \text{eta}(s)} \sum_{t = \text{eta}(s)} \text{ld}(s) d_{t+\delta(s,u)}^{u,s} - z_{u,s}^{u,s}
\]

s.t. \[
\sum_{t=1}^{T} x_{\ell,t} \leq B_{\ell} \quad \forall \ell \in [L]
\]

\[
l \leq \sum_{t=0}^{w} x_{\tau+t}^{\ell} \leq u \quad \forall \ell \in [L], \forall \tau \in [T-w]
\]

\[
y_{\ell}^{s} = \sum_{t=\text{validlines}(s,t)} x_{\ell-\text{offset}(t,s)}^{t} \quad \forall s \in [S], \forall t \in \text{trange}(s)
\]

\[
\sum_{t=\text{eta}(s)}^{\tau} z_{u,s}^{u,s} \leq \sum_{t=\text{eta}(s)}^{\tau} d_{t+\delta(s,u)}^{u,s} \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall \tau \in \text{trange}(s)
\]

\[
0 \leq z_{u,s}^{u,s} \leq K_{u,s}^{\text{platform}} y_{\ell}^{s} \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall t \in \text{trange}(s)
\]

\[
x_{\ell}^{t} \in \{0,1\} \quad \forall \ell \in [L], \forall t \in [T]
\]

\[
y_{\ell}^{s} \in \{0,1\} \quad \forall s \in [S], \forall t \in \text{trange}(s),
\]
If each segment corresponds to exactly one line, there is a direct correspondence between objectives (5.1) and (5.6), and constraints (5.2) and (5.7), (5.3) and (5.8), (5.4) and (5.10), and (5.5) and (5.11). This comes from the decomposition of lines into segments provided through notation in Table 5.1 and illustrated in Figure 5-1. In cases where multiple lines share a common segment, the schedules will have to be coordinated through constraint (5.9). The demand \(d_{t}^{u,s}\) is defined on the segments, so they are shared across lines. The model (5.6)–(5.13) can be written as

\[
\min_{(x,y)\in X(B)} Q^{\text{arrival}}(y, d),
\]

where we optimize over the set of feasible schedules

\[
X(B) = \{(x, y) : \sum_{t=1}^{T} x_{t}^{\ell} \leq B_{t} \quad \forall \ell \in [L],
\]

\[
l \leq \sum_{t=0}^{w} x_{t+\tau}^{\ell} \leq u \quad \forall \ell \in [L], \forall \tau \in [T-w]
\]

\[
y_{t}^{s} = \sum_{\ell \in \text{valid lines}(s,t)} x_{t-\text{offset}(\ell,s)}^{\ell} \quad \forall s \in [S], \forall t \in \text{trange}(s),
\]

\[
x_{t}^{\ell} \in \{0, 1\} \quad \forall \ell \in [L], \forall t \in [T],
\]

\[
y_{t}^{s} \in \{0, 1\} \quad \forall s \in [S], \forall t \in \text{trange}(s),
\]

under the Arrival model

\[
Q^{\text{arrival}}(y, d) = \min_{z} \sum_{s=1}^{S} \sum_{u \in \text{stops}(s)} \sum_{\tau = \text{eta}(s)} \sum_{t = \text{eta}(s)} \lambda_{t}^{u,s} - z_{t}^{u,s}
\]

s.t. \[\sum_{t = \text{eta}(s)} z_{t}^{u,s} \leq \sum_{t = \text{eta}(s)} d_{t+\delta(s,u)}^{u,s} \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall \tau = \text{trange}(s)\]

\[0 \leq z_{t}^{u,s} \leq K_{u,s}^{\text{platform}} y_{t}^{s} \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall t \in \text{trange}(s).\]

For the rest of this chapter, we keep the set of feasible schedules \(X(B)\) fixed, and replace \(Q^{\text{arrival}}\) with alternative models of commuting behavior that accounts for transfers between services in multi-modal commutes and peak period congestion. This
allows for formulations with different trade-offs between model accuracy and computational tractability. We provide a model to account for synchronizing transfers in Section 5.3.3 and two additional models to account for congestion in Section 5.3.4, before providing a unified viewpoint of the different models in Section 5.3.5.

5.3.3 Coordinating Timetables for Transferring Commuters

An implicit assumption earlier is that the times at which commuters arrive at a given stop are known. This is reasonable for services with high frequencies since the impact of a missed connection is kept small, making it easy to forecast travel times. However, it is not the case during off-peak periods, making it important to synchronize schedules for minimizing the number of missed connections for commuters who have to transfer between different services. Therefore, we extend the Arrival model to account for transferring commuters, and optimize based on demand $d_t^{u,v,s}$ for the number of commuters entering stop $u$ along segment $s$ at time $t$ with the intention of travelling to stop $v$. This is achieved by replacing the auxiliary variables $z_t^{u,s}$ with a new set of variables $z_t^{u,v,i}$ denoting the number of commuters from $u$ to $v$ boarding a service for the $i$-th leg of their commute.

Boarding Constraints

We replace the boarding constraints in (5.10) with

$$
\sum_{t = \eta\text{eta}(\text{seg}(u,v,1))}^{\tau} z_t^{u,v,1} \leq \sum_{t = \eta\text{eta}(\text{seg}(u,v,1))}^{\tau} d_t^{u,v,\text{seg}(u,v,1)} + \delta(\text{seg}(u,v,1),u) \tag{5.15}
$$

for all commutes $(u, v)$ and time $\tau$ from $\eta\text{eta}(\text{seg}(u,v,1))$ to $\text{ltd}(\text{seg}(u,v,1))$. This is analogous to (5.10), with the variables $z_t^{u,s}$ and demand $d_t^{u,s}$ being replaced by $z_t^{u,v,s}$ and demand $d_t^{u,v,s}$ to keep track of their destinations, and the segment $s$ being replaced by the first stage of the commute from $u$ to $v$, given by $\text{seg}(u,v,1)$.
Transferring Constraints

For multi-stage commutes \((u, v)\), we enforce similar constraints

\[
\sum_{t = \text{eta}(\text{seg}(u,v,i))}^{\tau} z_{t}^{u,v,i} \leq \sum_{t = \text{eta}(\text{seg}(u,v,i-1))}^{\text{ltd}(u,v,i)} z_{t}^{u,v,i-1}
\]

(5.16)

at each transferring stop for legs \(i = 2, \ldots, |\text{legs}(u,v)|\) and time \(\tau\) in the period from \(\text{eta}(\text{seg}(u,v,i))\) to \(\text{ltd}(\text{seg}(u,v,i))\). Here, we define \(\text{ltd}(u,v,i)\) as the minimum of \(\text{ltd}(\text{seg}(u,v,i-1))\) and the time

\[
\tau - \delta(\text{seg}(u,v,i-1), xfr(u,v,i)) + \delta(\text{seg}(u,v,i), xfr(u,v,i))
\]

that a service run will have to start on segment \(\text{seg}(u,v,i-1)\) to be synchronized with a service run starting on \(\text{seg}(u,v,i)\) at time \(\tau\).

Capacity Constraints

For all segments \(s \in [S]\) and stops \(w \in \text{stops}(s)\), we replace the capacity constraints in (5.11) with

\[
\sum_{\{(u,v,i): \text{seg}(u,v,i)=s, xfr(u,v,i)=w\}} z_{t}^{u,v,i} \leq K_{w,s}^{\text{platform}} y_{t}^{s}
\]

(5.17)

for all time \(t \in \text{trange}(s)\). This generalizes the left-hand side to account for the occupancy on a service run as it is passing through stop \(w\) on segment \(s\) at time \(t\). It is achieved by keeping track of those commutes \((u, v)\) that are transferring into segment \(s\) at stop \(w\) at some stage \(i\) of their itinerary, thereby replacing \(z_{t}^{u,s}\) in (5.11) by \(\sum_{\{(u,v,i): \text{seg}(u,v,i)=s, xfr(u,v,i)=w\}} z_{t}^{u,v,i}\).

Total Waiting Time

Finally, the total waiting time to be minimized is replaced by

\[
\sum_{(u,v)} \sum_{i=1}^{\text{legs}(u,v)} \text{ltd}(\text{seg}(u,v,i)) \sum_{t = \text{eta}(\text{seg}(u,v,i))}^{\text{ltd}(u,v,i)} \sum_{t = \text{eta}(\text{seg}(u,v,i-1))}^{\text{ltd}(u,v,i-1)} z_{t}^{u,v,i-1} - z_{t}^{u,v,i},
\]

(5.18)
where we define $\text{seg}(u, v, 0) := \text{seg}(u, v, 1)$ and $\text{ltd}_r(u, v, 1) := \tau$ to unify the expressions across the boarding constraints (5.15) and transfer constraints (5.16).

Putting everything together, we have the Connect Model where $Q_{\text{arrival}}$ is replaced by with $Q_{\text{connect}}$. Therefore, we solve for

$$\min_{(x, y) \in X(B)} Q_{\text{connect}}(y, d),$$

based on the following model of commuting behavior:

$$Q_{\text{connect}}(y, d) = \min_{u,v,i} \sum_{(u,v) i \in \text{legs}(u,v)} \sum_{\tau \in \text{eta}(\text{seg}(u,v,i))} \sum_{t=\text{eta}(\text{seg}(u,v,i-1))} \text{ltd}_r(u,v,i) \text{ltd}_r(u,v,i) z_{u,v,i-1}^{u,v,i}$$

subject to

$$\sum_{t=\text{eta}(\text{seg}(u,v,i))} z_{u,v,i}^{u,v,i} \leq \sum_{t=\text{eta}(\text{seg}(u,v,i-1))} z_{u,v,i}^{u, v, i-1} \forall (u,v), \forall i \in \text{legs}(u,v), \forall \tau \in \text{trange}(\text{seg}(u,v,i))$$

(5.20)

$$z_{t}^{u,v,0} = d_{t+\delta(\text{seg}(u,v,1),u)}^{u,v,\text{seg}(u,v,1)} \forall (u,v), \forall t \in \text{trange}(\text{seg}(u,v,1))$$

(5.21)

$$\sum_{\{(u,v,i): \text{seg}(u,v,i)=s, \text{xfr}(u,v,i)=w\}} z_{t}^{u,v,i} \leq K_{\text{platform},w}^{s} y_{t}^{s} \forall s \in [S], \forall w \in \text{stops}(s), \forall t \in \text{trange}(s)$$

(5.22)

$$z_{t}^{u,v,i} \geq 0 \forall (u,v), \forall i \in \text{legs}(u,v), \forall t \in \text{trange}(\text{seg}(u,v,i)).$$

The objective (5.19) corresponds to (5.18). The boarding constraint (5.20) corresponds to the combination of (5.15) and (5.16). Constraint (5.21) models the auxiliary relationship that the demand corresponds to the zeroth stage of the boarding variables for (5.20). Finally, the capacity constraint (5.22) corresponds to (5.17).

### 5.3.4 Mitigating Congestion in Transit Services

Although the Arrival and Connect model accounts for platform congestion, they do not account for service congestion in popular services. It occurs when transit vehicles are fully occupied, and some commuters have to remain behind for the next available
service, adding to the total waiting time. To handle this, we extend the Arrival and
Connect models with the Segment and Network models respectively. The relationships
between all four models (Arrival, Connect, Segment, Network) will be explained
in Section 5.3.5.

The Segment Model

The Segment model extends the Arrival model to account for the effects of service con-
gestion. It assumes that each commute pair \((u, v)\) lies on a single segment. Therefore,
we can define boarding variables \(z_{t}^{u,v}\) that keeps track of the number of commuters
boarding at time \(t\) on stop \(u\) travelling to stop \(v\), but does not keep track of the leg \(i\)
of the commute that it is on. The formulation is defined as

\[
Q_{\text{segment}}(y, d) = \min_{\mathbf{z}} \sum_{s=1}^{S} \sum_{u \in \text{stops}(s)} \sum_{v \in \text{dests}(s,u)} \sum_{\tau = \text{eta}(s)}^{\text{ld}(s)} \sum_{t = \text{eta}(s)}^{t + \delta(s,u)} d_{t}^{u,v} - z_{t}^{u,v} \quad (5.23)
\]

s.t.

\[
\sum_{t = \text{eta}(s)}^{\tau} z_{t}^{u,v} \leq \sum_{t = \text{eta}(s)}^{\tau} d_{t}^{u,v} + \delta(s,u) \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall v \in \text{dests}(s,u), \forall \tau \in \text{trange}(s) \quad (5.24)
\]

\[
\sum_{v \in \text{dests}(s,u)} z_{t}^{u,v} \leq K_{s}^{\text{platform}} y_{t}^{s} \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall \tau \in \text{trange}(s) \quad (5.25)
\]

\[
\sum_{u \in \text{stops}(s), v \in \text{dests}(s,u) : u \leq s \leq v} z_{t}^{u,v} \leq K_{s}^{\text{service}} y_{t}^{s} \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall \tau \in \text{trange}(s) \quad (5.26)
\]

\[
z_{t}^{u,v} \geq 0 \quad \forall s \in [S], \forall u \in \text{stops}(s), \forall v \in \text{dests}(s,u), \forall \tau \in \text{trange}(s)
\]

The total waiting time (5.23) mirrors (5.27). The boarding constraints (5.24) mirrors
(5.28), the platform capacity constraint (5.25) mirrors (5.30), and the service capacity
constraint (5.26) mirrors (5.31).

The Network Model

In the Network model, we incorporate both considerations of coordinating services
(in Section 5.3.3) and service capacity (in this Section). The formulation is defined
as

\[
Q_{\text{network}}(y, d) = \min_z \sum_{(u, v) \in \text{legs}(u, v)} \sum_{i \in \text{legs}(u, v)} \sum_{\tau \in \text{eta}(\text{seg}(u, v, i))} \text{ltd}(\text{seg}(u, v, i)) \sum_t \sum_{\tau \in \text{eta}(\text{seg}(u, v, i))} z_{t, \tau}^{u, v, i} - z_{t}^{u, v, i}
\]

(5.27)

\[
\text{s.t. } \sum_{t = \text{eta}(\text{seg}(u, v, i))} z_{t}^{u, v, i} \leq \sum_{t = \text{eta}(\text{seg}(u, v, i))} z_{t}^{u, v, i-1} \forall (u, v), \forall i \in \text{legs}(u, v), \forall \tau \in \text{trange}(\text{seg}(u, v, i))
\]

(5.28)

\[
z_{t}^{u, v, 0} = d_{t + \delta(\text{seg}(u, v, 1), u)}^{u, v, \text{seg}(u, v, 1)} \forall (u, v), \forall t \in \text{trange}(\text{seg}(u, v, 1))
\]

(5.29)

\[
\sum_{\{(u, v, i): \text{seg}(u, v, i) = s, \ xfr(u, v, i) = w\}} z_{t}^{u, v, i} \leq K_{\text{platform}}^{w, s} y_{t}^{s} \forall s \in [S], \forall w \in \text{stops}(s), \forall t \in \text{trange}(s)
\]

(5.30)

\[
\sum_{\{(u, v, i): \text{seg}(u, v, i) = s, \ xfr(u, v, i) \leq s, w \leq xfr(u, v, i+1)\}} z_{t}^{u, v, i} \leq K_{s}^{\text{service}} y_{t}^{s} \forall s \in [S], \forall w \in \text{stops}(s), \forall t \in \text{trange}(s)
\]

(5.31)

\[
z_{t}^{u, v, i} \geq 0 \ \forall (u, v), \forall i \in \text{legs}(u, v), \forall t \in \text{trange}(\text{seg}(u, v, i)).
\]

The total waiting time (5.27) is the same as that of the Connect model in (5.19). The boarding constraints (5.28) mirrors (5.20), the demand constraint (5.29) mirrors (5.21), and the platform capacity constraint (5.30) mirrors (5.22). The only difference from the Connect model is the service capacity (5.31). For a service run on segment \(s\) at time \(t\), the occupancy at stop \(w\) is given by the commutes \((u, v)\) transferring into segment \(s\) that are passing through stop \(w\) at some stage \(i\) of their itinerary. This is given by the expression \(\sum_{\{(u, v, i): \text{seg}(u, v, i) = s, \ xfr(u, v, i) \leq s, w \leq xfr(u, v, i+1)\}} z_{t}^{u, v, i}\). To enforce service capacity, we bound it by \(K_{s}^{\text{service}} y_{t}^{s}\), which takes the value \(K_{s}^{\text{service}}\) if there is a service run (i.e. \(y_{t}^{s} = 1\)) and 0 otherwise.
5.3.5 A Unified Perspective for Data-Driven Scheduling

The relationships between all four models (Arrival, Connect, Segment, Network) are summarized in Table 5.2.

<table>
<thead>
<tr>
<th>Schedule coordination</th>
<th>No schedule coordination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Congestion mitigation</td>
<td>$\min_{(x,y)\in X} Q_{\text{network}}^{\text{network}}(y, d)$</td>
</tr>
<tr>
<td>No congestion mitigation</td>
<td>$\min_{(x,y)\in X} Q_{\text{connect}}^{\text{connect}}(y, d)$</td>
</tr>
</tbody>
</table>

Table 5.2: Overview of the four scheduling formulations.

By using a data-driven approach to learn which of these constraints are relevant, we have a way of speeding up optimization of the transit services. We summarize the connections between the four models as follows:

(i) $Q_{\text{network}}^{\text{network}}(y, d) = Q_{\text{segment}}^{\text{segment}}(y, \pi_y^{\text{seg}}(d))$.

(ii) $Q_{\text{network}}^{\text{network}}(y, d) = Q_{\text{connect}}^{\text{connect}}(y, d)$ if capacity constraints are removed.

(iii) $Q_{\text{segment}}^{\text{segment}}(y, d) = Q_{\text{arrival}}^{\text{arrival}}(y, \bar{\pi}^{\text{arr}}(d))$ if capacity constraints are removed.

(iv) $Q_{\text{connect}}^{\text{connect}}(y, d) = Q_{\text{arrival}}^{\text{arrival}}(y, \bar{\pi}^{\text{arr}}(\pi_y^{\text{seg}}(d)))$,

where $\pi^{\text{arr}}$ is a projection from origin-destination (OD) demands with only single-stage commutes into OD demands where each origin has only a single destination. It achieves this by aggregating over all the destinations from that into the destination corresponding to the end of the segment, and is defined as

$$\bar{d}_{tu} = \begin{cases} 
\sum_{v \in \text{dest}(s,u)} d_{tu}^{v} & \text{if } s \in [S], u \in \text{stops}(s), t \in \text{trange}(s), \\
0 & \text{otherwise},
\end{cases}$$

for the projection $\bar{d} = \pi^{\text{arr}}(d)$.

Similarly, for a given schedule $y$, $\pi_y^{\text{seg}}$ is a projection from OD demands that might contain multi-stage commutes into OD demands comprising of single-stage
commuters. It achieves this by aggregating demand over multi-stage commutes that share the same leg, and is defined as

$$d_{u,v}^\mu = \begin{cases} 
\sum_{(\tilde{u}, \tilde{v}, i) \in \text{commuters}(u, v)} d_{\tilde{u}, \tilde{v}} - \Delta_{\tilde{u}, \tilde{v}}(u, y) & \text{if there is a } s \in [S] \text{ such that } u \in \text{stops}(s), \\
v \in \text{dests}(s, u), \text{ and } t \in \text{trange}(s), & \\
0 & \text{otherwise,}
\end{cases}$$

for the projection $\bar{d} = \pi^\text{seg}_y(d)$, where $\Delta_{\tilde{u}, \tilde{v}}(u, y)$ corresponds to the time it takes a commuter traveling from $\tilde{u}$ to $\tilde{v}$ to reach stop $u$ under transit schedule $y$.

For computational reasons, it is preferable to solve the Segment (respectively Arrival) Model in place of the Network (respectively Connect) Model. But the projection $\pi^\text{seg}_y$ creates a co-dependence between the schedule $y$ and the projected demand. To speed up optimization, transit planners can break this dependence by providing a forecast $\hat{d}$ of the projected demand $\pi^\text{seg}_y\ast(d)$ (respectively $\pi^\text{arr}(\pi^\text{seg}_y\ast(d))$) and solving the Segment (respectively Arrival) Model in place of the Network (respectively Connect) Model.

During peak periods, transit services are expected to arrive frequently and regularly, so the projected demand $\pi^\text{seg}_1(d)$ based on a schedule which returns $1^s_t = 1$ regardless of segment $s$ and time $t$ might be a good estimate. During off-peak periods, we might not need to account for congestion. But transit services might arrive more sporadically and irregularly due to low schedule frequency, so it might be prudent for the projected demand to depend on the schedule. All of these factors will affect the performance of the different models. In the next section, we provide a discussion of their merits and trade-offs in a case study of the MBTA network for both peak-period and late night commuters.

### 5.3.6 A Scalable Approach for Transit Passenger Assignment

Among passenger assignment models, the equilibrium assignment [176, 84] and the shortest path assignment are the most common methods. For an overview of different models of assignment models, we refer the interested reader to [110]. Although the
shortest path that commuters take through a network should be dependent on the 
schedules of different services [102, 140, 136], [66] showed that timetable-based routing 
is not always reliable under stochastic travel times. Therefore, we adopt an approach 
of assuming the shortest path based on the average total in-vehicle travel times.

Even with these approximations, the problem of computing the shortest-path 
from each origin to every destination remains a challenging one in large transit 
networks [155]. As there is shared structure between the shortest paths of each 
origin-destination pair, we ran an all pairs shortest paths algorithm [69, 94] which 
takes $O(V^3)$ time, as compared to running a shortest path algorithm for each origin-
destination pair which takes $O(V^3 \log(V))$ time. We developed an approach to solve 
it based on the notion of a transfer graph, which is a compact hierarchical represen-
tation of the transit network. This similar to the method of hierarchies [147, 72] for 
speeding up shortest-path routing, allowing us to scale up the problem to optimize 
over millions of possible origin-destination pairs.

The process of constructing a transfer graph is as follows. We build up a graph 
with nodes corresponding to stops that connect two or more lines, and edges corre-
sponding to segments in the transit network that connects these stops without passing 
through any other transfer stop; the cost of each edge corresponds to the total travel 
time along the corresponding segment. Such a graph is often much smaller than the 
actual network. For example, the MBTA transit network for the Greater Boston 
area has 354 services over 3,833 stops, and corresponds to a network with approxi-
mately 12,000 platforms, and a transfer graph with approximately 1,200 nodes. As 
the All Pairs Shortest Paths (APSP) algorithm runs in $O(V^2)$ space and $O(V^3)$ time, 
this transforms the task from intractable (when $V = 12,000$) to implementable (when $V = 1,200$) in practice.

5.4 Computational Experiments

In this section, we evaluate the performance of our models based on the MBTA 
network in the Greater Boston area. All methods were implemented using the Julia
language [25] and the optimization package JuMP [114] using Gurobi solver v8.1 [82]. Computational experiments for the synthetic data were run on a MacBook Pro with a 2.6 GHz Intel Core i5 processor and 16GB of RAM.

In Section 5.4.1, we show that the data-driven models generate high quality schedules on a small-scale network comprising just the inbound red and orange lines (Section 5.4.1). We show that the data-driven models can speed up solve time by one to two orders of magnitude while maintaining competitive objective values (Section 5.4.1). We also show that we can provide a consistent commuter experience at little penalty to total waiting time (Section 5.4.1).

In Section 5.4.2, we demonstrate the tractability of the Arrival and Connect models in optimizing transit services at scale (Section 5.4.2). To this end, we did a case study based on data from late night transit commuters across both the buses and subway in a multi-modal transit network with hundreds of services over thousands of locations (Section 5.4.2). We show that the Connect model generates reasonable schedules that have a high correspondence with a pilot program ran by the MBTA, and provide some practical recommendations (Section 5.4.2).

5.4.1 Model Comparisons on the MBTA Subway

As the public agency that operates most public transit services in the Greater Boston Area, the MBTA network provides a good case study for the integration of transit services, and consists of both buses and the subway. Buses are typically numbered, and the subway system consists of five intersecting lines called the Red, Orange, Green, Blue, and Silver lines.

Minute-level turnstile (both entrance and exit) data on the subway stations were made available in a visualization project by Michael Barry and Brian Card in 2014. As data on the buses were not available in this dataset, we restricted ourselves to the subway network, and studied how the computational runtimes of the different models scale with transit network size. We focused on the rush-hour from 5 to 6pm, where there is the potential to reduce network congestion when the system is at capacity, and ran a generative model based on the approach in [24] to produce peak-hour
origin-destination demand matrices from 4 to 7pm.

(a) The transit network that we optimize over can be fairly complex, comprising multiple services, one corresponding to each color, each comprising multiple lines. We focus on the north-bound red lines in Figure 5-1b.

(b) The lines running from Ashmont and Braintree to Alewife share a common segment from JFK to Alewife. Therefore, we decompose the two lines into the three segments Ashmont to JFK, Braintree to JFK, and JFK to Alewife.

Figure 5-1: The MBTA subway network.

We compared results across networks of different scales at a time limit of 10,000 seconds. Although the Network formulation is hard to solve to optimality, it is relatively easy to evaluate $Q_{\text{network}}(y, d)$ for a given value of $y$, so we have a way of comparing the quality of the incumbent solutions from the Arrival, Connect and Segment Models. The models were solved to optimality using Gurobi with the parameter $\text{MIPFocus}$ set to 3, and the results are shown in Figure 5-3. The choice of the $\text{MIPFocus}$ parameter value was to allow the solver to spend more time at the root node, and to focus on the best bound, which greatly facilitated the branch and bound process. (Without setting the $\text{MIPFocus}$ parameter, some of the models did not reach optimality within the time limit.)

The inbound red and orange lines.

For the smaller network, we optimize over a subnetwork with the red and orange lines running in the inbound direction. The network comprises 3 services over 4 segments, 40 stops and 51 links. There were 13,966 commuters over 142 origin-destination pairs; 55 of the origin-destination pairs corresponded to multi-leg commutes. This network was chosen as there is a sufficient volume of commuters for congestion to
Figure 5-2: Coordination of Timetables. We built visualizations of Marey diagrams [116] to illustrate the schedules and occupancies of the different services. The lines are colored based on the services that they correspond to, and their width corresponds to the occupancy along each service run. This is an illustration of the best found schedule for the MBTA in-bound red and orange subway network.

occur, transferring commuters between the red and orange lines, and coordination for the red line on the segment between JFK/UMass and Alewife. We used a single budget of 60 trains to be distributed across all lines for the entire planning period, and set the train capacities to 1,000.

As we see in Figure 5-2, all the models coordinates the schedules from Ashmont and BrainTree to Alewife such that there is no clumping of services on the shared segment from JFK/UMass to Alewife illustrated in Figure 5-1b. Across both the red and orange services, the generated schedules manages to capture heterogeneity in the flow of commuters over the time period, with a higher frequency for all services from 5 to 5:30pm and a lower frequency from 5:30 to 6pm. Although there is a net-neutral transfer of commuters between the orange and red lines from 5 to 5:30pm, there is a higher transfer of commuters from the orange to the red line at Downtown Crossing.
Table 5.3: Computational Performance on the Red and Orange Lines. We ran all the models on the inbound red and orange lines with a time limit of 10,000 seconds, and keep track of the number of the top three classes of cuts generated. All but the Network Model terminated at optimality. We report the MIP Gap for each of them, defined as the lower and upper objective bound divided by the absolute value of the upper bound. We also report the time when the Network Model found a better schedule (if applicable).

<table>
<thead>
<tr>
<th>Models</th>
<th>Gomory</th>
<th>MIR</th>
<th>Flow Cover</th>
<th>MIP Gap</th>
<th>Time Overtaken (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival</td>
<td>145</td>
<td>333</td>
<td>2147</td>
<td>30.07%</td>
<td>8319</td>
</tr>
<tr>
<td>Segment</td>
<td>256</td>
<td>783</td>
<td>8589</td>
<td>30.21%</td>
<td>8319</td>
</tr>
<tr>
<td>Connect</td>
<td>707</td>
<td>1573</td>
<td>8921</td>
<td>31.30%</td>
<td>7955</td>
</tr>
<tr>
<td>Network</td>
<td>296</td>
<td>1353</td>
<td>10911</td>
<td>25.80%</td>
<td>N.A.</td>
</tr>
</tbody>
</table>

Table 5.4: Computational Performance on the Full Subway Network. We ran the models on the full MBTA subway network with a time limit of 10,000 seconds, and keep track of the number of the top three classes of cuts generated. All but the Connect Model terminated at optimality.

<table>
<thead>
<tr>
<th>Models</th>
<th>Gomory Cuts</th>
<th>MIR Cuts</th>
<th>Flow Cover Cuts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival</td>
<td>472</td>
<td>416</td>
<td>4339</td>
</tr>
<tr>
<td>Segment</td>
<td>748</td>
<td>781</td>
<td>8369</td>
</tr>
<tr>
<td>Connect</td>
<td>2383</td>
<td>1866</td>
<td>15603</td>
</tr>
</tbody>
</table>

for the runs that begin after 5:30pm.

As it is the peak-hour, the Network and Segment formulations were expected to outperform the Arrival and Connect formulations, since the former two includes capacity constraints that ensure that commuters are able to board the trains. The incumbent solutions for the Arrival, Connect and Segment models were not required to be monotonically decreasing in $Q_{\text{network}}(\cdot, d)$, but the results were still surprising. In particular, the Arrival Model took two minutes to arrive at a schedule that outperformed the Connect and Segment Models, and was only overtaken by the Network Model after more than two hours (see the last column in Table 5.3). This was due to two reasons. Even though it included transfer constraints, the Connect Model wrongly assumed that commuters will always be able to board, resulting in the bunching of commuters into batches that exceeds the train capacity. Secondly, the forecasts of transferring commuters were slightly off, causing the Segment Model to pre-emptively schedule more trains based on wrong predictions of congestion periods.
The full subway network.

We ran the models for the full subway network comprising the red, orange, blue and green lines (see Figure 5-1a). The network comprises 16 services over 20 segments, 113 stops and 310 links. There were 54,874 commuters over 659 origin-destination pairs; 476 of the origin-destination pairs corresponded to multi-leg commutes. We used a single budget of 300 trains to be distributed across all lines for the entire planning period, and set the train capacities to 1,000 as before. The Network Model failed to generate any solutions or bounds within 10,000 seconds and so we left it out, and used the Segment Model as the basis of comparison instead.

(a) **The in-bound red and orange lines.** All except the Network model terminated at optimality.

(b) **The full subway network.** All except the Connect model terminated at optimality.

Figure 5-3: Comparison of the Model Runtimes. We ran the Network, Connect, Segment and Arrival formulations on two different networks with a time limit of 10,000 seconds and MIPFocus=3. For each, we plot the incumbent solutions based on their objective value under the network formulation and the corresponding time they were generated. We also plot lower bounds (in orange) on the optimal objective value as time proceeds.

All the models made heavy use of lifted flow-cover cuts (see Tables 5.3 and 5.4). As flow-cover inequalities are not facet-defining, [78] developed a computationally efficient class of lifted flow cover inequalities that were derived from [138]. Another class of cuts is based on mixed-integer rounding (MIR) inequalities [81, 118], which are generalizations from the basic flow-cover inequality developed by [138]. Along with Gomory cuts, they are all general-purpose cutting planes for binary mixed integer programs that have accounted for a component of the speedup in mixed-integer linear
solvers on benchmarks [30]. However, our computational experiments show that they are not sufficient for the class of scheduling problems here. Instead, our results show that data-driven formulations that perform well for some scenarios (but not on others) can provide order-of-magnitude speed-ups in optimization.

**Regularity of the transit schedules.**

![Figure 5-4: Computational runtimes with regularity constraints. We ran the Network, Connect, Segment and Arrival Model with regularity constraints on inbound red and orange lines with a time limit of 10,000s and MIPFocus=3. For each, we plot the incumbent solutions based on their objective value under the network formulation and the corresponding time they were generated. We also plot lower bounds (in orange) on the optimal objective value as time proceeds. Finally, we added points to show the MIP start solutions of the Network Model before it started the branch-and-bound process.](image)

The earlier formulations resulted in irregular schedules that do not provide a consistent commuter experience. Therefore, we set $l = 1$, $u = 3$, and $w = 5$ such that we have between 1 to 3 runs for any given window of 5 minutes, and re-ran all the models on the network with inbound red and orange lines (see Figure 5-4). All the models found solutions of similar quality (compared to when they did not have the regularity constraints), indicating that there is negligible penalty (in total waiting time) to pay for enforcing a more consistent commuter experience. Although they do not improve the best bound found by the solver, we observe modest improvements.
(over not having the regularity constraints) in getting solutions of higher quality earlier. For instance, the Network Model was able to find start solutions with average waiting times of less than 1.3 minutes in less than 2,000 seconds with the regularity constraints, whereas it found them only after 6,000 seconds without the constraints. This is helpful for transit planners if the transit network is too large to hope for optimality, and getting solutions of high quality in a timely manner is of greater importance.

5.4.2 A Multi-Modal Large Scale Case Study

Historically, two prior attempts at providing late-night services sustainably have failed. The “Night Owl” bus service of 2001 to 2005 used bus routes to replicate subway lines on weekends and eventually saw minuscule ridership. Late-night service on both bus and subway lines from 2014 to 2016 led to maintenance concerns and eventually ceased due to a budget deficit. In August 2018, the MBTA started a $1.2 million pilot of a system of public transit buses to provide late night workers an affordable way to get home, expanding bus services between 10pm and 3am on a number of routes in Boston and the surrounding neighborhoods, providing more frequent trips and extended services for lines 15, 34E, 66, 93, 104, 108, 109, 111, 116, 117, 442, SL1, and SL4. In this section, we optimize services based on data available from the past, and compare them with the services being offered by the MBTA in 2018.

**Demand and network characteristics.**

We used data on late night transit commuters for both the buses and subway that was made available at https://github.com/MassBigData/LateNightT in the Late Night T Data Challenge organized by MassBigData in 2014. The data comprises transactions recorded by MBTA fare gates, buses, and light rail vehicles after 10pm on Fridays and Saturdays from March 2013 to June 2014. We filtered out stop locations that were outside of our network of coverage, and restricted ourselves to those stops.
for which we have postcode information. We then ran an inference algorithm to generate an origin-destination matrix based on conditional probabilities (based on the hour of the day and the postcode of the stop) estimated from historical data. We focused on the time period from 10pm to 11pm. The demand matrix is highly sparse and has a total of 17,535 commuters across 2.74 million origin-destination pairs, with 2.56 million of the pairs corresponding to multi-leg commutes.

**Solving the transit passenger assignment problem at scale.**

The network itself comprises 354 services over 358 segments, 3,833 stops and 11,584 links. In Section 5.3.6, we discussed the computational challenges in solving the transit passenger assignment problem at scale. Although this is performed as a pre-processing step prior to solving the model itself, it turns out to take a significant component of the total time as well. To give a sense of the computational requirements: the original approach on a network with 12,000 nodes took more than a day and still did not terminate. Moreover, it took 1.15GB of memory just to store the distance matrix, which scales as $O(V^2)$ where $V$ is the number of nodes in the transit network. By contrast, our approach using the transfer graph completed in less than 15 minutes, and just 11.5MB of memory to store the distance matrix.

**Model construction and solution times.**

We scaled the cost of operating a bus to 30% that of a train for the budget constraint, and set `MIPFocus=1` and a time limit of 10,000 seconds to prioritize finding high quality schedules rather than improving the best bounds for optimality. It took about 10 minutes for the transit network to be constructed from Floyd Warshall’s algorithm (see Section 5.3.6), and another 10 minutes to construct the Connect Model from the network, resulting in a model with approximately 1.6 million variables and 2 million constraints. Only the Arrival and Connect Models produced good schedules within a reasonable amount of time (see Figure 5-5). As there is not much congestion in off-peak late night commutes, and the Segment and Network Models were too slow (the construction of the service capacity constraints resulted in cumbersome expressions
Figure 5-5: Computational runtimes for the Late Night T network. We ran the Connect Model (in blue) and Arrival Model (in purple) models on the full MBTA network including both buses and trains. For each, we plot the incumbent solutions based on their objective value under the connect formulation and the corresponding time they were generated. We also plot the best bounds (in orange) on the optimal objective value as time proceeds.

corresponding to the occupancies of the vehicles), we left them out of consideration.

Generating practical recommendations.

The Arrival Model performed worse compared to the Connect Model (see Figure 5-5), generating schedules with an average waiting time of 1.7 minutes compared to the Connect Model’s average waiting time of 1.1 minutes. Therefore, we used the schedules generated by the Connect Model for comparison with the MBTA schedule. We found a strong correspondence between the two schedules, suggesting the potentiality for data-driven optimization approaches to supplement human level insights on the impact of different routes. The schedule we generated was in agreement with the MBTA schedule in operating services 15, 66, 111, 116, 117 and SL1. However, it did not capture services 34E, 93, 104, 108, 109, 442 and SL4, which could be due to the overly stringent budget that we set. On the other hand, it brought up a few routes which are not currently being operated past 10pm, such as services 17 and 747.
illustrated in Figures C-1a and C-1b respectively.

5.5 Conclusion

In this work, we provide methods for coordinating and synchronizing schedules across multiple providers in large scale transit networks. We present four models that balances between accuracy and tractability in minimizing the total waiting time of commuters over all platforms in a data-driven approach while accounting for congestion on vehicles and platforms. We discuss how to scale up the transit assignment model to handle thousands of locations using contraction hierarchies, and present computational experiments to show that our models produce high quality solutions in practical amounts of time. We provide a way for transit operators to balance in maintaining evenly-spaced and regular timetables, while synchronizing schedules across different services. Finally, we perform a challenging case study on late night services for the MBTA network, and generate some insights for potential services to be considered.
Appendix A

Appendix to Chapter 2

A.1 Node-Arc Adjacency Matrix Example

For the following graph:

![Graph Diagram]

The corresponding sub-matrices are:

$$B_I = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \\ i_1 & 1 & 1 & 0 \\ i_2 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad B_J = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \\ j_1 & -1 & 0 & 0 & 0 \\ j_2 & 0 & -1 & -1 & 0 \\ j_3 & 0 & 0 & 0 & -1 \end{bmatrix}$$

A.2 Proofs of Propositions

Proof. Proposition 1. Fix $d = (d_j)_{j \in J}$. By definition, we have for some $\epsilon > 0$:

$$\gamma_j^{\text{single}}(\alpha) = \text{V@R}_\alpha(-\text{Poisson}(\hat{\gamma}_j^{\text{single}} + \epsilon)) \to \infty \quad \text{as} \quad \alpha \to 0 \quad \forall j \in J,$$

$$\gamma_j^{\text{local}}(\alpha) = \text{V@R}_\alpha(-\text{Poisson}(\hat{\gamma}_j^{\text{local}} + \epsilon)) \to \infty \quad \text{as} \quad \alpha \to 0 \quad \forall j \in J,$$
\[ \gamma_{\text{regional}}(\alpha) = V@R_\alpha(-\text{Poisson}(\hat{\gamma}_{\text{regional}} + \epsilon)) \to \infty \text{ as } \alpha \to 0 \ \forall i \in I, \]
\[ \gamma_{\text{global}}(\alpha) = V@R_\alpha(-\text{Poisson}(\hat{\gamma}_{\text{global}} + \epsilon)) \to \infty \text{ as } \alpha \to 0. \]

Therefore, let
\[ \alpha_{j}^{\text{single}} \text{ small enough such that } \gamma_{j}^{\text{single}}(\alpha_{j}^{\text{single}}) \geq d_j \ \forall j \in J, \]
\[ \alpha_{j}^{\text{local}} \text{ small enough such that } \gamma_{j}^{\text{local}}(\alpha_{j}^{\text{local}}) \geq \sum_{k \in \delta_j} d_k \ \forall j \in J, \]
\[ \alpha_{i}^{\text{regional}} \text{ small enough such that } \gamma_{i}^{\text{regional}}(\alpha_{i}^{\text{regional}}) \geq \sum_{j \in \delta_i} d_j \ \forall i \in I, \]
\[ \alpha^{\text{global}} \text{ small enough such that } \gamma_{\text{global}}(\alpha^{\text{global}}) \geq \sum_{j \in J} d_j. \]

and let \( \alpha = \min\{\min_{j \in J}\{\min\{\alpha_{j}^{\text{single}}, \alpha_{j}^{\text{local}}\}\}, \min_{i \in I}\{\alpha_{i}^{\text{regional}}\}, \alpha^{\text{global}}\} \). By construction, we can see that \( d \) satisfies all of the constraints in the uncertainty set
\[ \mathbb{D}(\alpha) = \left\{ d \in \mathbb{Z}^{|J|}_+ \mid d_j \leq \gamma_{j}^{\text{single}}(\alpha) \ \forall j \in J, \right. \]
\[ \left. \sum_{k \in \delta_j} d_k \leq \gamma_{j}^{\text{local}}(\alpha) \ \forall j \in J, \right. \]
\[ \sum_{j \in \delta_i} d_j \leq \gamma_{i}^{\text{regional}}(\alpha) \ \forall i \in I, \]
\[ \sum_{j \in J} d_j \leq \gamma_{\text{global}}(\alpha) \right\}. \]

and therefore \( d \in \mathbb{D}(\alpha) \).

\[ \text{Proof. Proposition 2. Observe that } \mathbf{B} = \begin{bmatrix} \mathbf{B}_I \\ \mathbf{B}_J \end{bmatrix} \text{ is a totally unimodular matrix, and therefore} \]
\[ Q_\phi(x, d) = \min_y \phi^\top y \]
\[ \text{s.t. } \begin{bmatrix} \mathbf{B}_I \\ \mathbf{B}_J \end{bmatrix} y \leq \begin{bmatrix} x \\ -d \end{bmatrix} \]

150
always has an integral optimal solution whenever the right-hand side \( \begin{bmatrix} x \\ -d \end{bmatrix} \) is integer (which is the case for all \( x \in \mathbb{Z}_+^{|I|} \) and \( d \in \mathbb{Z}_+^{|J|} \)). Therefore the integer constraints can be relaxed:

\[
Q_\phi(x, d) = Q_{\phi}^{LP}(x, d) = \min_y \phi^\top y \\
\text{s.t.} \begin{bmatrix} B_I \\ B_J \end{bmatrix} y \leq \begin{bmatrix} x \\ -d \end{bmatrix} \\
y \in \mathbb{R}_+^{|E|}
\]

Taking the dual of \( Q_{\phi}^{LP}(x, d) \), we get

\[
\max_{p, q} x^\top q - d^\top p \\
\text{s.t.} \quad q^\top B_I + p^\top B_J \leq \phi^\top \\
p \leq 0, q \leq 0
\]

Finally, \( y = 0 \) is always a feasible solution to (2.1) for all \( x \in \mathbb{Z}_+^{|I|} \) and \( d \in \mathbb{Z}_+^{|J|} \), with an objective function value of 0. Therefore, \( Q_\phi(x, d) \) is finite, and strong duality holds for all \( x \in X \) and \( d \in D(\alpha) \).

**Proof.** Proposition 3. Since \( D(\alpha) \) is finite for any fixed \( \alpha > 0 \), we have convergence of the C&CG algorithm in at most \(|D(\alpha)|\) iterations by the following result:

**Proposition 4** ([180]). Let \( p \) be the number of extreme points of \( D(\alpha) \) if it is a polyhedron or the cardinality of \( D(\alpha) \) if it is a finite discrete set. Then, the C&CG algorithm will converge to the optimal value in \( O(p) \) iterations, where \( p \) is the number of extreme points of \( D(\alpha) \) if it is a polyhedron or the cardinality of \( D(\alpha) \) if it is a discrete set.

The proof for proposition (4) can be found in [180].
A.3 Scenario Generation in the CCG algorithm

Figure A-1: Scenarios generated by the CCG algorithm. This is based on the robust deployment model for 35 ambulances, with parameter $\alpha = 0.01$. We illustrate the first 16 scenarios generated as chloropleths, and corresponding deployment plans as orange circles, from left to right, top to bottom.
### A.4 Deployment Plans Generated

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of Ambulances</th>
<th>Location</th>
<th>Number of Ambulances</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>1</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>2</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>2</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>3</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>3</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>4</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>4</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>5</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>5</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>6</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>6</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>7</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>7</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>8</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>8</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>9</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>9</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>10</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>10</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>11</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>11</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>12</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>12</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>13</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>13</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>14</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>14</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>15</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>15</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>16</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>16</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>17</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>17</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>18</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>18</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>19</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>19</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>20</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>20</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>21</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>21</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>22</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>22</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>23</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>23</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>24</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>24</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>25</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>25</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>26</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>26</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>27</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>27</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>28</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>28</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>29</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>29</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>30</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>30</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>31</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>31</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>32</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>32</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>33</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>33</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>34</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>34</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>35</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
<td>35</td>
<td>1 1 1 1 1 1 1 1 1 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of Ambulances</th>
<th>Location</th>
<th>Number of Ambulances</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
<td>40</td>
<td>1 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>41</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>41</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>42</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>42</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>43</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>43</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>44</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>44</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>45</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>45</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>46</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>46</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>47</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>47</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>48</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>48</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>49</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>49</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>50</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td>50</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

**Table A.1:** Robust Deployment Plans. Generated for different parameter values $\alpha$, with varying numbers of ambulances. For values of $\alpha$ above 0.01, the model saturated quickly which resulted in deployment plans that were not as competitive.
Table A.2: Deployment Plans. Generated for each model with varying numbers of ambulances. Both the Stochastic and Robust formulations evenly distribute the allocation of ambulances, but the MEXCLP and MALP clusters them in a few central locations.
A.5 Model Performance Comparisons

<table>
<thead>
<tr>
<th>n</th>
<th>Response (min)</th>
<th>Stochastic</th>
<th>Robust01</th>
<th>Robust005</th>
<th>Robust001</th>
<th>Robust0001</th>
<th>Robust00001</th>
<th>MEACLP</th>
<th>MALP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>15</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>20</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>25</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>30</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>35</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>40</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>45</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
<tr>
<td>50</td>
<td>shortfall (±0.25)</td>
<td>74.30 (0.183)</td>
<td>73.05 (0.178)</td>
<td>72.85 (0.181)</td>
<td>72.56 (0.183)</td>
<td>72.36 (0.185)</td>
<td>72.16 (0.187)</td>
<td>70.64 (0.189)</td>
<td>70.64 (0.189)</td>
</tr>
<tr>
<td></td>
<td>fraction (±0.25)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
<td>0.61 (0.020)</td>
</tr>
</tbody>
</table>

Table A.3: Coverage Peak Steady. Simulated coverages based on 12 replications, each for 360 hours of continuous peak-hour ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored.

Table A.4: Coverage Peak Volatile. Simulated coverages based on 12 replications, each for 360 hours of continuous peak-hour ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored.
### Table A.5: Coverage Off-Peak Steady. Simulated coverages based on 12 replications, for each 360 hours of continuous off-peak ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored.

<table>
<thead>
<tr>
<th>Response (min)</th>
<th>Robust001</th>
<th>Robust005</th>
<th>Robust01</th>
<th>Robust05</th>
<th>Robust10</th>
<th>EMLP</th>
<th>MALP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ± S.D.</td>
<td>15.1 ± 1.0</td>
<td>14.9 ± 1.0</td>
<td>14.4 ± 1.0</td>
<td>14.0 ± 1.0</td>
<td>13.5 ± 1.0</td>
<td>13.0 ± 1.0</td>
<td>12.5 ± 1.0</td>
</tr>
<tr>
<td>Median ± IQR</td>
<td>15.0 ± 0.9</td>
<td>14.8 ± 0.9</td>
<td>14.4 ± 0.9</td>
<td>14.0 ± 0.9</td>
<td>13.5 ± 0.9</td>
<td>13.0 ± 0.9</td>
<td>12.5 ± 0.9</td>
</tr>
<tr>
<td>Fraction (%)</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
</tr>
<tr>
<td>Lower 95% CI</td>
<td>14.9 ± 0.9</td>
<td>14.8 ± 0.9</td>
<td>14.4 ± 0.9</td>
<td>14.0 ± 0.9</td>
<td>13.5 ± 0.9</td>
<td>13.0 ± 0.9</td>
<td>12.5 ± 0.9</td>
</tr>
<tr>
<td>Upper 95% CI</td>
<td>15.3 ± 1.0</td>
<td>15.0 ± 1.0</td>
<td>14.6 ± 1.0</td>
<td>14.2 ± 1.0</td>
<td>13.7 ± 1.0</td>
<td>13.2 ± 1.0</td>
<td>12.7 ± 1.0</td>
</tr>
</tbody>
</table>

### Table A.6: Coverage Off-Peak Volatile. Simulated coverages based on 12 replications, for each 360 hours of continuous off-peak ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored.

<table>
<thead>
<tr>
<th>Response (min)</th>
<th>Robust001</th>
<th>Robust005</th>
<th>Robust01</th>
<th>Robust05</th>
<th>Robust10</th>
<th>EMLP</th>
<th>MALP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ± S.D.</td>
<td>15.1 ± 1.0</td>
<td>14.9 ± 1.0</td>
<td>14.4 ± 1.0</td>
<td>14.0 ± 1.0</td>
<td>13.5 ± 1.0</td>
<td>13.0 ± 1.0</td>
<td>12.5 ± 1.0</td>
</tr>
<tr>
<td>Median ± IQR</td>
<td>15.0 ± 0.9</td>
<td>14.8 ± 0.9</td>
<td>14.4 ± 0.9</td>
<td>14.0 ± 0.9</td>
<td>13.5 ± 0.9</td>
<td>13.0 ± 0.9</td>
<td>12.5 ± 0.9</td>
</tr>
<tr>
<td>Fraction (%)</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
</tr>
<tr>
<td>Lower 95% CI</td>
<td>14.9 ± 0.9</td>
<td>14.8 ± 0.9</td>
<td>14.4 ± 0.9</td>
<td>14.0 ± 0.9</td>
<td>13.5 ± 0.9</td>
<td>13.0 ± 0.9</td>
<td>12.5 ± 0.9</td>
</tr>
<tr>
<td>Upper 95% CI</td>
<td>15.3 ± 1.0</td>
<td>15.0 ± 1.0</td>
<td>14.6 ± 1.0</td>
<td>14.2 ± 1.0</td>
<td>13.7 ± 1.0</td>
<td>13.2 ± 1.0</td>
<td>12.7 ± 1.0</td>
</tr>
</tbody>
</table>

### Table A.7: Response Peak Steady. Simulated response times based on 12 replications, for each 360 hours of continuous peak-hour ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored.

<table>
<thead>
<tr>
<th>Response (min)</th>
<th>Robust001</th>
<th>Robust005</th>
<th>Robust01</th>
<th>Robust05</th>
<th>Robust10</th>
<th>EMLP</th>
<th>MALP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ± S.D.</td>
<td>15.1 ± 1.0</td>
<td>14.9 ± 1.0</td>
<td>14.4 ± 1.0</td>
<td>14.0 ± 1.0</td>
<td>13.5 ± 1.0</td>
<td>13.0 ± 1.0</td>
<td>12.5 ± 1.0</td>
</tr>
<tr>
<td>Median ± IQR</td>
<td>15.0 ± 0.9</td>
<td>14.8 ± 0.9</td>
<td>14.4 ± 0.9</td>
<td>14.0 ± 0.9</td>
<td>13.5 ± 0.9</td>
<td>13.0 ± 0.9</td>
<td>12.5 ± 0.9</td>
</tr>
<tr>
<td>Fraction (%)</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
<td>0.95 ± 0.02</td>
</tr>
<tr>
<td>Lower 95% CI</td>
<td>14.9 ± 0.9</td>
<td>14.8 ± 0.9</td>
<td>14.4 ± 0.9</td>
<td>14.0 ± 0.9</td>
<td>13.5 ± 0.9</td>
<td>13.0 ± 0.9</td>
<td>12.5 ± 0.9</td>
</tr>
<tr>
<td>Upper 95% CI</td>
<td>15.3 ± 1.0</td>
<td>15.0 ± 1.0</td>
<td>14.6 ± 1.0</td>
<td>14.2 ± 1.0</td>
<td>13.7 ± 1.0</td>
<td>13.2 ± 1.0</td>
<td>12.7 ± 1.0</td>
</tr>
</tbody>
</table>

156
### Table A.8: Response Peak Volatile
Simulated response times based on 12 replications, each for 360 hours of continuous peak-hour ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored.

<table>
<thead>
<tr>
<th>Report (min)</th>
<th>Stockholms</th>
<th>Robotlon1</th>
<th>Robotlon05</th>
<th>Robotlon10</th>
<th>Robotlon100</th>
<th>MEX005-LP</th>
<th>MEX01-LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>mean (2.98)</td>
<td>3.89 (0.10)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
</tbody>
</table>

### Table A.9: Response Off-Peak Steady
Simulated response times based on 12 replications, each for 360 hours of continuous off-peak ambulance operations, with steady turnaround times. Shaded cells are results that should be ignored.

<table>
<thead>
<tr>
<th>Report (min)</th>
<th>Stockholms</th>
<th>Robotlon1</th>
<th>Robotlon05</th>
<th>Robotlon10</th>
<th>Robotlon100</th>
<th>MEX005-LP</th>
<th>MEX01-LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>mean (2.98)</td>
<td>3.89 (0.10)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
<tr>
<td></td>
<td>mean (2.98)</td>
<td>3.94 (0.10)</td>
<td>24.92 (0.99)</td>
<td>21.77 (1.39)</td>
<td>21.90 (1.40)</td>
<td>22.00 (1.41)</td>
<td>21.77 (1.39)</td>
</tr>
</tbody>
</table>

### Table A.10: Response Off-Peak Volatile
Simulated response times based on 12 replications, each for 360 hours of continuous off-peak ambulance operations, with volatile turnaround times. Shaded cells are results that should be ignored.

157
Appendix B

Appendix to Chapter 3

B.1 Input Parameter Estimation

The MNL model requires estimation of the $\beta$ coefficients that represent the commuter’s valuation of time and money, cf. equation (3.15). Although estimation of commuter utility functions is not the focus of this paper, we estimated $\beta$ coefficients in a reasonable and intuitive way, guided by survey data and basic economic principles [105].

For ease of presentation, the model outlined in Section 3.4 assumes a homogeneous population with a shared utility function. However, generalizing to multiple segments of the population that are each described by a different utility function is straightforward, and it is the model we use in our computational experiments. In this section, we will take $\beta$ to be a $G \times 2$ vector, where $G$ is the number of commuter segments, and $\beta_{g1}$ and $\beta_{g2}$ represent the values of time and money for commuter segment $g$, respectively. The commuter segments are taken to be the different income levels in census surveys.

To estimate reasonable $\beta$ values, we propose the following relationships. First, for simplicity, we assume that $\beta_{1}^{q}$ and $\beta_{3}^{q}$, the marginal utilities for time and comfort, are the same for all income groups: extra time or comfort makes everyone equally happy, but they vary in their ability and willingness to pay for it. We arbitrarily set $\beta_{1}^{q} = 1 \forall g = 1, \ldots, G$. The value of money, $\beta_{2}^{q}$, should clearly vary with income, so
we propose the relationship

\[
\frac{\beta_g^2}{\beta_g^1} = \frac{y_g'}{y_g},
\]

(B.1)

where \(y_g\) represents the income of income group \(g\). Equation (B.1) indicates that someone with $10,000 of income experiences 10 times the incremental utility for an extra dollar as compared to someone with $100,000.

To relate \(\beta_g^1\) and \(\beta_g^2\), we turn to a travel survey by [133]. When survey-takers were asked to list the most important factors in their route choice, 81.3% answered that time was important, 31.5% answered that cost was important, and 24.7% answered that comfort was important. The numbers do not add up to 100% because survey-takers were allowed to list multiple important factors. There were additional factors of reliability, safety, and emissions that could influence the route choice, but since those are exogenous to the model, we focus solely on the time and cost factors. The answers to the travel survey motivated us to weight the utility coefficients accordingly as follows:

\[
\sum_{g=1}^{G} \pi_g \beta_g^2 \sum_{g=1}^{G} \pi_g \beta_g^1 = \frac{81.3}{31.5},
\]

(B.2)

where \(\pi_g\) represents the proportion of the city population that is at income level \(g\).

Similarly, we took the marginal value of comfort to be

\[
\frac{24.7}{31.5} = 0.784.
\]

Equations (B.1) and (B.2) are together \(G\) equations in \(G\) unknowns (recalling that we have set \(\beta_1^1 = 1\)). Boston-area income levels and distributions were obtained from the publicly-available U.S. Census [165], and Tokyo-area data was obtained from a Japanese survey [156]. Solving equations (B.1) and (B.2) produced a range of \(\beta_g^2\) values roughly between 0.5 and 15, which we used in the computational experiments that follow. Again, the purpose of this section was only to discuss a reasonable way of generating input parameters, and we do not claim that these values are exact.
Appendix C

Appendix to Chapter 5

Figure C-1: Some suggestions for the MBTA Late Night T. Both images taken from https://www.mbta.com/schedules/. Map data ©2019 Google.
Figure C-2: Recommended Late Night Services for the MBTA Network. A map of the recommended MBTA services based on optimization using the Late Night T data. All the subway lines (red, blue, orange and green) are maintained. Among the bus services, the ones that are maintained are plotted in purple, and the ones to be left out of late night services are plotted in grey. Map tiles by Stamen Design, under CC By 3.0. Data by OpenStreetMap, under CC BY SA.
Bibliography


172
[135] Agostino Nuzzolo, Umberto Crisalli, and Francesca Gangemi. A behavioural


