INTEGRATED MICROSIMULATION MODELLING OF CROWD AND SUBWAY NETWORK DYNAMICS FOR DISRUPTION MANAGEMENT SUPPORT

by

Sivakrishna Srikukenthiran

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Civil Engineering
University of Toronto

© Copyright 2015 by Sivakrishna Srikukenthiran
Abstract

Integrated Microsimulation Modelling of Crowd and Subway Network Dynamics For Disruption Management Support

Sivakrishna Srikukenthiran
Doctor of Philosophy
Graduate Department of Civil Engineering
University of Toronto
2015

In many large cities around the world, public transit networks have been carrying an ever-increasing burden of commuters. This has resulted in large movements of crowds in major transit hubs, and high levels of congestion in systems running near or at capacity. In such situations, service disruptions can negatively impact service and transit users well after they are resolved.

Currently, transit agencies handle these disruption situations in an ad-hoc fashion. This is due to the lack of a tool capable of analyzing the network-level impacts of response strategies, especially with performance sufficient to handle large-scale systems. Existing tools are limited in their ability to accurately simulate all dynamic facets of the transit network, or model the behaviour of passengers following disruption events. As a result, they do not adequately capture the two-way interaction between crowds and mass transit service, or the complexities of train operation.

This dissertation presents Nexus, an innovative crowd dynamics and transit network simulation platform, that enables full simulation of all actors in the transit system and integrated dynamic transit assignment, while being both flexible and scalable to handle large-scale networks. Instead of developing new simulators for surface transit, trains and stations, Nexus enables interfacing of existing simulators together to form a network. An agent-based framework was overlaid to allow for responsive agents, while a virtual communication system permitted on-the-fly modifications to transit service operation.

With crowd behaviour in stations key to network performance, new models were constructed to better explain passenger behaviour at two critical locations. First, discrete choice models were developed of passenger choice of stairs versus escalators. Second, a simulation-based model of passenger behaviour on train platforms was developed using a diffusion-inspired approach and accounting for preferred waiting locations. Field data collected across several Toronto subway stations informed both models.

Finally, a proof-of-concept case study was conducted on the Toronto transit network. The impact of disruptions of various lengths at a key platform in the network was examined, and an illustrative example was conducted to show how the system could be used to test strategies to reduce impact on transit passengers.
Acknowledgements

Completing a doctorate degree is a long and challenging journey, and I could not have completed it without the assistance and encouragement of many people throughout the last several years. I would like to thank them all for allowing me to reach this point. My appreciation goes beyond those who I have highlighted here who played particularly prominent roles, to everyone who I have had the pleasure of getting to know who made this trip much more enjoyable.

To start, I would like to thank my thesis advisor, Prof. Amer Shalaby, for both his support and guidance throughout my degree. His mentorship was invaluable, both in providing direction for the research and giving insightful feedback. I would also like to thank him for the confidence he displayed in me both in accepting a student with a slightly unorthodox background, and for championing the work to extend beyond my degree. Knowing that the work I was doing would not just stay on the shelf afterwards provided great motivation throughout.

I would also like to extended my thanks to the rest of my examination committee, Prof. Khandker Nurul Habib and Prof. Matthew Roorda for the insight they provided in their feedback to my thesis, and, particularly Prof. Habib for allowing me to tap his expertise in choice modelling. I would also like to thank my external examiner, Prof. Hani Mahmassani, for his insightful critique, and taking the time to be a part of my examination.

I was fortunate in having the assistance of two undergraduate students that helped to greatly speed up my work. To start, I would like to thank Carl Zhan for the time he spent processing pedestrian counts and his Excel wizardry in visually presenting timed counts at many locations. Next, I would like to send a special thanks to David King for the many hours spent processing video data for the vertical circulation study, and for the construction of the Bloor/Yonge station simulation model.

It is quite common for doctoral students to thank their lab for general support, but in my case, I had a steady stream of students who played an important role in this thesis. Pedestrian research requires a lot of data that is labour intensive to collect, and for the three data collection efforts that were conducted, I never had a problem getting my colleagues to help out. For that, I would like to thank the dozens of students from the ITS Lab I have known over the years for not only their moral support, but also for their many hours counting and tracking people in subway stations.

Next, I would like to thank everyone at the Toronto office of Arup, who I was lucky to have partnered with for some key areas in my thesis. In addition to providing funding to better understand pedestrian behaviour, they allowed for me to avoid having to code my own pedestrian simulation software, and provided the access to interface their software and legitimize the platform I developed. Specifically, I would like to thank Erin Morrow, Hilary Holden, Daniel Fisher, Micah Zarnke and Ian Mackenzie for their support and expertise, and, even though I was not an employee of Arup, for making me feel like I was.

I would also like to thank Bernard Farrol and Peter Janas at the Toronto Transit Commission for providing additional data that was critical to the development of the pedestrian and station models.

Last, but not least, I would like to thank my family: my parents and sister, for their love and support throughout my life, but especially in the last several years. Without them, I would not have gone down this route or been able to complete my degree.
## Contents

1 Introduction ................................................. 1  
  1.1 Background and Motivation .......................... 1  
  1.2 Thesis Objectives .................................... 3  
  1.3 Thesis Approach ..................................... 3  
    1.3.1 Nexus Development .............................. 3  
    1.3.2 Crowd Modelling .................................. 4  
    1.3.3 Case Study of the Toronto Subway Network .. 4  
  1.4 Thesis Outline ..................................... 5  

2 Literature Review .......................................... 6  
  2.1 Introduction ......................................... 6  
  2.2 Modelling of Large Scale Urban Transportation Systems ............................ 6  
    2.2.1 Distributed and Parallel Computing ................. 7  
  2.3 Rail Simulation ...................................... 8  
  2.4 Station and Pedestrian Simulation ....................... 9  
    2.4.1 Statistical Models ................................ 10  
    2.4.2 Mesoscopic Models of Pedestrian Movement ...... 11  
    2.4.3 Microscopic Models of Pedestrian Movement ..... 11  
  2.5 Existing Methods of Transit Assignment ................ 16  
  2.6 Disruption Management of Transit Networks .............. 17  
  2.7 Moving Forward ................................ 19  

3 System Frameworks ........................................ 20  
  3.1 Research Framework .................................. 20  
  3.2 Designing a Crowd & Transit Network Simulation Framework .................. 21  
  3.3 Simulator Requirements ................................ 22  
  3.4 Simulation Framework ................................ 23  
    3.4.1 Components ...................................... 24  
    3.4.2 Data Structures .................................. 31  
  3.5 Agent Framework ..................................... 32  
  3.6 Communication Framework ................................ 33  
  3.7 Conclusion ........................................ 35
4 Modelling Vertical Circulation Choice

4.1 Current Literature ................................................. 38
4.2 Data Collection Process ........................................... 39
  4.2.1 Toronto Transit Commission (TTC) Network ................. 39
  4.2.2 The Data Collection Plan ..................................... 40
4.3 Aggregate Models .................................................. 41
  4.3.1 Data Extraction Process ...................................... 42
  4.3.2 Explanatory Variables Considered for Model Estimation .... 42
  4.3.3 Modelling Framework ........................................... 43
  4.3.4 Model Estimation ............................................... 44
  4.3.5 Analysis ......................................................... 46
  4.3.6 Model Validation ................................................ 48
4.4 Importance of Dynamic Conditions on Choice ....................... 50
  4.4.1 Modelling Framework ........................................... 50
  4.4.2 Data Processing and Qualitative Observations ............... 52
  4.4.3 Method of Analysis and Model Estimation .................... 53
  4.4.4 Analysis ......................................................... 55
4.5 Binary and Mixed-Logit Models .................................... 57
  4.5.1 Motivation ....................................................... 57
  4.5.2 Data Extraction ................................................ 57
  4.5.3 Data Processing ................................................ 58
  4.5.4 Model Framework .............................................. 59
  4.5.5 Model Estimation ............................................... 60
  4.5.6 Parameter Analysis ............................................ 62
  4.5.7 Validation ....................................................... 62
4.6 Application .......................................................... 67
4.7 Conclusions .......................................................... 68

5 Modelling Pedestrian Distribution Along Platforms .................. 70

  5.1 Current Literature ................................................. 70
  5.2 Model Structure .................................................... 72
    5.2.1 Local Movement (Kinematic) Model ......................... 72
    5.2.2 Active Targeting Model ...................................... 76
    5.2.3 The Overall Platform Model .................................. 78
  5.3 Data Collection and Processing ................................... 79
    5.3.1 Pedestrian Distribution Field Data ......................... 79
    5.3.2 Platform Section Targeting Data ............................ 82
  5.4 Model Estimation .................................................. 85
    5.4.1 Genetic Algorithms ............................................ 85
    5.4.2 Parameters for Estimation .................................... 87
    5.4.3 Initial GA Agent Generation ................................. 87
    5.4.4 Platform Simulator ............................................ 87
    5.4.5 Evaluating Simulation Results: Fitness Function Definition ... 88
    5.4.6 Subsequent Population Generation Creation and Convergence ... 90
8.6 Base Model Results .......................................................... 133
  8.6.1 Visualization of Results ........................................... 133
  8.6.2 The Base Model ....................................................... 133
  8.6.3 Characterizing the Base Model .................................... 134
  8.6.4 Impact of Incorporating the Developed Pedestrian Models .... 136
8.7 Analyzing the Impact of Disruptions .................................. 139
  8.7.1 Specifying the Disruption ......................................... 139
  8.7.2 Disruption Results .................................................. 140
8.8 Testing Disruption Management Strategies .......................... 140
8.9 Summary ................................................................. 145

9 Summary and Conclusions ................................................ 146
  9.1 Summary ................................................................. 146
  9.2 Conclusions ............................................................ 148
  9.3 Key Contributions ...................................................... 151
  9.4 Future Research Directions .......................................... 153
    9.4.1 Improvements to Component Simulators ....................... 153
    9.4.2 Transit User Behaviour .......................................... 154
    9.4.3 Transit Service Rescheduling During Disruptions .......... 154
    9.4.4 Validation of Greater Toronto transit network ............ 155
    9.4.5 Applications ..................................................... 155

Bibliography ................................................................. 156
List of Tables

2.1 Overview of Commercial Transportation Network Simulation Software ............... 7

4.1 Summary of Vertical Circulation Facilities Observed ........................................ 41
4.2 Summary of Final Aggregate Vertical Circulation Model Estimation ...................... 44
4.3 Summary of vertical Circulation Facilities Observed ......................................... 53
4.4 Summary of Latent Class Vertical Circulation Model Estimation ......................... 54
4.5 Summary of Final Disaggregate Vertical Circulation Model Parameter Estimation .... 61
4.6 Summary of Final Disaggregate Vertical Circulation Model Settings and Goodness-of-Fits 62
4.7 Vertical Circulation Discrete Choice Model Validation Results .......................... 67

5.1 Platform Data Collection Locations ................................................................. 80
5.2 Platform Dispersion Model Estimation Results .................................................. 91

8.1 Characteristics of Transit Agencies Modelled .................................................... 122
8.2 Info on GTFS Data used for Modelling Transit Service .................................... 123
# List of Figures

3.1 Research framework .............................................. 21
3.2 Simulation framework services schematic ...................... 23
3.3 Example process model for running a simulation using the framework .......... 25
3.4 Simulation framework components and communication flow .................. 26
3.5 External and internal interfaces for the coordination Server .................. 27
3.6 Data structures defining transit network service .......................... 32
3.7 Data structures defining transfer of agent information ..................... 33
3.8 Stages of control signalling ......................................... 34
4.1 Observation locations along the TTC subway network and two example locations ................. 40
4.2 Prediction error of estimated aggregate vertical circulation models (ascending direction with (a) and without (b) PRM individuals, and descending direction with (c) and without (d) PRM individuals) for 10 second segment choice splits. ............ 49
4.3 Ability of aggregate models in predicting facility use for train alighting pulses ....................... 50
4.4 Latent-class model flow diagram. .................................... 51
4.5 TTC station observation locations at St. George (left) and Bloor/Yonge (right). .............. 52
4.6 Predictive accuracies of estimated latent class and binary logit models ................. 56
4.7 Predictive accuracy of individual stair or escalator choice for the estimated vertical circulation discrete choice models. ................................. 65
5.1 Overall platform movement model schematic ........................... 73
5.2 Plot of motivation function for pedestrians targeting a platform section for a $\gamma$ of 0.25 ........ 78
5.3 Observation locations, dividing platform into 8 sections ......................... 81
5.4 Situations faced during platform data processing. ................................ 82
5.5 TTC transfer points between Yonge-University-Spadina and Bloor-Danforth ............ 84
5.6 Platform model estimation process ..................................... 86
5.7 Platform model simulation steps ....................................... 89
5.8 Effect of estimated values of $\gamma$ and $\alpha$ on target motivation and entrance influence. ...... 92
5.9 Targeting probability as a function of the spread of pedestrian entrance volume ............ 93
5.10 Performance of the peak period model with agents permitted to target locations, showing overall boarding distribution deviations .............................. 94
5.11 Performance of the peak period model with agents only diffusing, showing overall boarding distribution deviations .............................. 94
5.12 Performance of the off-peak period model with agents only diffusing, showing overall boarding distribution deviations .................................................. 95
5.13 Performance of the peak period model with agent targeting in predicting max-volume sections ......................................................................................... 96

6.1 Process model for running a simulation for the stage I system .................. 99
6.2 Schematic of the small-scale test network .............................................. 103
6.3 Schematic of the large-scale test network .............................................. 104
6.4 Scalability in population size and number of computers for stage I system .................................................. 105
6.5 Improving simulation performance with increasing number of computers for stage I system .................................................. 105
6.6 Diminishing returns with increasing synchronization periods for stage I system .................................................. 106
6.7 Vehicle dwell time distributions for the large stage I network .............. 107
6.8 Effect of line control on network performance for the large stage I network .................................................. 108

7.1 Transit assignment path finding algorithm ........................................ 112
7.2 Handling of agents and vehicles for the MassMotion interfacing application .................................................. 117
7.3 User interface of network analyzer .................................................. 118

8.1 Distribution of origin and destination points for the case study .............. 126
8.2 Rail networks of the Greater Toronto case study .................................. 128
8.3 Schematic of the TTC subway network highlighting St George and Bloor/Yonge stations .................................................. 129
8.4 MassMotion model of Bloor/Yonge station ........................................ 131
8.5 MassMotion model of St. George station ........................................ 132
8.6 Average TTC station volume for base model ................................... 135
8.7 Average TTC platform volume for base model ................................... 135
8.8 TTC trains total passenger load for base model ................................. 136
8.9 Agent experience for base model .................................................. 136
8.10 Performance of the Yonge-University-Spadina subway line for the base model .................................................. 137
8.11 Impact of vertical circulation choice model on platform crowding ....... 138
8.12 Impact of incorporating the platform model on network performance .... 139
8.13 Impact of various duration disruptions at Bloor/Yonge station ............ 141
8.14 Impact of response strategy on platform counts ................................ 143
8.15 Impact of response strategy on agent experience ................................ 144
Chapter 1

Introduction

1.1 Background and Motivation

Public transit authorities are considering a multitude of investments and strategies, some long term while others more immediate, to combat the ever growing congestion across transit networks and deteriorating quality of service. The day-to-day functioning of transit systems is even more challenged by recurring service disruptions of varying magnitudes and occasional threats of major emergencies. This is especially true in systems that are currently running at capacity with short headways [1], where the problem of knock-on effects are well known but difficult to predict [2]. The selection, prioritization and optimization of transit improvement strategies and interventions require a thorough analysis and evaluation using a high-fidelity modelling system. Currently, however, agencies instead rely on ad-hoc methods when responding to sudden disruptions. This is because of the lack of such an analytical tool to model or analyze the network-level impacts of response strategies, especially with performance high enough to handle large scale systems in a reasonable time frame.

Key to such a system is a proper representation of the flow of passengers throughout the network, which would necessitate sufficient modelling of passenger movements within stations to account for the two-way interaction between crowd behaviour and transit service operation. The consideration of crowd movements when analyzing the performance of facilities and spaces has been gaining attention in recent years. This has been enabled by a new class of simulation software which has emerged in the last decade that specifically model pedestrians, allowing for the analysis of crowd movements in complex spaces. While the body of research in this area has grown significantly in recent years, many questions remain on the various facets of pedestrian navigation and movement. Much of the focus to date has been in understanding motion at a local level in order to reproduce the various types of flows possible in open spaces and passageways. Relatively little research, however, has been done in better understanding the process taken by pedestrians for overall navigation, particularly by transit users in stations, which has the potential to impact transit service in high volume situations. Similar limitations exist in the ability of traditional transit assignment models to adequately replicate passenger path choice through complex multi-modal transit networks, especially if operating under congested and overcrowded conditions or where the network has been subjected to an unexpected disruption. Recent advances in path choice models and crowd simulation methods have recognized such limitations and made some progress in overcoming them; however, these advances have occurred in isolation from one another.
A high-fidelity modelling system usable for disruption management support needs to both model transit supply in detail while accounting for the behaviour of passengers at both global and local levels; however, among the existing methods of transit network modelling, no single approach meet these requirements. The current approaches are generally either macroscopic or micro-simulation based. On the macro-modelling side (e.g. EMME, VISUM), an abstract (graph) representation of the network is used, with aggregate demand models used to model network flow in order to perform transit assignment. On the other side, traditional micro-simulation software like Aimsun, VISSIM or PARAMICS are at-their-core traffic simulators, and do a relatively poor job modelling transit service and demand, particularly rail and the behaviour of transit users, lacking the capability to perform transit assignment. In both cases, the effect of crowds on transit performance has been traditionally ignored. This has recently begun to change with some micro-simulation software beginning to incorporate pedestrian movements. But as they are constrained by their auto-focussed origins, pedestrians are not agents that travel through the transit network from origin to destination; instead, they are introduced into localized spaces, like stations or intersections, using arrival rates or counts, and removed after completing their walk.

Away from these network-level approaches, when precisely modelling passenger and train movements, both researchers and transit practitioners have fallen back on mainly evaluating operational policies by treating them as isolated from the transit network. Analysis of the impact on crowd flow due to crowd control or structural changes, are performed without directly accounting for how these changes might affect train service; alternatively, evaluation of modifications to train movement, for example from new signalling systems, do not consider the resulting implications on crowds within stations. This has originated both from traditional divides within transit agencies, where train operation and station management are handled by essentially separate entities, and similar siloing in the research community, where pedestrian and train modelling are handled by different research groups. The aforementioned relative newness of the pedestrian modelling field has added to this current paradigm with researchers focussed on understanding the basics of crowd motion, with how they interact with transit vehicles not yet having attracted significant focus.

This type of isolated analysis of transit network components (stations and lines), while much simpler and requiring significantly less effort, can lead to uncertain results. In situations of low passenger flow and low frequency train service, with sufficient buffer for schedule recovery, this may be appropriate, where modifications to either service is unlikely to have significant knock-on effects to other parts of the network. For the environment examined in this thesis, however, with high frequency mass transit operating near capacity and significant pre-existing congestion at stations, the likelihood of perturbations to crowd behaviour in stations and train movements remaining independent from the network is remote. There is a well researched link between boarding/alighting distribution and train dwell times, as well as an understanding that train delays tend to propagate throughout train networks. As a result, failing to consider these transit system components in the context of their network lends to a high probability of ignoring feedback effects and producing incorrect results of line capacity and station passenger flow. This is particularly true where the study period constitutes a significant length of time (e.g entire morning peak period or day).

As a result, there remains a significant gap for a dynamic modelling system with the capability to handle, in detail, both transit supply and passenger behaviour, and the flexibility to handle unexpected changes to the transit network. This leaves open great potential for a system that can combine micro-simulation modelling of transit supply with advances in crowd dynamics and passenger behaviour.
modelling to address current limitations that prevent detailed analysis of complex issues in transit operations research and disruption management.

1.2 Thesis Objectives

This dissertation aimed to bridge this gap by developing Nexus, an interconnection platform for crowd and transit simulators, that allows for building a sophisticated microsimulation model of the transit network with detailed passenger movements in key areas. The overall goal was to allow for the examination of network effects of delays throughout the entire transit service due to pedestrian congestion or unforeseen disruptions. This would allow for the testing of mitigation strategies for disruption management, and provide a framework for the future implementation of real-time decision support after unexpected situations. A secondary goal of Nexus was to enable transit research requiring system-wide simulation, including testing for any network-level impact of changes at a local level. In achieving these goals, this dissertation targeted satisfying the following key objectives:

- Development of Nexus as a modular, flexible and scalable system to allow for network wide analysis of passenger movement through transit networks and simulate ITS deployment, disruption events, and response measures
- Development of pedestrian models for key bottlenecks within stations: level changes and mass transit platforms
- Development of the Toronto Subway Network within Nexus, analyzing the network impact of disruptions and illustrating its ability to test simple response strategies

1.3 Thesis Approach

In attempting to reach these goals, this thesis followed two parallel tracks, one dealing with development of Nexus, and a second dealing with some key crowd modelling questions. Convergence of these two tracks, through integration of the developed models, resulted in the final prototype platform, upon which a case study was conducted.

1.3.1 Nexus Development

Development of Nexus first involved creating a set of frameworks to inform its computing, agent and communication architecture. The creation of the computing framework consisted of research into the types of computing architectures, what was currently possible with available programming tools, and the existing state of urban simulators and their weaknesses in simulating large and dynamic systems. This resulted in a distributed services-oriented approach that enables the use of existing specialized commercial simulators in order to build the broader network. The agent framework focussed mainly on the needs of dynamic transit assignment methods, including how to best structure and locate the decision making process of the agents, the expected process of decision making for agents throughout their trip, and the necessary data to transfer to maintain consistent agent behaviour and characteristics. Finally, the communication framework focussed on how feedback on transit service and agent experience occurred, and the method of actuating change on both service and agents dynamically during simulation.
Following framework creation, implementation was tackled in two stages. The first stage involved building the base components in a simplified form with a focus on synchronizing the distributed components, and ensuring basic viability of the approach specified in the designed frameworks. To start, the implementation was tested against small and medium sized theoretical networks. The second stage looked to move this basic implementation towards one that could handle the large-scale nature of the real-world case study. This included programming tools to easily import network structure from publicly accessible files, optimizing performance of certain tasks (like network routing) to a viable level of computation speed, automatic visualization of results for debugging and reporting purposes, and incorporating detailed crowd simulation of key stations by interfacing the commercial 3D pedestrian simulation software, MassMotion.

1.3.2 Crowd Modelling

With crowd dynamics being a critical component of Nexus and the overall dissertation, the crowd modelling track focussed on improving our understanding of pedestrian and crowd behaviour at key locations in subway stations. This was performed with an expectation that local behaviour at bottlenecks either within components (stations, surface or rail), and particularly at the interfaces between adjacent components would have the potential to have impact beyond affecting the local environment. It was also conducted to provide illustration of the value of Nexus, as a base upon which to combine transit and pedestrian research over multiple projects. In both areas of development detailed below, models were incorporated or adapted for use within Nexus where possible.

The first area of research was to better model the behavioural process of pedestrians in transit stations when deciding between stair or escalator mode at level transitions. This research had two main stages. The first stage aimed to develop an aggregate model with the goal of predicting flow splits between co-located stair-escalator facilities under a variety of conditions. The second stage examined individual choice, developing and estimating a set of discrete choice models for implementation within the MassMotion software. Also of interest was the impact on network performance from a modelling change at a local level.

The second area of research, involved developing a behavioural framework for modelling passenger dispersion along platforms, and estimating the associated models. In contrast to prior research in the field, this framework took the approach of incorporating the overall network path choice of agents to introduce preferred waiting locations. Borrowing from the field of particle diffusion, the developed model also accounted for the timing of the entrance of pedestrians relative to train arrival and the resulting distribution of passengers at the time of boarding. In addition to incorporating the tendency of commuter populations to develop habits when waiting for their trains, the goal was to provide boarding distributions endogenously rather than the exogenous input required in existing simulators. Given that boarding (and the resulting alighting) distributions play a key role in setting the dwell times of trains, it was believed that an improved understanding of the phenomena would be central to the workings of the overall simulator in correctly modelling subway performance.

1.3.3 Case Study of the Toronto Subway Network

Finally, to illustrate proof-of-concept of Nexus, a real world large-scale network was implemented within the software. The network chosen was the Toronto subway network, given its proximity and known
issues of congestion, with two of the most utilized stations (Bloor/Yonge and St. George) modelled fully within MassMotion. As the network is highly integrated with surface routes and has significant connections to neighbouring transit agencies and the regional rail system, these adjacent networks were also modelled where public data on their structure and service characteristics were available. An agent population was synthesized to be representative of the AM peak period, with data sourced from the 2011/2012 Transportation Tomorrow Survey. As a calibrated transit assignment process was outside the scope of this thesis, the platform was evaluated on its ability to provide reasonable and expected results over common disruption scenarios and modifications to service operation. The developed case study was also intended to be the first step towards a complete model of the Toronto region’s transit networks.

1.4 Thesis Outline

To provide the details of the approach taken in reaching the objectives, this thesis has been divided up into several chapters. First, a literature review (Chapter 2) is presented, providing an overview of current urban transportation simulators, simulation specific to each component of the transit network (surface, rail and station) and transit assignment. Also covered are studies and developments dealing with the current state of disruption management for rail systems. This is followed by a description of the frameworks developed to allow for network-level analysis of crowd movement and transit service (Chapter 3).

Before the prototype implementation of Nexus and case study, work on two key areas of pedestrian behavioural models are presented. The first, is an examination of vertical circulation choice (Chapter 4). The chapter presents a set of models, both for easier application at an aggregate level (10-sec flows) and for incorporation into a microsimulator; this latter application was tested within MassMotion. The second passenger model developed was one to predict the dispersion of passengers along a train platform, considering both the volume and positioning of entering flows and the tendency of commuter passengers to have preferred waiting locations (Chapter 5).

The development of the Nexus prototype is then presented, divided into two stages. Stage I (Chapter 6), the initial software implementation, was developed with a focus on developing the main coordination engine and proving the viability of the system. To enable implementation of the case study GTA network, modifications were required of this initial prototype. These changes are explained in Stage II (Chapter 7). They included some structural and process changes to greater increase performance, necessary to handle the much larger network, and creation of the interface wrapper to allow for MassMotion to connect with Nexus and perform the task of detailed crowd simulation in key stations. The methods used to incorporate the two developed passenger models are also described.

The main case study illustrating proof-of-concept of Nexus in handling large-scale crowd and transit simulation is dealt with in Chapter 8. An overview of the GTA network, with a focus on the TTC subway network, is provided, along with how the network structure was imported and the network built using the Nexus specifications. Data sources for transit service and user population, and any processing of this data, are also detailed. Lastly, analysis of the base system results, and the results of several disruption scenarios and simple response strategies is presented.

Finally, the last chapter presents overall conclusions and lists the key contributions of this dissertation. The thesis is concluded with future directions of research and development leading from the presented work.
Chapter 2  

Literature Review  

2.1 Introduction

The Nexus platform structure, as one linking pre-existing transportation simulation software, is unique in both the commercial and research worlds. Nevertheless, there exists urban transportation simulators that are similar in some of the goals and capabilities of Nexus, and these similarities and differences are highlighted in this chapter. With improved modelling of mass transit and crowd movements being the focus, the chapter also contains a more expansive review of models of rail and pedestrian simulation, highlighting major dedicated commercial and research software, and existing methods of transit assignment. Finally, existing research on disruption management of transit network is presented. This includes an overview of the types of disruptions that occur, how response is currently approached in the field, and the various research methods under study.

2.2 Modelling of Large Scale Urban Transportation Systems

The modelling of large scale urban transportation systems has generally fallen into two categories: simplified results using graph theory, and full microsimulation of the entire network, vehicles and agents.

Methods of analysis of transit networks using a graph theory approach are contained within the area of complex network analysis. These types of analysis are statistical in nature, taking into account network characteristics like the distribution of the degree of connectivity of nodes, clustering coefficients and network size in comparing networks and deducing how they might perform [3]. Analysis of transportation networks using these methods is a newer occurrence, with the original application areas being in communication network, biological systems and social networks [4]. Public transport networks are the main target of complex network analysis, with all found to be examples of small-world networks, networks with the property of low direct connectivity between most nodes, but a low number of hops to travel between them [4]. Studies using the methodology have been conducted around the world, with a particular focus on subway networks, including systems in Europe, Asia and North and South America [5, 4, 6, 3]. Many of these studies have mainly acted just to characterize the network, calculating common metrics and distributions, to confirm these public transport networks as complex networks and identify similarities in these calculated values. Some have gone beyond to examine how best to evolve these networks, identifying ideal growth points [6], or how these network characteristics lend to grad-
ing network resilience [7, 5]. Nevertheless, while these methods allow for a computationally efficiency first-pass examination, their analysis is limited to network structure and basic examination of passenger flows. More sophisticated analysis necessitates more detailed simulation methods.

There are currently several urban microsimulation packages commercially available, all of which began their existence as pure traffic simulators. Their focus, however, has been on modelling surface transit, where interactions can occur between transit vehicles and cars. With the emerging importance of pedestrian simulation in recent years, some leading network simulators have incorporated pedestrian modelling within their software allowing for the analysis of how pedestrians and vehicles interact, often partnering with existing standalone software. In addition, while focus remains mainly on surface transit, some simulators, like PTVs VISSIM have added some limited capability in simulating mass transit and the influence of pedestrians on dwell time during boarding and alighting.

These simulators, however, remain rooted in their origins in moving vehicles around a network rather than people, and as such have taken the approach of introducing transit passengers into spaces to understand flow at a local level. Pedestrians simulated in this fashion have no awareness of network changes or the trip they are making, instead being created and destroyed as they enter and exit the space, respectively. The simulator does not perform transit assignment; instead, specific boarding volumes or alighting percentages are specified for each stop, and origin/destination matrices are specified for local spaces to define pedestrian volumes. A summary of their abilities is presented in the Table 2.1.

Table 2.1: Overview of Commercial Transportation Network Simulation Software

<table>
<thead>
<tr>
<th>Software</th>
<th>Pedestrian Handling</th>
<th>Transit Handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paramics</td>
<td>Uses a continuous space agent-based model with pedestrians taking paths based on least effort, while avoiding collision. Pedestrians have low intelligence and can get stuck with poor spatial design requiring manual intervention</td>
<td>Models surface transit (bus, LRT) specifying physical characteristics, schedule and dwell times; pedestrians do not influence dwell times</td>
</tr>
<tr>
<td>UAF with Myriad II</td>
<td>Based on the social forces concept with dynamic routing</td>
<td>Similar to Paramics but dwell time influenced by passengers</td>
</tr>
<tr>
<td>VISSIM</td>
<td>Simplified social forces using a pre-calculated potential field defined for each pedestrian group as the driving force of motion</td>
<td>Simulates rolling stock for boarding/alighting Integrates with Open-Track for simulated train schedules</td>
</tr>
<tr>
<td>SimWalk Transport</td>
<td>Agent-based in continuous space moving based on least effort. Pedestrian behaviour based on data collected in similar locations/context</td>
<td>Focused on surface transit, allowing for bus/LRT physical characteristics, schedule and dwell times</td>
</tr>
<tr>
<td>Aimsun with Legion</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.2.1 Distributed and Parallel Computing

In adding detailed pedestrian movement to network simulators, performance can be sluggish, especially for larger crowds. While some progress has been made by adjusting how collision detection is performed [8], or harnessing multi-core processing in the case of the MassMotion simulation software [9], the simulation of large numbers of people cannot be accomplished on a single computer at speeds necessary for time-critical applications. In the more mature area of traffic simulation, however, the concept of
Chapter 2. Literature Review

Spreading simulations over multiple computers has been attempted, either through the use of clustered computing or over a local area network [10, 11]. In either case, this requires partitioning of the network into cells, distributing the calculation of vehicle movement for each cell across computers, and having a mechanism to deal with transitions of vehicles as they cross cell boundaries [10, 11]. While this may be possible in traffic simulations where vehicle flow is relatively low and trajectories can be more easily predicted to aid in the transition process, it is not feasible when simulating large crowds in 2D or 3D environments. As a result, a different approach is taken for the framework and prototype presented in this thesis, only allowing flows of agents between components at key interfaces.

2.3 Rail Simulation

Proper simulation of train movement involves dealing with train physical and operational characteristics for appropriate movement, modelling the signalling and dispatching systems, and scheduling [12]. Train movement dynamics are determined by traction of train wheels against the tracks, and relevant resistances [13]. Based on the train specifications, a speed curve can be generated to be used to model movement from station to station [13]. Braking distance is normally defined as the distance a train requires to come to a stop plus a safety amount, preventing collision even with an abrupt stop of the preceding train [13].

Unlike other forms of transit, railway operation is characterized by periodic travel of high volumes of passengers, resulting in a higher requirement of safety in their operation [14, 15]. Theoretically, based on train and track specifications, models can be built to precisely model movement of trains from station to station, and calculate braking distances [16, 13]. However, in conventional systems, operators are unable to ascertain the precise location of the train preceding them. In particular, within subways, due to the much lower traction available to trains during braking, lack of visibility of preceding trains, and relatively frequent service, specialized control systems are required to ensure that sufficient spacing is maintained between trains to prevent collision by allowing trains to proceed into the next segment only after authorization has been provided [15, 13].

In traditional signal control systems, such as is currently used on the TTC, train movement authorization is performed normally using a fixed-block signalling system [15, 16]. In this signalling configuration, the track is divided up into blocks, with only one train permitted to occupy a block at a given time. Therefore, before a train can enter the next block, the preceding train must have cleared the block and any overlap. Blocking can be done in either a closed or open format; in the closed format, the block is normally in the blocked state, while the opposite is true in the open format [13]. Normally block lengths are set such that if all pre-warning and stop signals are followed, no collision can occur.

In traditional signal control systems, such as is currently used on the TTC, train movement authorization is performed normally using a fixed-block signalling system [15, 16]. In this signalling configuration, the track is divided up into blocks, with only one train permitted to occupy a block at a given time. Therefore, before a train can enter the next block, the preceding train must have cleared the block and any overlap. Blocking can be done in either a closed or open format; in the closed format, the block is normally in the blocked state, while the opposite is true in the open format [13]. Normally block lengths are set such that if all pre-warning and stop signals are followed, no collision can occur.

Fixed-block signal control systems consist of a few central components. Subway train detection is accomplished automatically with block detection systems (track circuits, axel counters) that are placed directly on the track to detect block occupancy. These sensors relay train position information (whether a train has occupied or left the block) via signals. Signals, which are normally relayed in a simple format (red, yellow, green) can be either wayside (along the track) or within the cab (on the drivers control panel)[14, 15]. Wayside signals were the first to be developed, looking very much like a traffic signal along the track, and for practicality purposes, normally provide information for a few blocks at a time [14], and are coupled with trip stops (devices on the track that raise to trigger emergency braking) for extra safety in case signals are missed. Cab signals, on the other hand, are more modern and display
signal information within the train operators cab, and can provide both signalling and speed information [15, 14]. A red signal directs the train to stop (and will be triggered automatically if ignored by the driver), yellow is caution indicating that the up-front signal is red, while green indicates clear. Certain signals also indicate interlocking status (a series of track locks that form a route) when a normal route is not followed [15, 16].

In operation, when a subway train enters an occupied area, the signal for not only that block is turned to red, but several blocks following as well to ensure collisions cannot occur. For example, in a simplified signal control system, a train only changes the signal for its occupied signal block to red, and the following one to yellow. If an approaching train sees the yellow signal and slows down, it will have enough time to stop before colliding with stopped preceding train. However, as yellow signals are not strongly enforced, if missed, an accident would occur. In the worst case, a train will reach the red signal at full speed with insufficient time to stop. As a result, a technique called overlapping is used, where each signal also controls a set of prior signals to ensure a strong red signal is given in time. In this type of operation, multiple blocks in addition to the occupied block are indicated as red, to allow a train moving at full speed when reaching the first red signal to safely brake avoiding any collision with the rear of the leading train.

Such a scheme of operation results in large safety gaps between trains, increasing the minimum headway for operation. Newer signal control systems, via automated control tackle these deficiencies to expand overall capacity [15, 13].

Microscopic subway line simulators combine simplified models of these real-world train dynamics and signalling systems with posted schedules, each customized to answer specific questions [12]. For the purpose of emergency response or dispatching, synchronous microsimulators are most appropriate, simulating all trains being modelled simultaneously, allowing for interactions and stochastic behaviour [12]. Several commercial simulators are currently available, often geographic specific; SIMON is used in Sweden, VituOS and SABINE in Germany, UXSIMU in Denmark, SIMONE in the Netherlands, RAILSIM in North America, while OpenTrack (developed at ETH Zurich) is used throughout Europe and recently started receiving more worldwide use [17]. OpenTrack, specifically, has the ability to work with the public transport facility pedestrian simulator SimWalk Transport via the RailML XML-based standard to allow for the ability to have more realistic dwell times; the integration, however, is not seamless, only being able to be run sequentially.

2.4 Station and Pedestrian Simulation

As part of the overall examination of crowd and subway network dynamics, the proper modelling of crowd behaviour is a necessity. This is particularly true in the context of their behaviour in stations, given their degree of importance in the transit system as areas of large crowd movements. Ideally, a comprehensive analysis and modelling of pedestrian behaviour in stations would be performed. That, however, was beyond the scope of this thesis, and modelling efforts were instead focussed on key bottlenecks within the station that would have the greatest chance of affecting the network as a whole. As detailed in sections 6.1.6, a simplified model was used for the majority of the stations, with the pedestrian simulator MassMotion integrated for key stations requiring more detailed modelling.

An accurate pedestrian motion model is particularly important for peak hour travel through subway stations, where the high-density movement of people can lead to non-linear behaviour difficult to
understand through closed form equations. Gaining a complete understanding requires empirical data collection, with regression analysis of real-world pedestrian movement, and basic model development to elucidate underlying mechanisms dictating pedestrian movement and interaction.

2.4.1 Statistical Models

Due to the high number of interactions between pedestrians in large crowds, mid-century, field measurements were limited to passenger volume and speed measurements, with an assumption made that flow was deterministically linked to density and corridor width, while the influence of other factors and local interactions were ignored [18]. Fruin performed further research in the 1970s by conducting much more detailed analysis of pedestrian flow, developing fundamental density-flow relationships, thoroughly detailing pedestrian types and dimensions, and defining levels of service in pedestrian planning [19]. His work continues to be used in guidelines for the design of pedestrian facilities even today.

The development of sufficient computing power allowed for the use of detailed simulated models of pedestrian flow. Observation, however, remained necessary for basic statistical analysis of flow, and also as a basis to develop, calibrate and validate models that were becoming microscopic in nature. Earlier studies focused on basic speed distributions and qualitative description of pedestrian tendencies during movement. Pedestrians were found to have a strong preference for the most direct route, even if taking a detour would avoid crowding, and preferred that a single direction be maintained for the majority of travel [20]. Speeds were found to follow a Gaussian distribution (mean of 1.34m/s, standard deviation of 0.26m/s), with an individualized preferred speed [20]. It was also noted that pedestrians tend to maintain a certain distance, varying inversely with density, from obstacles and other pedestrians; those at rest prefer to distribute themselves uniformly unless they are with acquaintances [20]. Individuals who are with companions tend to move together, with the group behaving as if it were a single person. Lastly, people were found to be creatures of habit, usually acting automatically without modifying their behaviour for every variation of situation [20].

More recent research has continued the measurement of modern transit systems, especially in Europe and Asia. Peak hour capacity analysis at Chinese subway stations showed that security checkpoints, stairs and escalators were the critical points with respect to capacity limitations [21]. Further studies looked at comparing the values obtained in Asian stations to models developed in Europe, and found that that existing fundamental models relating speed and density could reproduce observations, but different parameters were required [22]. Specifically, walking speed and pedestrian density up stairs for Asians were both higher than Europeans [22].

Field observations have, additionally, recently served as a basis for empirical models. Transfer times between subway-lines have been investigated, with walking time found to follow a lognormal distribution [23]. Other research has found that aggregate crowd movement is dependent on overall forward drive and the level of crowdedness [24]. The behaviour of pedestrian usage of stairs and escalators has also been examined. Cheung et al developed a logit discrete choice model for the decision of either taking the stairs or escalator. This allowed for a discovery of increased sensitivity to travel time down escalators relative to stairs, while also elucidating a cultural difference in walking speed and density (and therefore capacity) between Asians and Europeans [25].

Laboratory experiments have also been used to understand basic crowd movements under controlled circumstances. These have included studies showing that propagation of lines of people are quicker to start and generally slower uphill [26], explaining discrepancies between studies, and determining whether
new theories of bottleneck capacity were valid [27]. In the latter research, the theory that capacity increases linearly rather than stepwise with bottleneck width was found to be valid [27]. Discrepancies between flow studies were explained through differences in initial conditions, and jams were found earlier in laboratory than expected by existing models [27]. This was attributed to three causes, local density fluctuations, a lack of local organization, and larger than predicted preference of personal space at higher density [27], all of which would require microsimulation models to explain.

For the presented research, these prior observational studies illustrated the need for disaggregate models (local variations in flow) and pinpointed specific issues (e.g. psychological issues between the use of stairs and escalators) that may need to be reflected in the utilized behavioural model for accuracy. In addition, the influences of geography and culture on pedestrian movements reflected in some studies highlight the importance of collecting data specific to Canada or North America for proper calibration. This data is currently severely lacking with research almost completely contained within Europe and Asia.

2.4.2 Mesoscopic Models of Pedestrian Movement

Moving beyond descriptive analysis towards a more fundamental analysis of pedestrian movement required the development of movement models that could replicate prediction of flow under various conditions. Early attempts focused on aggregate movement by treating moving crowds as moving fluids, unconcerned with individual behaviour. This, however, ignored the stochastic nature of pedestrian motion requiring a more disaggregate approach.

Mesoscopic models are in between macroscopic fluid models and microscopic models in terms of both level of detail and run time. These models represent the spatial environment using a network graph. Modelling pedestrian travel has mainly involved the use of queuing models, focussing on key areas that cause delay for pedestrians in a station [28]. In queuing-based models, pedestrian pathways and stops are represented as a queuing network, with pedestrians as a customer of a queue, spending a stochastic amount of time (being served) in each section [29]. In one study, increasing pedestrian congestion resulting in the congestion has been modelled by state-dependent service rates, with decreasing rates for higher densities, but this requires that density be uniformly distributed throughout the system [30]. An alternate time-step method is used in place of queuing in the pedestrian mesoscopic simulator PEDSTREAM; agents advance along links between nodes based on the Fruin speed-density relationships, using the calculated surrounding agent density at each point along their trip [31]. The significantly increased computational speed achieved by such a mesoscopic approach allows for simulation of large scale crowds [32].

2.4.3 Microscopic Models of Pedestrian Movement

Modelling of any local disturbances (i.e. an individual temporarily stopping or falling down) is not feasible in these more aggregate models. Dealing with complex pedestrian movements of varying density and direction towards different exits in an urban space cannot be accomplished, and requires a different approach, instead understanding crowd flow as the patterns emerging from the movements and interactions of individual pedestrians. These newer models are microscopic in nature and have taken several forms, some based on existing traffic theory, while others more sophisticated to handle the peculiarities of pedestrian behaviour.
Force-based Microsimulation Models

Researchers first developed microsimulation models by treating individuals as particles moving through space influenced by various forces. An early version of these force-based models borrowed from magnetic theory [33], but modern force models are primarily based on the social forces concept [34]. Borrowing from traffic theory and sociology, pedestrians are guided by internal forces, which drive movement, and both physical and psychological forces, which prevent collisions with their surroundings and other pedestrians [34]. These forces are placed in a Newtonian-like stochastic differential equation that can be solved to determine the updated velocity and position of each pedestrian at each time step. In addition, with pedestrians being treated as particles, they react automatically to the fields, and have no ability to think; however, they can be given various characteristics for how they react. The model has been shown to be able to reproduce some self-organizing phenomena including side preference and the formation and persistence of lanes [20].

The social-force model is, however, quite computationally intensive. Researchers applying the concept perform some simplifications in order to have reasonable run times. These simplifications have included pre-calculation of pedestrian velocity at each point before simulation based on the force fields (ignoring pedestrian fields) [35], ignoring lateral forces and collapsing the model to one-dimension [36], and eliminating the stochastic term altogether [35, 34, 36]. On the other hand, model extensions have included incorporating contact and dampening forces, and the idea of subgroups (i.e. families), which are of particular relevance in movements through subways stations, through the use of formation attractor points [37]. The model, however, does suffer from some fundamental issues, primarily the inability to deal with any pedestrian-pedestrian contact (avoiding collisions altogether), requiring precise tuning of force parameters for realistic output [38], and an inability for pedestrian learning or complex routing.

Cellular Automata Microsimulation Models

Cellular automata models discretize space into a grid of cells, with each cell only able to hold a single pedestrian at any time step [39]. In general, cellular automata models are developed on the premise that by using minimal local rules, complex emergent behaviour can be produced while maintaining efficient computation. Pedestrians move from their current cell to a neighbouring cell based on a transition probability matrix using behavioural rules that can be influenced by various factors depending on the model (status of neighbouring cells, pedestrian characteristics, etc.) [39]. This structure results in some key built-in assumptions: acceleration and deceleration are instantaneous, direct collisions cannot occur, and long range complex routing is not possible. This concept, after being used extensively in traffic simulations, was first applied to pedestrians by Blue and Adler in 2001 using key behavioural rules for side-stepping (lane changing), moving forward and conflict avoidance via place switching [39]. Simulation results proved consistent with the fundamental flow-density diagrams for unidirectional flow, but produced higher speeds for high-volume interspersed flow, with pedestrians able to negotiate and avoid jams [39].

Subsequent researchers attempted to deal with limitations in their application to pedestrian movements. Cellular automata's inability to allow for long-range interactions or reproduce some key phenomena (jamming and oscillations at bottlenecks, and herding during panics) led to the introduction of floor fields [40]. Floor fields were defined to have two types that could affect pedestrians with varying degree, static fields generated by stationary objects (barriers, exits) and dynamic fields produced by
pedestrians. The static fields could be repulsive or attractive, dependent on if barriers or destinations, respectively, produced them. The dynamic fields, on the other hand, are strictly attractive to simulate the herding and trail making behaviours of pedestrian movement [40]. These changes, along with a friction parameter to model the blocking of movement, allowed for the cellular automata model to reproduce bottleneck oscillation and herding behaviour [40]. More recent modifications have included allowing for non-orthogonal movement through the use of a real coded lattice gas framework [41], and pedestrian perception [42].

While improvements in force-based and cellular automata models have allowed for more sophisticated and realistic modelling, their basic framework presents some key challenges for complex urban movement. Both models oversimplify pedestrian motion, either treating them as particles (force-based) or similar to vehicles (cellular automata). They also both require oversimplified spaces with an inability to perform complex rerouting or have pedestrians visit multiple destinations. Therefore, while they might be appropriate for simple open spaces or passages, they are insufficient for more elaborate areas, as one would find in a subway station.

Agent-Based Microsimulation Models

In this framework, pedestrians are modelled as a set of autonomous agents navigating through space meeting specific goals. This allows for simulated pedestrians to perform more sophisticated path finding, gaining the ability to navigate more complex spaces. In addition, pedestrians also can be programmed to have cognitive capabilities, although models to date have sided towards simple mental processes. As with the previously mentioned models, they are premised on the idea that autonomous agents acting for their self-interest via simple rules can produce complex emergent crowd behaviour. [43]

Several agent-based models have emerged over the last decade. While different from cellular automata, these models have still tended to use a discretized grid for agent movement. One of the first pedestrian movement models to employ agents was the STREETS model near the turn of the century, simulating pedestrian movement through an urban sub-region constructed from detailed GIS data after entering at specific gateway points [44]. While final destinations and overall routes are predetermined, the model incorporates pedestrian vision to allow for attractive locations to provide temporary distraction [44]. Legion (later developed into Myriad II) simulated the process of mobile agents finding their way towards their goals using a technique of simulated annealing via a path of least effort [8]. With a focus on safety, Still provided his agents with very limited intelligence in how they moved and reacted to the environment to find areas that had potential safety issues [8]. The model was able to reproduce some observed pedestrian phenomena such as edge effects, crowd compression around corners and fingering of opposite flows [8]. PEDFLOW focussed on movement along sidewalks and shop-fronts, with pedestrians having both limited perception and incomplete information while attempting to reach their goals [45]. Acknowledging that pedestrian behaviour was context-specific, this model incorporated a rules-based approach that all pedestrians followed dependent on their immediate circumstances [46]. Rules were determined by both watching videotapes of pedestrian movement, but also via interviews for subjective behaviour (attitude, perceptions, etc.) that could not be directly observed [47].

More recent models (NOMAD, SCA, MAGE) have incorporated more sophisticated mental and movement processes than their predecessors. The NOMAD simulation software used a discrete-choice activity-based approach, allowing for a pedestrian to visit multiple locations in a chain, with intermediate movements free and continuous similar to the particle-based social forces concept [48]. The incorporation
of time-sensitive activities allows NOMAD to be used in more complex urban settings. In situated cellular agent (SCA) models, pedestrians are treated as autonomous agents moving through a discretized space formulated as a graph. They are exposed to fields generated from their surroundings, responding differently based on their internal state [49]. While similar to fields in the social-forces model, the fields in the SCA model are of greater variety, transmitting any type of information pertinent to pedestrians [49]. MAGE takes a somewhat different approach with a focus on subway stations. Each agent moving through the station has a vision field to be aware of other entities, while its general path is formulated as a directed graph to navigate the closed space [50]. Along the way, the agent will avoid other pedestrians and obstacles within its field of view, while having a possibility of also being attracted off-route by entities like ticket booths and stores. At higher densities the system assumes that behaviour changes, and people organize into a crowd traveling with a group velocity [50]. This method allows for an examination of the effect of collision avoidance and group behaviour on flow rates in confined spaces, but ignores more complex behaviour at high density by assuming that a distinct aggregate crowd personality emerges.

Validation of Pedestrian Models

While microsimulation systems allow for a more fundamental investigation of pedestrian movement, their intensive data requirements make validation difficult. In addition to more aggregate density-flow data, data specific to individual pedestrians and the flow patterns of groups of pedestrians are required. To date, this information has been collected via video camera footage, with extremely time-consuming manual tracking of individual pedestrian trails. As a result, researchers developing models often limit their validation to either aggregate fundamental density-flow diagrams and/or qualitative validation of reasonable behaviour for the simulated pedestrians [8].

This is a key challenge as no consistent framework is currently available to properly judge the quality of a pedestrian modelling technique or system. Ad hoc validation of models was shown to be problematic in recent work that examined whether the self-organizing phenomena seen in pedestrian movements could be replicated by the NOMAD simulation system [51]. While the model did produce results that somewhat agreed with aggregate speed-density relationships, an excessive rear-force term was required and head-on collisions were a problem at the agent level [51]. In addition, lane formation was underestimated, and parameters could not be found to ratify this error [51]. This study illustrated the need for detailed pedestrian-level calibration data for proper validation of microscopic models.

Researchers have begun to attempt to tackle this key challenge through improved collection techniques. Of particular interest is a formal stochastic behavioural model that provides assistance to automatic video-based pedestrian tracking systems. With a different focus from other pedestrian models, a more precise modelling system of short-length movement was required to allow for predictive capabilities at a microscopic level [52]. A nested logit model was developed for pedestrian decisions of speed and direction at each time step through a discretized space, while taking into account the locations and velocity of other pedestrians [52]. This concept has been implemented within a simulation framework with qualitatively appropriate results, but requires field-testing to verify quantitative accuracy [53].

MassMotion Pedestrian Simulator

MassMotion, a commercial pedestrian simulator, was chosen to act as the station simulator for stations chosen to be simulated in detail. This choice was driven both by the unique capabilities of the software (described later in this section), and because of a partnership formed with the Toronto office of Arup,
the developers of the software, allowing for programming access to enable tighter integration with the developed system.

MassMotion is an agent-based three-dimensional simulator. The environment, as such, is built using 3D architecture, with separate floors connected with level-change elements like stairs, escalators, ramps and elevators. Agents enter and exit the environment via portals, and are able to be given a series of tasks to do in the meantime, including visiting specific areas and being involved in queueing processes. As with most agent-based pedestrian models, movement in MassMotion is governed by two layers of guidance. The first, a short-range ‘reflexive’ model, adapts the Social Forces concept, with surrounding agents and obstacles producing forces on individual agents that guide their decision of velocity at each time step [54]. This behaviour has been calibrated against Fruin’s Level of Service (LOS) standard to ensure that speed-density profiles are consistent with field values [54]. The second part is a longer range pathfinding model, based on an application of a modified Djisktra’s algorithm on a network representation of the entire space, with a cost function that takes into account a variety of factors including level changes [54]. This navigation model also considers agent congestion along the route, making MassMotion one of the few pedestrian simulators with responsive agents. In addition to basic way finding, MassMotion contains elements that allow for more complex behaviour. These include process chains, to simulate queuing processes as would be present at ticket counters, and gates, allowing for controlled flow and simulation of doorways.

These tools within MassMotion allow for it to be used to model passenger movement through stations, with some limitations. The combination of barriers, floor, link (to connect floors), stairs and escalator objects allow for the construction of the general architecture of the station. Portals provide entrance and exit locations at doorways and train cars, while links with timed gates permit the simulation of opening and closing of train doors. By default, however, these train door gate timings have to be set as an input parameter; the same holds true for introducing agents into the simulation. For proper integration of the software as part of a network simulator, a method was, therefore, required to allow for train door timings and agent entry to be set at runtime; the software was modified to permit such ability (see Chapter 7).

**Boarding and Alighting Models**

Boarding and alighting models are focused on the dwell processes that occur when subway trains (or any transit vehicle) make a stop at a station. The pedestrian movement models described in the prior section are key in understanding these processes [55], but the situation is unique, requiring specific consideration. Because of pedestrian interaction, dwell times are not constant, and vary according to the number of people alighting and departing, as well as train configuration and the level of crowding [56]. The most commonly used gauge of dwell time is Westons formula (Equation 2.1), which takes into account the number of alighting \((A)\) and boarding \((B)\) passengers, the number of doors \((D)\), the peak door/average door factor \((F)\), the number of seats \((S)\) and the number of through passengers \((T)\), while assuming a standard time of 15s for doors to open and close.

\[
SS = 15 + \left[ 1.4 \left( 1 + \frac{F}{35} \frac{T-S}{D} \right) \right] \\
* \left[ \left( \frac{F * B}{D} \right)^{0.7} + \left( \frac{F * A}{D} \right)^{0.7} + \left( 0.027 * \left( \frac{F * B}{D} \right)^{0.7} \left( \frac{F * A}{D} \right)^{0.7} \right) \right] 
\]  

(2.1)
The formula has been shown to provide reasonable values for systems around the world, although parameters do require tweaking [57]. However, it assumes a deterministic dwell time, and the stochastic nature of driver and passenger behaviour that accounts for a significant portion of variability is not considered [56]. Nevertheless, regression models of field data have been the most popular method of estimating dwell time for a situation [58, 59, 60]. Laboratory experiments with video tracking of pedestrians have also been performed [61].

More recently, moves have been made towards simulation models to capture dwell times in more unpredictable situations, and incorporate behavioural tendencies of pedestrians [62]. Generally, however, the stochastic nature of delays has not been given much attention. This is problematic due to the strong relationship between dwell time variation, the level of propagation and the impact on overall system performance [63].

The main limitation in the use of these models is the peak door to average door factor, which is often set exogenously. Understanding how passengers spread themselves across train cars and train doors is an area of research onto its own. The pertinent literature is reviewed within the platform modelling chapter of this dissertation (Chapter 5).

### 2.5 Existing Methods of Transit Assignment

Transit assignment is the process of loading passengers onto the transit routes of the network to match real world flows. Generally, the transit assignment process will involve both determining transit paths for individuals, and some model of the transit service; these interact to result in route loads and level of service measurements [64]. While the development of a transit assignment method was not a goal of this dissertation, it is a necessary component of a proper transit network simulator. In particular, Nexus had a goal of having multiple use cases. As such, it was built with the assumption that the degree of sophistication of transit assignment might vary.

Transit assignment types can first be classified based on whether they assign agents in the aggregate or treat them individually. For aggregate models, two general types have been developed and researched over the years. The first, strategy-based assignment, works under the assumption of fixed demand, and frequency-based transit service. Each individual is assumed to have a strategy of attractive transit routes they would consider as they make their way from origin to destination. This method is not responsive to time-variant transit service, congestion along routes and sudden service changes. The second method, schedule-based assignment, attempted to incorporate the variability of transit service, by moving from a headway-based representation to the actual schedule of vehicle trips. Additions have also been made to this method to allow for capacity of vehicles to be introduced, allowing for the effect of congestion on choice to be incorporated. Nevertheless, both of these methods fail to allow for proper modelling of service dynamics or more complex behavioural responses of passengers, particularly to new on-route information received about the service [64]. This is in part due to the aggregate nature in which they operate, with overall flows between origin and destination assigned, rather than treating transit users individually. In addition, as both have simplified network graphs to represent transit service intrinsic to their method of solution, neither lend well to the simulation-based transit service model used within Nexus.

Instead, Nexus focuses on disaggregate models. In this type of assignment, transit users are treated as autonomous agents moving through the transit service; the service itself can be modelled in aggre-
gate or using detailed simulation. These models can be further split into static and dynamic models. The former disaggregate origin-destination transit user data into individual trips, through the use of a generalized cost function, normally choosing the lowest-cost route. Such a type of static disaggregate assignment is used by MADITUC, a commercial transit assignment software developed by Ecole Polytechnique in Montreal, and in use by several transit agencies in Canada [65]. Newer methods have emphasized improving the dynamic nature of the models, both from a service and user standpoint. Their focus has been on enabling incorporation of vehicle characteristics (both physical and operational), the effects on service due to agent choices, and behavioural models for individuals to respond to varying transit network conditions and information provision to simulate transit ITS deployments [64].

In the final implementation of Nexus, it is expected that one such learning-based approach, MILATRAS (Microsimulation Learning-based Approach to Transit Assignment), will be used as the transit assignment engine. MILATRAS is a novel transit assignment method developed at the University of Toronto which uses a population of agents that learn to choose their path through the network and appropriate departure time using a learning-based procedure, converging over many iterations [64]. It has been shown to perform comparatively well against traditional equilibrium-based transit assignment methods such as EMME, while allowing for more policy-sensitive analysis [65]. Incorporation of MILATRAS into Nexus will allow for more accurate performance evaluation by incorporating crowd flow for use within its decision-making processes, while harnessing its ability to dynamically route based on information provision.

2.6 Disruption Management of Transit Networks

Disruption management is the process of detecting and responding to unforeseen situations that disrupt operations, such as train breakdowns, non-panic emergency situations (person on track) or power outages. The duration and severity of disruptions are usually not known with clarity, requiring continuous adjustment, making the process one that has to be handled in real-time [66]. Current methods to manage disruptions in public transit systems normally involve manual intervention by transit operators on an ad hoc basis [2]. The aim is to return the distribution of transit vehicles and crews to normal and back on schedule as quickly as possible through timetable, rolling stock and crew rescheduling [2]. While operations research techniques are used by agencies in modern transit planning, these techniques have not made significant inroads in performing operational tasks [66]. Instead, the rescheduling and repositioning tasks are often carried out sequentially in order of greatest urgency, under tight time constraints, resulting in suboptimal solutions where interactions between tasks are not taken into account, with decisions made that often act at cross-purpose [2].

Various strategies are used to return service to normal, involving rescheduling of timetables, rolling stock and crews. Timetable adjustments involve rescheduling of trains surrounding or affected by the disruption. Buffer times usually inserted in schedules to allow for recovery from small disruptions, while somewhat more serious disruptions are dealt with using dispatching rules, either repositioning transit units or changing driving and stopping patterns [2]. For larger disruptions, to prevent significant build-up, pre-planned emergency scenarios are implemented. These usually involve cancellation of service surrounding the disruption, premature turnarounds before the incident, rerouting or even full line cancellation [2]. While performing these timetable adjustments, operators also are required to ensure that rolling stock are rescheduled and repositioned based on the changes, and that all adjustments make
sense with crew rosters [66]. In subway systems, the ability to make changes is normally more restrictive than surface rail resulting in optimal rescheduling being more difficult [67].

The various types of delays and the methods of response across the various orders of public rail transit were catalogued recently in a wide ranging survey of international transit agencies [68]. Interviews were conducted with representatives from 48 passenger rail transit agencies, ranging from operators of light-rail transit to commuter inter-city rail. Agency representatives were queried on a range of issues, generally surrounding their agency-specific causes of unplanned disruptions and their approach in and challenges of managing these disruptions. Disruptions were found to be caused by five categories: medical emergencies, weather/natural disasters, track and signalling problems, mechanical problems with rolling stock, and conflicts with vehicles from other rail services. Problems with track or train equipment was most likely to result in service delays, while medical emergencies, particularly suicides, could cause partial or complete line closures. Weather or natural-disaster induced disruptions were also long-term and serious, but due to this long-term nature allowed for service planning after the initial event.

While the causes were defined to be many, survey respondents indicated that the approach to respond collapsed these varied causes into two main scenarios [68]. The first is whether the disruption event resulted in a disabled train. In these situations, the most common response was to transfer passengers to a subsequent train, often also putting a spare train into service if available, followed by encouraging the use of alternative transport [68]. Altering the schedules of other trains to compensate was also performed by a small number of respondents (10%). The second type was whether the event caused a complete or partial blockage in the rail lines. Bus bridging was by far the most popular approach (86%), followed by bypassing the disrupted segment through the use of crossovers where available, and diverting commuters to alternative transport networks or other lines [68]. While, bus bridging was popular, it was not the first option by many agencies due to the inability to acquire sufficient buses for use in a reasonable timeframe [68]. It should be noted that 10% of respondents provided no alternatives to disrupted commuters.

While automation methods have yet to reach widespread use in the transit industry, the last decade has seen an emergence of researchers applying operations research techniques to the rescheduling problem [66]. Techniques such as mixed-integer and constrained programming have been applied to rescheduling following a series of compounding minor delays [69], and for simultaneous timetable-rolling stock [67] or timetable-crew roster [70] optimization following larger disruptions.

Such optimization methods are an important part of decision support systems, computer systems designed to estimate the current state of the network and provide assistance in determining the best course of action in the event of disruption. Researchers have designed several systems using a range of approaches. These have included using the previously mentioned OR techniques [69, 70, 67] for rescheduling without rerouting, using agent-based dispatching to minimize passenger delay during connecting transfers after minor delays [71], and assistive systems for delay-induced route conflicts through rerouting, reordering and rescheduling [72]. Generally, these alternative networks were more prevalent in urban areas, mainly consisting of surface routes, but sufficient capacity during peak periods has been a problem.

Current research and developed systems suffer from some problems when it comes to application to subway networks. Research has heavily focused on rail networks, rather than urban subways, which have very different physical constraints and service frequencies. Most importantly, the influence of pedestrian dynamics is often ignored, and the need to account for and clear pedestrian build-up is often
neglected [73]. As a result, solutions have focused on rescheduling service as close as possible to normal operation, without considering the effect on passengers and level of service. Lastly, a focus on surface rail has allowed researchers to ignore the other components of the transit network (i.e. buses) and build singularly focused support systems. Overall, a complete system that inputs live data, estimates network performance and allows for testing of a range of strategies, as one sees emerging in evacuation support systems for traffic management [74], are not currently formulated for subway networks or transit in general.

2.7 Moving Forward

As was detailed in this chapter, there has been significant research and development efforts in recent years pertaining to all facets of the transit network. In particular, pedestrian modelling and disruption management response are both relatively new fields, with many gaps, as detailed in this chapter, remaining individually. One such gap in knowledge in pedestrian modelling, understanding how transit users make their way between levels in a station, is tackled in Chapter 4. However, the major gap that is addressed in this dissertation is the tendency, because of their individual complexity, to treat these areas in isolation. While some progress has been made in transportation network simulators to incorporate transit and pedestrians, their auto-focussed origins have limited their ability to do this properly. The Nexus frameworks and implementation (Chapters 3, 6 and 7), and the modelling of passenger behaviour on platforms (Chapter 5) specifically target this deficit.
Chapter 3

System Frameworks

In order to conduct the analysis for this thesis, a tool was required that could model the individual components of the network, properly handle interfacing, and allow for a high level of control to test scenarios, direct actors, and produce the necessary feedback data. While significant work has been done in each of these individual areas, little attempt has been made towards integration, relying instead on more abstract methodologies to understand network flow. As detailed in the prior chapter, current commercial traffic simulation software have begun to move towards incorporating pedestrian and transit vehicle movements; however, they are at an early stage of development, concentrate on safety and interactions with autos, focus on localized pedestrian flow, and have limited programming interfaces to allow for modification of pedestrian behaviour for research purposes. On the other side, pedestrian simulators are able to perform well when examining detailed movement through specific spaces, but have very limited connectivity with the transit network and do not scale with high computational requirements. In all cases, current systems are unable to take into account pedestrian decision-making and route choice at a network level, or allow for activity, behaviour or learning-based models to be incorporated, necessary to properly understand pedestrian/passenger response to sudden network changes. Therefore, before being able to analyze disruptions and the delays they cause, a set of frameworks with a focus on network-level movements and behavioural responses must first be developed.

This chapter outlines the frameworks that would enable efficient, scalable and flexible agent-based transit network simulation, and informed the construction of Nexus. They are presented in their idealized forms, specifying end-points goals, abilities and characteristics of the framework and its components. As this is an ambitious effort that will require several rounds of research to approach completion, the aspects that have been dealt with in this thesis will be identified. Any simplifications or modifications that were made upon implementation of the prototype are detailed where the implementation is described (Chapters 6 and 7).

3.1 Research Framework

As mentioned in the introduction of this thesis, one of the key drivers for Nexus is as a research platform for general transit modelling research. As a result, while the main goal was a tool to allow for transit disruption analysis and management support, a broader consideration was needed on longer-term needs that would shape its required capabilities. This took the form of a research framework that could act as
a roadmap for both the thesis presented here and possible future directions, but also define how major transit network modelling components would be arranged and communicate at a high level. It, however, was meant to act as a guide during development and did not attempt to be comprehensive as to its potential research applications. As shown in the diagram, the framework’s focus is on research into both modelling the effects of and responses to disruptions; these include operational strategies and better understanding of passenger behaviour. In addition, in order to properly predict network-level impact, also of interest are key areas both within individual transit system components and at their interfaces. Lastly, the framework places research specific to network-level simulation in context with other models that would be required that provide input (often with feedback) and interact with other transportation modes and land use. The portions that are dealt with in this thesis are highlighted, with a focus on simulator architecture and improved crowd simulation, including the development of some key models. The remainder of this chapter details the frameworks surrounding the simulator architecture, while the implementation of Nexus and component behavioural models are dealt with in subsequent chapters.

![Figure 3.1: Research framework](image)

### 3.2 Designing a Crowd & Transit Network Simulation Framework

An appropriate simulation framework requires some key characteristics that are important for performing network level analysis of transit networks in an efficient manner. With respect to model design, appropriate and accurate models must be incorporated for each of the individual components and actors involved in transit flow (surface transit vehicles, pedestrians, line operation and communication services). The system should also be able to have some provision to incorporate the effect of other vehicles on transit, whether through speed distributions which account for their impact or by direct modelling of these vehicles. Individual model components should ideally be designed to allow for scalability, dynamically and intelligently avoiding highly detailed modelling where unnecessary to improve computational efficiency, if accuracy can be maintained. This is an issue with current simulators, which
maintain a single detailed microsimulation model regardless of density or circumstance. As simulators with such an ability do not currently exist, the framework should at minimum have the capability to seamlessly coordinate event-based and time-step simulation types allowing the most appropriate model type to be selected by the modeller. This coordination should be possible across multiple computers to bring sufficient computing power to bear for simulation of large scale networks.

Of particular importance are requirements necessary for systems-level analysis of pedestrian behaviour and its influence on the transportation network, and specifically on public transit with its higher pedestrian volumes. Current simulation software do a poor job of properly handling network pedestrian flow; traffic simulators which have begun incorporating pedestrian movements essentially have assimilated existing pedestrian simulators, focussing on examining pedestrian-vehicle or pedestrian-space interactions with no coherent link to network routing. Proper understanding would require an agent-based approach with the ability to re-route in response to network changes. In addition, to account for the effect of pedestrian dynamics and level of service (LOS) on transit LOS, particular attention must be paid to points of interaction of large volumes of passengers and transit vehicles (e.g. dwell processes in metro stations). This would be in contrast to current methods in network microsimulators, where boarding and alighting rates are normally assumed constant, independent of crowding. Lastly, the simulation of pedestrian movement should follow the previously stated principle of avoiding unnecessarily detailed modelling in individual components where possible.

With respect to overall design from a research perspective, it is important that the framework be designed with a high level of modularity to provide sufficient flexibility to allow for modification and expansion without affecting other components, and allow for independent operation of modules. This is particularly key for subway networks, where interaction between components occurs periodically, allowing for parallel computation to be harnessed. In addition, a common communications and data framework is required to allow for proper flow of data and extraction of results. Critical for disruption response testing, this communications framework should allow for on-the-fly simulation of operational response scenarios. To increase the chance of adoption, existing commercial software should be leveraged wherever possible for system components. Lastly, the ability to run in parallel should be inherent for simultaneous evaluation of various scenarios or for statistical averaging.

### 3.3 Simulator Requirements

Based on these considerations, the following list summarizes the requirements considered.

- Allow for custom or existing commercial simulators of each major transit system component (surface, station, line) to communicate through a common interface with standardized data structures
- Ability to handle simulators using a variety of resolutions (discrete-event to detailed time-step)
- Contain provisions to handle other forms of transportation that might influence transit service
- Use a high degree of modularity to allow for piecemeal modifications and additions, parallelization of individual simulation runs
- The ability to run multiple simulations in parallel
- A communication framework to enable relaying of pertinent results from all components, and on-the-fly control of components
• An agent-based approach in modelling transit users to allow for intelligence in routing, including dynamically adapting to network changes

• Mechanism to maintain agent awareness of its own network path while navigating through all components

3.4 Simulation Framework

To address the needs of modularity and scalability, the proposed computing framework is designed using a service-oriented approach to build a network simulator by connecting individual simulators. A service-oriented approach is an application architecture where the overall solution is composed of separate applications each providing specific functionality as "services" to other applications over a connection interface. In accordance with this design pattern, the transit network is divided up into its logical components (stations, subway network, surface network), with the possibility of separating out components to run as external applications where necessary. As shown in Figure 3.2, the underlying simulators are interfaced with the main Nexus system by means of an intermediary communication layer; this layer handles preparing data in the proper format for transfer and storage of any relevant data to be maintained outside of the underlying simulation software. This permits two key abilities; the first is to allow external software to handle simulation for specific components; the second is to allow for simulation to occur on a single computer, multiple computers over a local network or on a cloud-computing platform. The service-oriented approach enables this ability; in this methodology, the individual programs that make up the combined package expose external connection points, allowing communication with other applications on the same or networked computer. This permits passage of information between various components, including data specific to the network structure, agents and vehicles, communication packets to control the simulation, and feedback or results.

Pedestrians, as agents, move between simulation services (station, train, surface vehicle) and have their routes determined through processing by decision-making modules. Also, in order to be able to analyze movement at the network level, the framework design prioritizes scalability, incorporating the
ability to interface with models of different abstraction levels simultaneously as required. A highly mod-
ular design combined with a specified interface allows for the ability for components to run standalone,
and the ability to modify or add framework components without affecting the rest of the system. It
also permits components to be written in languages most appropriate to them, and hardware to be
customized to each task (for example parallel graphics card processing for pedestrian simulation). For
this flexibility and ability to run in a distributed fashion, there is a drawback of communication latency
when data has to travel between applications and an expanded amount of memory. As a result, the
framework calls for network components to be contained within the central application where possible,
separated only when simulation on a separate computer will improve performance. Ideally, a preliminary
automatic evaluation should occur to determine the best configuration.

The component simulation software is designed to connect as needed to the main server for design,
simulation operation and/or analysis. In order to allow for efficient computing in a distributed envi-
ronment, they run in an asynchronous fashion, with all heavy computation kept within the boundaries
of each software to avoid latency issues in transferring large amounts of data. To facilitate this, at
the beginning of the simulation run, complete data describing the network and individual components,
and pedestrian info specific to each component is pushed from the data server. During simulation, all
processing of detailed information is stored locally and at the end of simulation, all data is sent for stor-
age and analysis. Data flow is, therefore, designed to minimize interaction between system components
during simulation runs, with exchanges occurring only during agent movements between components,
for clock synchronization or response testing. Such a strategy is common to distributed computing
traffic microsimulations [10]. The system can then be implemented on a network of computers or in
a cloud-computing environment without requiring the purchase of expensive high powered computing
infrastructure. This approach also enables rapid multi-run simulation by providing the ability to run
multiple instances of the combined system simultaneously over a network of computers, either identi-
cal copies in the case of stochastic simulation or for testing the results of different network changes or
response strategies.

The simulation process itself is highly dependent on the specific implementation of the framework. As
simulator components are running in parallel, with vehicle transfers occurring between synchroniza-
tion points, individual components must be aware of any expected signals or information that connecting
components expect to receive at each synchronization step. Outside of the simulation run, however, a
generalized process model can be stated, specifying expected overall inputs, and the major modelling
steps (Figure 3.3).

### 3.4.1 Components

The various components of the framework (Figure 3.4) are defined by specific and unique tasks. The
network designer is used to specify how the individual simulation components are to link together, and
access APIs provided by each simulation component on the data server for individual component design.
These linked simulation components (coordinator, station/line/surface/pedestrian simulators, transit
assignment) can be thought of as one service that can be accessed to perform a simulation given a set
of operational data input, producing detailed results as output to be analyzed, describing pedestrian
and transit system level of service. The remaining components are specified to either access the engine
directly via the coordinator or process simulation data on the data server to perform specific tasks such
as driving the simulation, or analyzing and formatting output.
Figure 3.3: Example process model for running a simulation using the framework

Coordination Server

The coordination server acts as the synchronization engine within the framework, and has several key responsibilities:

- Load network and agent data
- Transfer all appropriate initialization data to component simulators
- Synchronize component simulators
- Handle data exchange between component simulators
- Log results, include those of transit vehicles

The Coordination Server will need to handle simulation clocks running time-step and discrete-event simultaneously. Such a situation might arise where a discrete-event surface traffic/bus or subway line simulator can be implemented to greatly reduce processing time, but where the pedestrian-station simulator still requires time-step simulation. To maintain synchronicity, the controller will pulse all simulators...
at regular intervals larger than their internal clocks, so they move forward relatively together. In addition, discrete-event components will also halt if one of their agents arrives at an interaction point with another component (i.e. train or bus arriving at a station), until told to proceed once the other components have caught up in simulation. This is premised on these periodic interactions being of importance for synchronization and pedestrian interaction not of importance in discrete-event modules, allowing entering pedestrians with earlier timestamps to be moved into place without jeopardizing the integrity of the simulation.
In order to work seamlessly, the coordination server handles all of the complexity of the distributed nature of the system. This involves keeping track of all components, including those which are hosted externally, and being able to correctly pass data (normally vehicles and agents) between components without individual components having to have knowledge of the locations of their neighbours. This also involves relaying any software-specific data (e.g. model files), after retrieval from the database, to appropriate components before simulation start. In addition, it acts as the connection point for the entire simulator service, allowing for external software to specify network and agent data, scenarios and receive feedback. All of this is handled through a set of exposed service points (Figure 3.5).

**Transit Assignment**

The transit assignment component acts to assign paths along transit routes to each agent. While this component typically encompasses representing service characteristics, for the purpose of the framework, it is defined as being limited purely to the decision making model to set the path choice for agents, with service modelling left to the surface and line simulators. As detailed in Section 2.5, several methods exist for this process. To maintain the highest level of flexibility, the transit assignment model is specified in the framework to be as flexible as possible in accommodating these various methods. In general, with respect to simulation-based methods, two possibilities exist, static (fixed path or strategy for each passenger) and dynamic (passengers adapt to information provision with possible modifications to their route during travel). The interface for the module, listed below, was specified to accommodate both of these methods by allowing for an optional function that would re-evaluate agent paths if requested.

- Load New/Existing Population
- Generate Possible Paths for Set of Agents
- Retrieve Next Set of Agents Departing in Period
- Advance Agents to Next Step in Path
- Update Agent Path Choice
- Report Actual Vehicle Arrival and Departure Times

As indicated by the interface, the transit assignment takes in an initial population (with origin, destination and either departure or intended arrival times specified) or loads an existing population
from the database who have already had routes determined. Next, a function is specified to conduct path finding through the transit network for all agents; the result of this effort should be either a set of path or a path graph for each agent to guide them through the network, with a departure time associated with each path. Agents are loaded with, at minimum, information about their current path segment. This information would normally be their boarding stop, route and alighting stop if at a travel segment, or their alighting stop and next transfer stop if making a transfer. As each segment is completed, trip details are assumed to be relayed back to the assignment module, which provides the next trip segment for the agent. Static versus dynamic transit assignment is dependent on the behaviour of this function; in static assignment, no update to path will occur, while dynamic assignment allows for re-evaluation after each segment of the trip. An additional interface is also provided in situations where the transit assignment method calls for agents to re-evaluate their paths even if they are not at an original transfer point. For dynamic assignment methods, information on service performance is required; some of this can be gleaned from agent travel times as they are reported, but an additional interface is specified to allow for vehicle arrival and departure times to be reported.

While a generic implementation of transit assignment was assumed in the overall framework, as shown in the component diagram, MILATRAS, was the main consideration. It was previously developed at the University of Toronto to provide more realistic transit assignment in changing networks through a learning-based agent approach [64]. It is, however, expected that a separate assignment method will be necessary to handle agent behaviour during disruption situations; little to no data is available of passenger behaviour in these circumstances.

In this initial version of the framework, the transit assignment component, while a separate module, was integrated within the coordination server, to permit quick access to network and agent data and permit memory sharing. While this sufficed for the offline implementation of the framework with static transit assignment, moving forward, it is expected that modifications will need to be made. This could involve permitting the module to be run in a distributed fashion to allow for rapid dynamic rerouting of agents.

Component Simulators

The three component simulators (surface, line and station) perform the actual simulations for their respective areas. These are detailed in the section below. In order to work with the framework, component simulators can take two forms, either being complete simulators by themselves, or by acting as wrappers and mediators for commercially available software that have application programming interfaces (API) to enable external manipulation. All component simulators are specified to have a few common interfaces (listed below) that allow for component-specific data to be loaded, which could consist of entire project files in the case of external simulators, initialize the component with the appropriate settings and passing the network structure, and one to perform any final steps (including closing the component files and retrieving component-specific results) at the end of simulation. Simulation, itself, is driven by a specific interface that indicates the duration of simulation period to advance, and returns any feedback produced in that time period. Inputs as part of the interface are listed in parentheses.

- Load Data (Project files)
- Initialize (Component settings)
- Advance Simulator (Duration, Agents transferring to/from street if applicable, Control signals)
Finalize

Surface Simulator

The surface simulator is defined to handle simulation of agents on streets and surface vehicles; it also is the origin and final destination location for all agents. While this does technically allow for simulation of all modes, care should be taken in simulating modes beyond transit (including any segments of other modes used for access and egress). As the focus of the framework is in incorporating the effects of crowd movements on transit service, the main benefits in its use is for handling higher order transit and stations. As a result, in general, the average speed or average speed distribution of surface transit vehicles for each route segment should be used in place for full simulation of all vehicles and traffic lights; this method was chosen for the initial prototype implementation (Chapter 6). If undertaking full multi-modal simulation is required, ideally a simulator that uses a mesoscopic approach rather than microsimulating all parts of vehicle movements should be used to speed up computation. The main concern in all cases is to avoid a situation where the surface simulator requires more significantly more computation time than any other individual component, so as to not be the limiting factor during simulation. In this initial version of the framework, provision has not been made to allow for routes to be input for non-transit agents (those purely driving, walking or bicycling); this would be required for the framework to evolve to a complete multi-modal system. Lastly, unlike with the other components, the framework assumes only a single surface simulator. In addition to the common component interface list, the following connection points are also defined to handle vehicle transfers in between synchronization periods:

- Indicate Vehicle Departure Time From Station
- Transfer Vehicle From Station (Along with On-Board Agents)

Line Simulators

The line simulators handle movement of all grade separated rail vehicles. At-grade rail systems (LRT and streetcar) that have frequent interactions with surface vehicles are assumed to be handled by the surface simulator. At each station, vehicle data is transferred via the coordination server to the appropriate platform and station where boarding and alighting is performed. As mentioned earlier, the component will generally act as a wrapper application for an external dedicated line simulation software, controlling the software via its programming interface and handling all communication with the Nexus system. As these programs do not generally deal with passengers, passenger data is contained only within the wrapper. The interface for the line simulator components is identical to that of the street component:

- Indicate Vehicle Departure Time From Station
- Transfer Vehicle From Station (Along with Agents)

Station Simulators

The station simulators handle movement of passengers as they transfer between transit lines that pass through the station or between the street and transit lines. The framework assumes that each station is handled by a separate simulator with individual connection points. The station simulators are responsible for all passenger movement, both inside and outside train cars and buses when inside the station. As
a result, they also handle simulation of train dwell operations, including timed opening and closing of doors, and boarding/alighting movements of agents; as such the station normally sets the dwell period of vehicles. Entrance into and out of the station, outside of vehicles, is assumed to occur at doorways; as this transfer occurs only at synchronization points, ideally a mechanism should be in place within the simulator to advance agents an appropriate distance to account for the delay. For open-access stations that might occur in proof-of-payment transit systems, specific access points need to be defined to act as pseudo-doorways. In order to allow for local path finding to depend on network routing, all stations should be provided with platform layouts of all possible alighting platforms at accessible stations. The station interface contains functions to handle vehicle arrivals when trains or buses enter the station; it also, however, has a separate function to set the departure time of vehicles where the line simulator has requested a hold of the vehicle beyond normal dwell operations. The interface is summarized below.

- Indicate Vehicle Arrival Time To Station
- Transfer Vehicle To Station (Along with On-Board Agents)
- Indicate Overriding Departure Time of Vehicle

Network Designer

The network designer is the network creation software for the framework. Its main task is to populate the network structure and agent databases sourced during simulation runs; the exact data required is provided in a subsequent section. In order to easily incorporate network data from external agencies, a key ability of the designer is to be able to read Google Transit Feed Specification (GTFS) transit network and service files from a variety of agencies in order to extract the necessary information. The interface should also allow for the saving of project files of external software if used within components, with this data also being stored within the database. Unlike other components, it connects with the database directly, so has no specific interface.

Network Analyzer

The Network Analyzer is the main controller of the framework and has several functions. First, it is tasked with specifying the network to be loaded, scenario creation and loading. Scenario creation includes defining the agent populations (origin, destination and departure time), simulation period, and a set of control signals to mark disruption events and responses. Second, it acts as the main simulation controller by specifying the length of time to run the simulation; for the most part, this will constitute the entire period, unless more control is needed for debugging purposes. Third, it provides visualization of key pieces of data during simulation based on the feedback settings level; this could include info on platform, entrance and exit volumes at stations, bus loads and vehicle locations. Alternatively, such data can be viewed after simulation is complete by loading the results from the database. Lastly, the analyzer, as indicated by its name, is defined as handling all analysis of raw results produced by the simulation, outputting any required graphs, maps and summary data files.

Data Handling

In order to allow for easily searchable results and quick writing, the framework assumes the use of databases for all data handling, both input and output. In the case of large networks and storage of
the results of many runs, the use of a database server is preferable, with the ability to span multiple volumes and allow for delayed insertion of simulation output. Otherwise, a file-format database, like Sqlite can be used. For security reasons, connection points to the database server or file should be limited to the coordination server, and components used to create network structure files or analyze simulation output. The quantity of tables and the fields contained in each is dependent on the implementation of the framework, with the main requirement that all required data are contained or can be calculated from the data structures as defined in the following section.

### 3.4.2 Data Structures

Allowing for communication between components necessitates a common data interface. While the exact format of the data will vary depending on framework implementation, this section deals with the basic data required for passing around information related to network structure.

#### Component Network

The network has two key types of divisions within this framework. The first applies to the connected platform structure, dividing the network by its physical/logical components (surface, grade-separated rail, stations). Definition of this structure allows for the coordination server to properly transition agents and vehicles between the various components of the system. It is assumed to contain the following information:

- Name
- Unique ID
- Type of Component (surface, rail, station)
- Type of Software (built-in vs specific external software)
- Link map to other components (doorway and platforms and the component they connect with)

Generally, the component network structure data should only be stored within the coordination engine, and for use by analysis software, with individual simulator component software not requiring this information, keeping the complexity of the distribution of system components hidden.

#### Transit Network

To allow for consistency across the network, the data structures used to specify the overall transit network needs to be standardized. This permits individual components to be aware of how they fit in the overall network structure, including the global identifications of local elements (doors, platforms). A shared structure also enables components that generate elements on-the-fly, for example routes and stops, particularly when network changes occur. The network structure data is stored in a series of related collections, illustrated in Figure 3.6.
3.5 Agent Framework

The agent framework defines the process agents take in deciding on their movement through the system, and specifies the set of agent data and agent behaviour required to maintain coherence as they are shuffled between components. Unlike the simulation and communication frameworks, the agent framework is defined much more loosely, with a focus on the agent data structures transferred during simulation rather than defining a behavioural architecture. The behavioural architecture is, instead, the purview of the transit assignment method utilized. It can range from only choosing the quickest schedule route to a full 'mental model' containing various possible paths from each transfer point and incorporating memory of prior trips found in learning-based assignment methods like MILATRAS [64].

Regardless of assignment method, agents are defined in the framework to go through a specified process prior to simulation. They are assumed to initially be input only with an origin, destination, either a specific departure time or intended arrival time, and access and egress modes, with population generation conducted by an external source. Initial processing is performed by the transit assignment module, following the steps detailed in Section 3.4.1. The end result are agents with a chosen path or path network for use. To allow for the widest level of compatibility, this path information is kept within the transit assignment module, with the next step in the trip provided on demand; this enables all network-level behavioural models to be confined to the transit assignment component. Basic agent properties like demographic information, walking speed, etc. are specified or generated within the transit assignment module using provided distributions. This information is provided to relevant components prior to the start of simulation in order to maintain coherence of agent properties across the network modules.

Agent data during simulation is passed via the various component interfaces as defined in the simulation framework. This occurs in three ways:

- Transfer from transit assignment to street simulator at beginning of trip
- Transfer at synchronization points when entering/exiting through doorways
- Along with vehicle data in between synchronization points

Figure 3.6: Data structures defining transit network service
While this setup could result in the possibility of entrance/exit times not being precise in the latter case, as the framework has a focus on large scale crowd movements, the extra signalling required to have precise agent transfers at doorways was deemed to not be of concern. It is also expected that any delay resulting from the synchronization period will be significantly less than the waiting time experienced by agents after they make their way to vehicle stops.

A summary of the data transferred using these various methods is shown in Figure 3.7. For the purpose of the prototype implementation, access and egress modes were assumed to both be walking; a more proper implementation should relax this constraint.

![Figure 3.7: Data structures defining transfer of agent information](image)

### 3.6 Communication Framework

The communication framework specifies how control signals and system feedback make their way through the simulation framework. A generalized version of this was shown in Figure 3.4, with more detail presented in this section.

This framework specifies the network control capabilities of the system in affecting change in particular components. Generally, control of system components can be divided into four categories. The first category involves control signals that are specified before simulation occurs and specify initial settings. The second are signals that occur during the simulation as part of the normal control of components necessary for simulation. These could include signals to control train doors within station or train movement within line operation. The third are signals that occur during simulation that cause extraordinary changes, specifically strong deviations from regular operation and might require changes to transit paths for large numbers of agents and/or changes in transit service. The final type of signals are those that occur following simulation completion, and specify any final analysis or data above the raw data files that are sent at the end of the run. The first and second categories of component control are not considered...
within this communication framework, instead handled by the simulation framework as part of general operation.

The communication framework deals, instead, only with the third and fourth category of control. As such, it deals with signals meant to modify normal operation, enabling the unique capability of the framework in simulating various operational conditions. This can be thought of as both mimicking the behaviour of a transit control centre (modification vehicle trips, addition of shuttle routes, etc), as well as unforeseen disruptions to service (e.g. train breakdowns). These signals, therefore, are aimed at both causing disruptions and changes in transit service to respond to them. Also covered are information signals that allow for network service information to be distributed to agents in specific components (e.g. via a PA system on a train or informational signs), that could trigger responses in agent behaviour. The latter is meant to allow for the system to be used to both test the effects of the provision of information to transit users, and the best way to implement transit information technologies.

For flexibility and ease of use, the framework divides these control signals into two stages (illustrated in Figure 3.8). Main control signals are sent from the software driving the simulation (the network analyzer in the framework) to the coordination server. These signals are limited to a specific set of instructions that are to be implemented by the coordination server. While the exact set will depend on the computing implementation of the framework, but generally will be commands relating to the various types of disruptions (e.g. mechanical problem with a specific train) or response plans (e.g. institute bus shuttles).

Actual implementation requires the coordination server to translate these more abstract commands to a set of specific signals to one or more components. For example, a mechanical train issue command would require the coordinator to send:

- A signal to the appropriate line simulator to take a train out of service at the next station
- A command to the transit assignment module to add a trip segment for each through-passenger to board the next train at the drop-off platform

To allow for such flexibility, translation data packets are specified to be as generic as possible within the computing implementation, with network component modules coded to properly interpret them, specific to their own properties.

The second function of the communication framework is to handle feedback from sensors placed in components around the network. Sensors in the framework are intended to mimic physical sensors; for example they could include vehicle location, platform volumes or flow through stairs and escalators. Each component registers available sensors with the system, allowing the coordination server and network

![Figure 3.8: Stages of control signalling](image-url)
Chapter 3. System Frameworks

Analyzer components to retrieve their values, either for reporting of results or as input to be used to enact response measures. This registration data should include at a minimum a unique ID identifying the sensor within the component, its name and the type of information. The exact data reported and format is implementation specific, but generally should include the time of the sensor data and its value. This data is then logged within the output tables, and sent to the analyzer if requested.

In the initial prototype of the framework developed for this dissertation, this ideal communication framework was not fully implemented. It was instead limited mainly to a simplified implementation of the sensor-feedback system, with allowance provided for a limited set of feedback values, with disruption and response measure hard-coded (described in Chapters 6 and 7).

3.7 Conclusion

This chapter presented the details of the Nexus frameworks that were devised to target issues in currently available software, and to enable the research conducted in this thesis. To date, transportation network simulators have generally attempted to incorporate all facets of transport systems, predominantly with a focus on the flow of autos. Given the intricacies present in the various components of transit systems (in particular rail and stations), network simulators generally do a poor job in providing the tools necessary to properly model transit networks. In addition, building large scale networks in such software necessitates having to build entire cities within single project files, also limiting the computational power that can be brought to bear. Finally, these individualized efforts have produced several packages, each with high degrees of similarities and different areas of deficiencies.

The computing framework presented in this chapter took a very different approach. Instead of developing yet-another traffic, rail or pedestrian simulator, the goal of this work has been to instead develop a platform to allow for current or future leading-edge software in each of these areas to be connected together to form a transit network simulator. The network was divided into its constituent parts (surface, separated rail and station), with the specific tasks and i/o of each clearly specified. This permits large scale networks to be built up in pieces, with the requisite amount of detail and complexity for each aspect of the network at the level set by the modeller. As an added bonus, the distributed nature of this design permits the network to be run across multiple computers; this should allow for larger and more detailed models to be simulated.

The agent and communication frameworks described in this chapter move the platform from solely a method to allow for interoperability between software to a coherent agent-based dynamic transit network simulator. The agent framework separates the primary decision making of agents to a defined transit assignment component. The interface layers, which mediate communication between the main system and external software, are then responsible for appropriately translating agent information and decisions to link their local behaviour and routing with their overall intentions as they move through the network. The communication framework performs the same task but for the transit service. It provides the foundation for signals to be sent to control important aspects (necessary for run-time modification to service), and signals to be received from virtual sensors placed throughout to acquire a detailed understanding of the performance of all aspects of the network.

The next several chapters of this thesis operationalize this framework. The following two chapters (4 and 5) present research of two important models of pedestrian behaviour at important areas of the station (level transition and platforms). The model of platform behaviour, in particular, is a key illustration
of how the framework enables the development of models contingent on having a network model with requisite detail at the interface between components, and network-aware agents. Next, chapters 6 and 7 describe the two major steps taken in the implementation of Nexus, while incorporating the developed pedestrian models. Also demonstrated is the ability of Nexus to successfully interface with MassMotion to allow it to act as a station simulator where such detail is required.
Chapter 4

Modelling Vertical Circulation Choice

As one of the main bottlenecks in subway stations, flow profiles at high volume vertical circulation (VC) locations (stairs and escalators) are key determinant of overall station performance. During peak periods where capacity is strained, and train frequency is highest, poor flow through these bottlenecks may affect pedestrian positioning and volumes on platforms, affecting dwell operations on trains. This is particularly true for stations that facilitate transfers between subway lines. As a result, undertaking research to better understand how pedestrians made choices with respect to vertical circulation facilities was believed to be important with respect to modelling the transit network. Due to the limitations in the ability to collect data on pedestrian movements at longer distances as they made their way through the entire station, research was concentrated on co-located adjacent stair-escalator pairs.

Part of the work was motivated by a goal of devising equations that could be more easily applied by transit practitioners to determine flow across the various facilities connecting floors. There was a special interest of accounting for interactions between adjacent facilities of varying type, rather than treating them independent, as is done in the current Transit Capacity and Quality of Service Manual. The second motivator was the development of improved models of pedestrian behaviour to be applied within MassMotion. This initial study into adjacent stair-escalator facilities is a first step towards a broader understanding of routing through subway stations. With respect to Nexus, a key driver was the ability to examine how changes at primary crowd bottlenecks within stations could have potential impact beyond the boundaries of the individual station. As a result, the research presented in this chapter led to analysis of the impact of implementation of these models on overall network and agent performance, dealt with in Chapter 8.

This chapter details the research performed to model pedestrian VC choice. First, a summary of existing studies that dealt specifically with vertical circulation is presented. This is followed by a description of the study area, the locations and times of observation and the data collected. The three studies are then detailed, including the developed models, and the validation process and results. To close out the chapter, the conclusions of all studies are presented.
4.1 Current Literature

As detailed in Section 2.4, pedestrian modelling is a relatively new area in the field of transportation modelling, with its emergence beginning in the late 1980s with the development of the Social Forces and Cellular Automata motion models. While the main research efforts have focused on these motion models, recent studies have formulated pedestrian-specific models of route choice. With respect to the use of stairs and escalators (which can be thought of as a mode choice within a pedestrian system), the use of discrete choice models has been popular in recent years. Nevertheless, research in this particular section of pedestrian modelling has been relatively sparse, limited to a few applications of logit techniques specifically attempting to predict choice between adjacent facilities [25, 75, 76] and broader studies examining vertical choice in the context of overall route choice in stations [25, 77, 78]. While recent studies have aimed to move away from simple logit formulations to more advanced mixed-logit structures allowing capturing of heterogeneous taste variation [77, 76], these have been limited in application.

The earliest effort was in 1998, where Cheung and Lam investigated pedestrian choice between escalators and stairs in six subway stations in the Hong Kong Mass Transit Railway [25]. The facilities under observation were adjacent (stair beside escalator), with the ability to approach from either side, and were of standard physical dimensions across sites. Data was collected for pedestrian movements in both directions on the staircases and escalators, as well as the walkways leading to the facilities. In formulating their models, the researchers considered that only perceived travel time would influence a persons choice of facility. It was in-turn assumed that individuals would have an internal model of expected travel time, which they would then use to mentally estimate travel time differences between stairs and escalators in making their choice [25]. The choice between stair and escalator was modelled as a logistic function. The parameter values indicated varying sensitivity to delays in the choice to use the escalator, with increased sensitivity in the descending direction. Likewise, in the ascending direction, pedestrians were found much more apt to choose the escalator and less sensitive to escalator delay [25]. However, the use of perceived travel time makes the model difficult to apply to other situations without extensive data collection, and ignores other variables that could influence choice, particularly mobility and physical effort.

Several years later, in contrast to the prior research, Daamen et al focussed on the influence of the presence of different types of vertical circulation on the route taken by pedestrians navigating a subway station. Instead of examining pedestrian choices only in the area surrounding the vertical circulation, the researchers followed individuals in two Dutch railway stations throughout station facilities. Route choice was examined under various configurations of trip length and trip factors, in particular those related to vertical routing (stairs, escalators, ramps), while ignoring any issues of congestion a priori as not being significant. Data was collected in late fall and winter, and included route characteristics (length, type, number of turns), personal characteristics (gender, age, luggage, familiarity), trip characteristics, and other factors such as the day and weather conditions. However, the mathematical model (a multinomial logit with a path-size variable added to handle overlapping routes), used only observed travel time on each of the segments of the trip (levels, stairs, ramps, escalators) and the direction of travel. [77]

Most recently, two research groups refocused on the question of adjacent stair-versus-escalator choice. Zhang et al investigated choice of pedestrians between stairs and escalators in three stations in Nanjing, China with varying heights and escalator directions. Data collection occurred during morning and afternoon peak travel with facilities that had a range of physical dimensions. Data collection focussed on the physical dimensions of the facilities and the pedestrian-specific information of gender, age, walking
distance and walking time. A disaggregate binary logit model was used, with utility functions for stair or escalator use incorporating the aforementioned walking distance, walking time, gender and age, along with a dummy variable to represent inherent bias towards either mode. Gender was generally not found to be significant, age was found only to be significant for a single station (marginally), while walking distance was also not significant in the downward direction for all stations. In addition, a consistently positive and significant value for escalator bias was found, illustrating a key preference. [75]

Lastly, one of the most recent studies dealing with examination of pedestrian vertical circulation choice was conducted in Austria. The scope of the data collection effort was relatively small compared to some of the previously mentioned studies, focussing on a single station, the Westbahnhof. The researchers observed some traditional variables (age, gender); however, in lieu of actual travel time, a revealed preference survey was conducted by interviewing people after they had ascended their choice of facility to acquire their trip purpose, frequency of visit to the location, self-reported walking speed and education level. In addition, an attempt was made to also consider some dynamic factors in choice, by taking note of the number of people queuing at the base with and without luggage. In addition to this revealed preference (RP) study, the authors investigated the use of the stated preference (SP) method in determining how people choose between the two facilities by showing them six different video sequences of the same location with different levels of crowding and asking them to choose between using stairs or escalators. The study results showed a severe overestimation by individuals in their claim to use the stairs (SP) compared to what actually occurred (RP). In addition, only the level of luggage-based queuing was found to be significant for the RP study, while differences in perception were found across age, perceived speed and general queuing. [76]

More aggregated regression methods have also found their place in relating rates of stair and escalator use to physical variables. Simpler in use and application, these techniques are of particular interest in the health and well being field, where there is a wish to promote stair use to improve overall health. They, however, have elucidated some interesting behaviour and relationships with respect to pedestrians and stair-escalator choice. Of particular note are findings of behavioural mimicry within stair/escalator choice (pedestrians are more likely to use stairs if observing others using stairs when they arrive) [79], diminishing return of stair usage with increasing stair width [80], and the ability to influence choice with motivational signs [81, 82, 83].

4.2 Data Collection Process

It was initially hoped that existing datasets may be available for use during this research. However, it quickly became apparent that the level of detailed information required to complete the research project would not be readily available off the shelf. CCTV footage offered one data source, but there are barriers to using such data, not least the fact that most CCTV cameras rove or swivel as required by security and station operations staff, making the retrieval of continuous datasets impossible. It was therefore decided at an early stage to gain approval from the TTC to complete video survey work on the Toronto subway system.

4.2.1 Toronto Transit Commission(TTC) Network

The TTC subway network (Figure 4.1(a)) consists of 69 stations spread across 4 subway lines, serving almost 900,000 passengers daily. It is fully integrated with TTC bus, streetcar and Wheel-Trans networks.
Chapter 4. Modelling Vertical Circulation Choice

(a) TTC Subway Network

(b) Bloor Station

(c) St George Station

Figure 4.1: Observation locations along the TTC subway network and two example locations

with daily patronage of approximately 1.6 million passengers. The stations vary in layout and design, but generally consist of a platform level, a surface/bus interchange level and an intermediate concourse level. Multiple stair, escalator and elevator options exist at most stations for passenger transition between levels. Stairs and escalators are frequently paired. Within the TTC system, ramps are not generally provided as an alternative to stairs, escalators or elevators and this is typical for most similar transit properties.

4.2.2 The Data Collection Plan

Data was collected at six stations. These included the three busiest stations in the network, located in the downtown core, Bloor, St. George and Union, and three suburban stations, Finch, Downsview and York Mills. All platform vertical transitions were surveyed for passenger flow (with the Bloor Line platforms selected at Yonge and the Spadina line platforms selected at St George), as well as transitions
between the upper concourse levels for Finch, Downsview and York Mills. A summary of the facilities where video was recorded is presented in Table 4.1.

**Table 4.1: Summary of Vertical Circulation Facilities Observed**

<table>
<thead>
<tr>
<th>Facility Location</th>
<th># Esc Lanes</th>
<th>Stair Width (cm)</th>
<th>Step Height (cm)</th>
<th>Number of Steps</th>
<th>Total Height (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>2</td>
<td>116</td>
<td>17</td>
<td>20</td>
<td>3.4</td>
</tr>
<tr>
<td>St. George</td>
<td>2</td>
<td>180</td>
<td>16</td>
<td>29</td>
<td>4.64</td>
</tr>
<tr>
<td>St. George</td>
<td>2</td>
<td>174</td>
<td>16</td>
<td>29</td>
<td>4.64</td>
</tr>
<tr>
<td>Bloor</td>
<td>2</td>
<td>180</td>
<td>15</td>
<td>33</td>
<td>4.95</td>
</tr>
<tr>
<td>Bloor</td>
<td>2</td>
<td>172</td>
<td>16</td>
<td>32</td>
<td>5.12</td>
</tr>
<tr>
<td>Bloor</td>
<td>2</td>
<td>170</td>
<td>16</td>
<td>32</td>
<td>5.12</td>
</tr>
<tr>
<td>Bloor</td>
<td>2</td>
<td>175</td>
<td>15</td>
<td>33</td>
<td>4.95</td>
</tr>
<tr>
<td>Downsview</td>
<td>2</td>
<td>193</td>
<td>17</td>
<td>32</td>
<td>5.44</td>
</tr>
<tr>
<td>Downsview</td>
<td>2</td>
<td>212</td>
<td>16</td>
<td>23</td>
<td>3.68</td>
</tr>
<tr>
<td>Downsview</td>
<td>2</td>
<td>215</td>
<td>16</td>
<td>22</td>
<td>3.52</td>
</tr>
<tr>
<td>Downsview</td>
<td>2</td>
<td>215</td>
<td>17</td>
<td>24</td>
<td>4.08</td>
</tr>
<tr>
<td>Finch</td>
<td>2</td>
<td>216</td>
<td>17</td>
<td>17</td>
<td>2.89</td>
</tr>
<tr>
<td>York Mills</td>
<td>2</td>
<td>173</td>
<td>16</td>
<td>39</td>
<td>6.24</td>
</tr>
<tr>
<td>York Mills</td>
<td>2</td>
<td>164</td>
<td>16</td>
<td>39</td>
<td>6.24</td>
</tr>
<tr>
<td>York Mills</td>
<td>1</td>
<td>164</td>
<td>17</td>
<td>78</td>
<td>13.26</td>
</tr>
</tbody>
</table>

Data collection was completed at each station on Saturday April 14th, Tuesday April 17th and Wednesday April 18th of 2012. Observation occurred over a 15-minute period in each of the weekday morning and afternoon peaks, as well as during the Saturday afternoon shopping period. This spread of time periods was intended to give a wide distribution of pedestrian flow densities and peak flow directions. Two methods of data collection were used. Video recordings were completed at two vertical transitions per station using hand-held devices. Video locations were chosen where stair and escalators in immediate proximity afforded passengers a mode choice between the two. Footage was subsequently reviewed manually to retrieve data. Manual counts were also completed at the remaining vertical transitions where video was not recorded using smartphone applications (Tio and Advanced Tally Counter for the iPhone and Android platforms, respectively). The video data allowed for the subsequent retrieval of a finer grain of information (personal characteristics and precise time measurements) than was possible using manual count techniques, and it is primarily this dataset that has been used during subsequent data analysis and model formulation.

Elevators serve all TTC platforms but are primarily intended for use by passengers with restricted mobility. They have usable capacity of between 5 to 8 passengers per car and are relatively slow with a travel speed of approximately 0.5 metres/second (compared with typical elevator speeds of 1.5+ metres/second for transit station elevators that are designed as the primary means of vertical circulation).

### 4.3 Aggregate Models

As a first stage of research into vertical circulation choice, the aggregate model targeted a gap in the body of knowledge of a simpler technique for agencies to predict how vertical facilities will be used,
taking into consideration attributes not previously considered, such as opposite flows.

4.3.1 Data Extraction Process

Data extraction from video was performed manually, due to the lack of available software capable of automating the process. The data collected included the time of entry into the facility, the route choice between stair or escalator, and the type of passenger using the London Underground Person with Restricted Mobility (PRM) categorization, which include categories for wheelchair users, luggage, strollers, and other disabilities [84]. In contrast to previous studies, it was believed that grouping by this method would be most applicable for mode choice of stair or escalator compared to traditional demographic categories (age, gender, etc.). In addition to these demographic categories not being found significant in prior research, PRM categorization is easier to accomplish and less prone to subjectivity. Identification of the PRM category was performed by inspection from the recorded video for each individual. Across the six stations and 14 stair/escalator pair locations, over 25 thousand individual pedestrian choices were extracted from the video data.

Each pedestrian was time stamped to the nearest second of arrival at vertical facilities. For the purpose of our analysis, a 10-second aggregation period was considered the most appropriate to use. This duration was chosen to be significant enough in length at which application could still be practical, but not too long as to average out the time-varying escalator-stair splits during surges of use. For each 10-second period, the following values were determined: the direction of interest (escalator direction), total using both stairs/escalators, number on each facility, the total of each PRM category, the number opposing in the opposite staircase direction, the direction of approach (escalator, stair, or both/neither), and the physical dimensions of the location (# esc lanes, stair width, total height). A total of 2543 data points were produced in this manner.

4.3.2 Explanatory Variables Considered for Model Estimation

The data was compiled in a form compatible with model estimation, consistent with the following explanatory chosen variables:

**Direction of Movement** A binary variable, representing pedestrians moving either in the ascending or descending direction. The direction of movement was specified as the direction of escalator flow. However, based on prior research and general observations that have showed a significant difference in the parameter values of other variables for pedestrians moving in ascending or descending directions, the data was separated into two for the development of separate models for each direction.

**Adjusted Input Flow Rate per Lane (AIF)** A continuous variable representing the utilization of the input channels (lanes) into the escalator in the 10-second period (people/lane), adjusted by the percentage of capacity provided by stairs. It was calculated by dividing total flow by number of escalator lanes, followed by a multiplication of the ratio of stair lanes to overall lanes. The rational behind this adjusted rate is based on the observed behaviour that stair use occurs normally after the escalator becomes crowded. In addition, the propensity for larger stair % use at higher flows will also be dependent on the relative capacity of stairs versus escalators. Escalator lanes were taken as observed (usually 2 lanes), while the number of stair lanes were determined by dividing the stair width by 75cm (the accepted lane width described in the Transit Capacity and Quality of Service Manual) [85].
**Opposing Flow Rate per Stair Lane (OFR)** A continuous variable representing the opposing flow on the staircase per lane in the 10-second period (people/lane). OFR was calculated by dividing opposing flow in the period by the number of stair lanes.

**Total Height** A continuous variable representing the height in metres of the facility (stairs/escalator).

**Direction of Approach** A discrete, three-level variable representing the observed direction of the entering group of pedestrians during each 10-second interval. The variable was assigned one of three values: escalator, stairs or none. A group was indicated to have arrived from either escalator or stair direction if the majority (90%+) were observed to be arriving from that direction (i.e. train arriving from one side), or none if there was no dominant direction of approach.

**PRM %** A continuous variable representing the percent of the individuals in the flow in the direction of interest that were categorized as having restricted mobility. The data showed that of the time segments with PRM groups, very few did not carry heavy luggage; as a result, the PRM groups were combined together into one group.

### 4.3.3 Modelling Framework

The elevators in use on the TTC system are primarily low-capacity and low-speed units intended to provide step-free access for those that need that facility. They are not intended to provide a primary means of vertical circulation. As such, the aim of the target model is to predict the expected escalator-versus-stair use where such facilities are co-located. The form of the model incorporates varying levels of pedestrian flow, mobility group representation and opposing flow, while taking into account the physical dimensions of the vertical facilities. Dealing with a situation where the choice is binary (either escalator or staircase), the appropriate model type is a logistic function. The logistic function has the property of being confined within the values of 0 and 1, which are natural limits for the percentage escalator use. In addition its S-shaped form is of the expected response form, where escalator use is heavily favoured, eventually dropping after a certain threshold of utility is reached, and plateauing at the split at which both facilities are at capacity. Generally, logistic models are used to provide probabilities of a choice made between two options. In the case of this study, while the unit of examination is the % split for 10-second segments, this percentage emerges from the individual choices of people making a decision within the current environment, so it can be interpreted as the average probability. The functional form is specified as shown below:

\[
P_{esc} = \frac{1}{1 + e^{-X}}
\]

where \(X\) is analogous to a utility function and has the following general form:

\[
X = \sum_i B_i x_i
\]

Here, \(x\) is the value of the \(i^{th}\) explanatory variable (utilization, height, PRM groups, etc.), and \(B\) is the parameter value (weighting) for each explanatory variable. Higher positive values of \(X\) result in higher probabilities or higher splits of escalator use. Significant variation was found in the stair/escalator splits for flows consisting purely of individuals without any restricted mobility. As a result, any trend resulting from increasing levels of those with
mobility restrictions was believed to be masked by this noise, exacerbated by the much larger data set for segments with no PRM groups (1500 vs. 350). To handle this, the dataset was segmented into two groups, those with and without PRM groups, which were estimated separately. The overall modelling framework, therefore, consisted of two logistic functions for each direction, for a total of four models.

### 4.3.4 Model Estimation

For these four models, estimation was conducted in the software package R using the internal functions for generalized linear modelling under the assumption of a binary choice set (number choosing escalators and stairs for each 10 sec period). This function uses the iteratively reweighted least squares (IRLS) method to find the estimates of the parameters. Based on the produced results, a step-wise backward elimination of variables was conducted to remove non-significant parameters. The resulting models are now presented, with their statistical information for the final parameters (Table 4.2). In all cases, a p-value of 0.05 was taken as the cut-off threshold for parameter significance.

|                   | Estimate  | Std. Error | z value | Pr(>|z|) |
|-------------------|-----------|------------|---------|----------|
| **Ascending, PRM Free Model** |           |            |         |          |
| (Intercept)       | 2.39741   | 0.07563    | 31.701  | < 2e-16  |
| AIF               | -0.25839  | 0.01096    | -23.578 | < 2e-16  |
| OFR               | 0.27883   | 0.02407    | 11.586  | < 2e-16  |
| Esc Approach      | 0.47125   | 0.0895     | 5.298   | 0.000000117 |
| **Ascending, PRM Model** |           |            |         |          |
| (Intercept)       | 1.80151   | 0.19336    | 9.317   | < 2e-16  |
| AIF               | -0.23925  | 0.02979    | -8.032  | 0        |
| OFR               | 0.37368   | 0.04974    | 7.512   | 0        |
| Esc Approach      | 0.6665    | 0.11079    | 6.016   | 0.000000002 |
| PRM%              | 1.23823   | 0.609      | 2.033   | 0.042    |
| **Descending, PRM Free Model** |           |            |         |          |
| (Intercept)       | -0.20671  | 0.17442    | -1.185  | 0.236    |
| AIF               | -0.18932  | 0.01714    | -11.045 | < 2e-16  |
| OFR               | 0.30526   | 0.04047    | 7.543   | 0        |
| Esc Approach      | 0.73002   | 0.12481    | 5.849   | 0.000000005 |
| Stair Approach    | -0.75142  | 0.12673    | -5.929  | 0.000000003 |
| Total Height      | 0.34052   | 0.03903    | 8.724   | < 2e-16  |
| **Descending, PRM Model** |           |            |         |          |
| (Intercept)       | -0.58415  | 0.33735    | -1.732  | 0.08334  |
| AIF               | -0.10986  | 0.03563    | -3.083  | 0.00205  |
| OFR               | 0.33066   | 0.08264    | 4.001   | 0.0000631 |
| Esc Approach      | 0.57014   | 0.13535    | 4.212   | 0.0000253 |
| PRM%              | 1.39811   | 0.70162    | 1.993   | 0.0463   |
| Total Height      | 0.29501   | 0.06297    | 4.685   | 0.0000028 |
Chapter 4. Modelling Vertical Circulation Choice

Ascending Direction for PRM-Free

For the ascending direction, examining solely periods where only those without any mobility restrictions were present, a high positive intercept, AIF, OFR, and the approach of pedestrians from the escalator side were found to be significant.

\[ X_{ascPRMN} = 2.39 - 0.25 \times AIF + 0.28 \times OFR + 0.47 \times esc \text{ approach} \] (4.3)

Ascending Direction for PRM

For the ascending direction, examining solely periods where only those without any mobility restrictions were present, a high positive intercept, AIF, OFR, and the approach of pedestrians from the escalator side were found to be significant.

\[ X_{ascPRM} = 1.80 - 0.24 \times AIF + 0.37 \times OFR + 0.67 \times esc \text{ approach} + 1.24 \times PRM\% \] (4.4)

Descending Direction for PRM-Free

In the descending direction for sections with purely mobility restricted free individuals, the intercept value was found to be significant and negative; however, the addition of the total height of the facility being positive and significant for this direction results in still a positive escalator bias for the heights examined. Approach from the stair side was also found to decrease escalator use.

\[ X_{descPRMN} = -0.21 - 0.18 \times AIF + 0.31 \times OFR + 0.73 \times esc \text{ approach} - 0.75 \times stair \text{ approach} + 0.34 \times height \] (4.5)

Descending Direction for PRM

Examining the corresponding periods with PRM individuals present, like what occurred with the ascending direction, the same variables were significant as with the non PRM periods, but with PRM% as an added variable.

\[ X_{descPRM} = -0.58 - 0.11 \times utilization + 0.33 \times opposing \text{ density} + 0.57 \times esc\text{ approach} + 1.40 \times PRM\% + 0.30 \times height \] (4.6)

Model fit is provided by the residual deviance for logistic regression fits, as well as the change in deviance between the null and final models; evaluation is performed by comparison to the appropriate chi-squared distribution. Residual deviance was not found to be significant (indicating good model fit) for three of the four models (both in the ascending direction and the PRM model in the descending direction). In all four cases, however, the change in deviance was significant. This indicates that the variables contained in the model did have a significant impact; however, in the case of the model for the non-PRM time sections in the descending direction, either other variables are required to better explain the choice made, or the level of randomness or noise is too high for accurate prediction of facility use.
4.3.5 Analysis

Parameter Analysis

**Intercept/Innate Escalator Bias** As shown in these models, a clear distinction was found between the upward and downward directions with respect to the intrinsic preference of escalator or stair. A strong preference to use the escalator was found in the ascending direction, while for the descending direction no significant innate bias was found. This is believed to result from a combination of the higher effort required for ascending versus descending the staircase, a key factor considered by pedestrians during route choice [77], and a sensitivity to height in the descending direction.

With respect to identified variables, within all estimated models, it was found that the utilization of the input channels into the combined stair/escalator facility, the opposing density on the staircase and whether the individuals entered from the escalator side were significant factors.

**AIF** Increased AIF had a consistently negative parameter across the four models, indicating that increased inflows result in higher expected stair use, as would be expected as the escalator becomes more crowded. Its impact was felt more strongly in the ascending direction, in part due to a lower preference for the escalator in the descending direction, which would reduce the ability of the flow rate to influence an already more even choice. It should be noted that because of the structure of the AIF, the model remains sensitive to both changes in escalator capacity (increases resulting in higher escalator use) and stair capacity (increases resulting in higher stair use).

**OFR** Opposing flow on the staircase was found to have a strong positive (escalator-favouring) parameter value, as would be expected (individuals are less likely to use a staircase if being opposed). The impact was seen to be particularly higher in the ascending direction for the PRM containing periods, with a greater propensity to avoid ascending the staircase.

**Direction of Approach** A similar result was seen for the variable representing the direction of approach. A positive value was found for the parameter representing approach from the escalator side (increasing the escalator split), and it was again higher in the descending direction. In the descending direction, approaching from the staircase side was also seen to increase the percent staircase usage. It is believed that this is due to the reduced affinity for the escalator in the descending direction.

**PRM %** Comparing the PRM versus non-PRM models, the key difference was the inclusion of the term representing the percent PRM in the total flow, with a significant positive value. It would be expected that as the PRM composition increased, that the affinity to use the escalator would also increase, and as such, the result is intuitive and expected. Given the aforementioned relatively small number of individuals within the PRM category and the high number of purely PRM-free time periods, this trend was only observed when the dataset was segmented into the two groups. It would be more beneficial if a single model could be used to predict splits instead, and as such more data collection is required of these periods containing PRM individuals.

**Total Height** Total vertical height was found to play a role in increasing escalator use, but only in the descending direction. It was expected that this would be the case for both situations; however the significantly higher escalator preference in the ascending direction, which remained high for all heights, indicate an associated lack of sensitivity. The range of heights tested was, however, somewhat limited and clustered around 5 metres of rise; therefore, it is unclear whether a statistically significant
impact in the ascending direction would be observed if a broader range of observations were made. It is certainly true that at, for example, 10 metres of vertical rise, more passengers may be expected to queue at the base of a busy escalator than they might do if the rise was only 5 metres.

Variables of Influence Not Considered

While several variables were considered for modeling stair-escalator split, a few others that have potential influence were not included. The most significant of these was the time of day, specifically whether pedestrian choices were observed during off-peak, morning peak or afternoon peak periods, where behavioural factors (e.g. being in a rush, being tired, etc.) may lead to different choices. The situations observed, however, did not allow for separate time-specific models to be developed, with the TTC setting the vast majority of escalators to the upward direction during the morning peak period. Nevertheless, additional dummy variables were introduced to represent this as a shift in preference for either stairs or escalators; however, significance was not found for any of these parameters.

Other variables, such as step-height, weather, visibility, and other more complex measures of the facility configuration were also eschewed for a few reasons. First, the situations examined presented a natural limitation in the range of observed variables (for example the stations had essentially uniform step heights, and data was collected within the same season on sunny days). Second, an assumption was made that choice occurred near the entrance, where visibility and other surrounding physical layouts would not be of significance. Lastly, the number of variables was minimized to maintain simplicity in model application. Future investigation might warrant examining whether these factors play a significant role; however, that would necessitate a larger data set, specifically including stations outside of Toronto.

Variable Choice and Model Comparison

Prior research has generally focussed on demographic factors and estimated travel time (or difference in travel time) as predictor variables. While demographic factors have had minimal success in such prediction, travel time has been the predominant variable. In addition, model types have mainly been individualistic and discrete choice in nature. In some cases, the effects of congestion were deliberately ignored through examination of non-crowded stations. In others, these more dynamic factors in models were incorporated, but resulted in models that are more difficult to apply, requiring the use of pedestrian simulators. In more aggregate models, as performed in the first detailed investigation with 30-second aggregation [25], it was assumed that individuals had a known understanding of the expected travel time up stairs and escalators. This expected travel time was modelled as a BPR (Bureau of Public Roads) function taking only incoming flow and free flow speed into consideration, and choice was based on this mental picture of travel time [25]. However, while this might be appropriate for simpler situations, its application in transfer stations, where significant counter-flows are present is questionable, as is its ability to incorporate other factors, like changing height, which take into account more behavioural responses to the environment (such as increased effort).

The models in this study, on the other hand, attempted to use a more expanded variable set to act as a surrogate of the perceived attractiveness of escalator versus stair usage. This expanded set allow the models to be adapted to a wider variety of situations, while limiting the variables chosen to those that could be easily known (incoming and opposing flow, physical dimensions of the situation, direction of approach and the mobility composition of the population). Its use, however, given the period in question is kept fairly small (10 seconds), is dependent on being able to be aware of the general shape of the
arrival curve into the facility for various trainloads. An explanation and method of doing such analysis in estimating the expected use of vertical facilities is provided in a later section.

4.3.6 Model Validation

Validation of logit-type models is often performed by two methods, either setting aside a portion of data (e.g. 20%) and estimating on the remaining data, or in the case of smaller datasets, performing estimation on the entire data set and attempting to use the model to recreate the entire input, but analyzing the level of success in predicting particular splits or choices. Given the aggregation of data and segmentation conducted in this study, resulting in somewhat low levels of data for certain segments, the latter method is used here. In addition, the ability of the model to reproduce the overall escalator split for particular train pulses observed was also examined to check viability in application and as an added step of overall validation.

For initial validation, the ability of the model to predict the splits observed in specific 10-second intervals was examined. This involved comparing the probabilities determined by the four models against the actual splits from the input data. The resulting histograms of the differences (predicted - actual) are shown in Figure 4.2.

As seen in the figure, the models for the ascending direction are much more apt in predicting actual splits, with some issues in over-predicting the level of escalator use. As would be expected, the residuals for all models are distributed about 0, with the ascending direction models displaying a normal shape, indicating an appropriate fit. At a significance threshold of 5%, the models are able to determine the actual splits 77% and 81% of the time for the PRM and PRM-free sections, respectively. On the other hand, for the descending direction, some issues in predictive accuracy are apparent. While, the PRM section model has moderate spread of residuals about the zero level, for the PRM free sections, a broad distribution is found, attributed to the much more random behaviour of pedestrians in the downward direction when making a choice between stair and escalator. In addition, while the shape of the residuals still follows a predominantly normal distribution, a minor but elevated level is seen at higher positive residuals, where the descending model over-predicted escalator use. This follows the lack of significance of the overall fit found for the descending direction, PRM free model previously detailed. Another issue is the relatively lower level of data (about half) available in the downward direction compared to the ascending direction. However, its more balanced under and over prediction may allow the descending PRM free model to still provide decent results when examining larger timeframes; this is examined in the next section.

Predicting use of escalator and stairs for train passenger alighting pulses

The presented models use 10-second periods of aggregation. This reflects a balance between the need to create a viable and applicable model (greater aggregation desirable) and the need to incorporate the changing dynamics of route choice (less aggregation desirable). Many practitioners may, however, wish to apply the models at higher levels of aggregation, for example, at a level of aggregation that reflects a complete train arrival pulse. Therefore, the ability of the model to predict usage on a train-by-train (or surge-of-pedestrians) basis was examined by isolating these surges of flow within the collected data, predicting escalator use for each component 10-sec period, and determining the overall split. The deviation between the predicted percentage escalator split and the actual percentages were then
Figure 4.2: Prediction error of estimated aggregate vertical circulation models (ascending direction with (a) and without (b) PRM individuals, and descending direction with (c) and without (d) PRM individuals) for 10 second segment choice splits.

The resulting histograms (Figure 4.3) show better predictive value for the ascending direction compared to the descending direction. There was an average error of +0.1% with 91% of predictions within 10% of actual escalator use in the ascending direction, compared to +5% error with 74% of predictions within 10% for the descending direction. The largest deviations in prediction for the descending direction, however, occurred for a single station, where stair use was abnormally high; this increased the average over prediction from 3% to 5%. This location was somewhat unique in the observed set with a long hallway leading to the facility. On the whole, however, the presented models, even with their aggregate nature, were able to perform quite well given the highly dynamic nature of stair and escalator choice at stations.
4.4 Importance of Dynamic Conditions on Choice

Based on visual observations made during the study, it was apparent that where pedestrians had pre-determined choices of which facility to use, they were also influenced, some more than others, by the dynamic conditions present when they reached the entrance. Prior to proceeding with the second stage of developing a individual-level discrete choice model, it was, therefore, decided to examine the relative weighting of these static versus dynamic factors on a subset of the collected data with facilities of equal dimensions. The analysis was also confined to the ascending direction, both due to feasibility (the collected variables were more easily apparent in the ascending direction) and because of the stronger escalator preference when ascending found in the aggregate model (Section 4.3). Two types of discrete choice models were estimated, a standard binary logit and a discrete mixture of binary logits. The latter type, also known as a latent-class model, was formulated to be able to examine the difference in weighting between static and dynamic factors by considering the process as a combined decision based on information available before and after reaching the facility.

4.4.1 Modelling Framework

This study fit two types of discrete choice logit models in an attempt to predict the choice of stair versus escalator for pedestrians in subway stations. The first model was a standard binary logit; this type of model is appropriate where two choices are available with no correlation in the error terms of the utility functions between the two choices. Such is the case of the choice between adjacent stairs and escalators, where complications resulting from choices that are not co-located (overlapping routes, differences in sources and destinations, etc.) are not of concern. The mathematical formulation for the binary logit is shown below:

\[ P_{str} = \frac{e^{V_{str}}}{e^{V_{str}} + e^{V_{esc}}} \]  

(4.7)

where \( P_{str} \) is the probability of taking the stairs, and \( V_{str} \) and \( V_{esc} \) are the utilities of taking the stairs or escalator, respectively. These utility functions are specified as a linear combination of the independent indicator variables and a constant term for base preference. For the binary-logit model, parameters are
constant throughout all individuals. This results in identical utility functions across all individuals. Since the utility functions are defined relative to each other, one can set $V_{exc}$ to zero, resulting in the following simplified version for the probability of taking the stairs for each pedestrian:

$$P_{str} = \frac{1}{1 + e^{-V_{str}}}$$  \hspace{1cm} \text{(4.8)}$$

The second model examined was the latent-class logit (or mixture of discrete logit) model. Latent-class models attempt to handle the issue of heterogeneity in the population by segmentation of the underlying population into specific classes of agents, each with different utility functions. This is in contrast to mixed-logit models, where taste heterogeneity is captured with a continuous distribution for each parameter, and the binary-logit model, where a homogenous population is assumed. In the latent-class formulation, there is, instead, a discrete number of classes; however, the association of a specific agent with each class is unknown, and the underlying choice model for each class is usually simple (for example, multinomial logit)[86]. Mathematically, the mixture model is specified as follows:

$$P_j = \sum_{i=1}^{n} w_i P_{ji}$$  \hspace{1cm} \text{(4.9)}$$

where $P_j$ is the probability of choosing alternative $j$, $P_{ji}$ is the probability of choosing alternative $j$ for class $i$, and $w_i$ is the probability of being in class $i$. Determining the class of an individual can be handled in different ways. One is to have indicator variables that can be used to formulate a probability of an individual belonging to a specific class. If, however, no such information is available, the class probabilities $w_i$ can be left as parameters for estimation.

![Latent-class model flow diagram.](image)

This latter method was used for this study, where the latent-class logit model structure is used to determine the relative weight of the sensitivity of pedestrians to the dynamic conditions faced when arriving at the stair-escalator facility in contrast to deciding beforehand based on a predisposition to use either the stairs or escalator (Figure 4.4). One can also interpret the resulting weight as the proportion
of pedestrians who decide based on predisposition to the escalator or based on the crowd volume, versus those who are open to either mode, and decide based on the conditions (escalator queuing, opposing flow, etc.) at the time of arrival.

As with other logit-based models, estimation of the latent-class, discrete-mixture can be performed using Maximum Likelihood Estimation. This involves minimizing the log-likelihood function for the latent-class discrete-mixture model, over all observations \( j \) and classes \( i \), defined as:

\[
LL = \sum_j \log \left( \sum_{i=1}^{n} w_i \frac{1}{1 + e^{-V_i}} \right)
\]  

(4.10)

4.4.2 Data Processing and Qualitative Observations

The data used for this stage of the study came from video taken at two of the six locations during the larger data collection effort, Bloor and St. George. Both stations are interchanges, permitting transfer of pedestrians between trains of the two major lines in the city (the u-shaped Yonge-University-Spadina line and east-west Bloor-Danforth line). In addition, the lower platforms where observation was made for both are also centrally located, serving trains in both directions along the adjacent line; therefore, crowds can approach the vertical circulation facilities from either direction. In addition, the locations analyzed within this study had identical configurations: staircase widths of 1.7m, 30 steps each of which were 16cm in height, a positive escalator offset of 2m (staircase entrance receded with respect to the escalator), and escalators moving in the upward direction. The video recordings used were ones made on April 14, 2012 for 15 min segments of time (length set based on time and resource constraints) at one stair-escalator pair at each of the locations, observing flow during Saturday afternoon from 12-4 PM.

Figure 4.5: TTC station observation locations at St. George (left) and Bloor/Yonge (right).
Data Processing

For each location, video was manually processed to extract the following information for each individual as they approached (within a few steps of entrance) and entered the facility:

- the number of other pedestrians queuing at the escalator
- the number of other pedestrians queuing at the stairs
- the number of pedestrians visible on the first 10 steps of the escalator
- the number of pedestrians visible on the first 10 steps of the stairs in the direction of the escalator
- the choice made between taking the stairs or escalator
- the direction of approach (stairs, escalator or head-on)

In addition, the opposing flow values were taken as the number of individuals reaching the bottom of the staircase in the opposite direction in the next 10 seconds (the approximate time for pedestrians to traverse half the staircase). All values were divided by the observed number of lanes (2 for escalator, 2.5 for staircase) to get the density per lane. Lastly, total flow density was calculated as the total number of pedestrians queuing and on the first ten steps (sum of the first four extracted variables) divided by the total number of lanes. The choice of examining only the first 10 steps of either escalator or staircase as the relevant measure of facility occupancy was both due to a limitation in the collection method (the video was only able to capture a certain initial section as shown in Figure 4.5), and based on an assumption that only the more immediate population ahead would influence an individuals decision. A summary of the collected data is presented in Table 4.3.

<table>
<thead>
<tr>
<th>Station</th>
<th>Total # of Observations</th>
<th>% Taking Escalator</th>
<th>% Taking Stairs</th>
<th>% With Opposing Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>St. George</td>
<td>163</td>
<td>84</td>
<td>16</td>
<td>26</td>
</tr>
<tr>
<td>Yonge/Bloor</td>
<td>363</td>
<td>73</td>
<td>27</td>
<td>53</td>
</tr>
</tbody>
</table>

4.4.3 Method of Analysis and Model Estimation

The method of model estimation for both models involved three steps: (1) processing the data with appropriate calculations to determine queue lengths per channel and facility densities (2) a step-wise subtractive method to develop the binary logit and latent-class binary logit models (3) validation of the model by evaluation of prediction accuracy on the provided data-set, goodness-of-fit and the appropriateness of the final coefficients.

Estimation was performed using BIOGEME, a software written to estimate discrete choice models using simulated maximum likelihood estimation, and with the capability to handle latent class models [87]. The software was specified to use the CFSQP algorithm, based on sequential quadratic programming [88]. A summary of the resulting model estimation is provided in Table 4.4.
### Binary Logit Model

The initial model, a standard binary logit, modelled the influence of approaching direction, stair and escalator density, opposing staircase density, escalator queue length, overall flow and intrinsic preference. Following estimation, stair density, opposing flow density, approaching from the stair side, escalator queue length and the alternate specific constant were found to be significant; escalator density and overall flow were not significant. The resulting utility function for stair use was as follows:

\[
V_{str} = -2.78 + 0.613SD - 0.504OD + 0.99SS + 0.108EQ
\]  

(4.11)

where SD is the stair lane density in the upward direction, OD is the stair lane density in the downward direction, EQ is the escalator queue per lane and SS is a dummy variable representing whether the pedestrian approached from the staircase side. The alternate specific constant (ASC) of -2.78 indicates that there is a base aversion to taking the stairs.

### Latent-Class Binary Logit Model

The subsequent model estimated was the latent-class model. As detailed earlier, this involved two separate binary logits with different utility functions, jointly estimated with their weights. The first logit model consisted of variables believed to be available on approach to the facilities, or incorporate predisposition: the approach side, the overall flow preceding the pedestrian and a constant term reflecting predisposition to the use of the staircase. All three parameters were found to be significant, resulting in the following utility function, where SS is the dummy variable representing approach from the staircase side, and F is the total number of people preceding the pedestrian in the previous 10 seconds:

\[
V_{str1} = -12.7 + 4.36SS + 0.809F
\]  

(4.12)

For the local conditions binary logit, variables for escalator and staircase densities, escalator queue and opposing density were included, as well as an alternate specific constant (ASC-STR1). Only staircase density and opposing flow density were found to have significant parameters, resulting in the second
utility function:

\[ V_{str2} = -2.22OD + 5.92SD \]  

(4.13)

Lastly, the weight parameter was found to be significant with a value of 0.268 for the latter logit model. The final latent class model is as follows:

\[ P_{str} = 0.732 \left( \frac{1}{1 + e^{-V_{str1}}} \right) + 0.268 \left( \frac{1}{1 + e^{-V_{str2}}} \right) \]  

(4.14)

4.4.4 Analysis

A standard binary logit model provided a baseline comparison for the latent-class model. The estimated standard binary showed, for the likelihood of taking the stairs, significant positive influences of existing staircase pedestrian density, existing escalator queue, and when approaching from the staircase side, a significant negative effect of opposing staircase density and a base aversion to stair use.

These trends agreed well with observations made during collection. During collection, it was observed that there was a clear preference for the escalator, even in the face of growing queues, with pedestrians waiting to use the escalator, even if the staircase was clear; the staircase was not seen to ever have a queue in the off-peak data collection period. There were, however, times where stair use was fairly high, even on moderate flow, and this generally occurred when several early individuals made the choice to use the stairs, enticing others to do the same in a possible case of mimicry. This tendency of pedestrians to mimic prior choice provides an explanation for the peculiar positive influence of staircase density on the utility of using the staircase, even in the presence of opposing flow, where upward channels were observed forming. Stair choice was visibly higher for crowds alighting from trains and approaching from the staircase side, but quickly reduced when opposing flow in the downward direction occurred; this was counteracted at times if sufficient individuals pushed their way to form upward lanes. Lastly, those with luggage, strollers, elderly and travelling in groups were found to markedly prefer the escalator as expected; however, these personal characteristics were not quantified at this stage, left instead for the full discrete choice analysis described in the next section.

With mimicry providing an explanation for the positive effect of staircase density (beyond any correlation with escalator queue density), all parameter signs and magnitudes are reasonable and expected. However, the variables used represented intentionally a subset of those which should be included in modelling vertical circulation choice. The dimensions of the facility, particularly height, should play a significant role in stair use, but its effects were not examined. With the purpose of this immediate study to examine how pedestrians weigh static versus dynamic factors in making their choice, the static variables were kept constant by limiting observations to those made at locations with identical physical dimensions and similar layouts. The influence of the direction of travel and the level of mobility of the individual should also be investigated, either as explanatory variables or via segmentation, with specific models made for each direction and mobility type.

The latent class model showed similar results with respect to significant explanatory variables on stair choice as the binary logit. The only difference was the inclusion of a positive influence of increasing overall flow in the initial choice model in place of escalator queue density in the local conditions model. The static-vs-dynamic weight parameter showed a dominance of the initial choice model with local dynamic conditions playing a relatively minor role. This result can be explained in two ways based
on the interpretation of the model. Under the first interpretation, the initial choice based on static conditions is primary for a pedestrian, with the chance of using a stair increasing only marginally from a choice based on more dynamic factors. With the second interpretation, this implies that most individuals tend to make their choice solely on their basic preference of stair-vs-escalator, the overall flow entering the facility and their direction of approach. Only a small segment of pedestrians instead make their choice by considering the decisions made by other pedestrians who had gone immediately in advance and the level of opposing flow.

Goodness-of-fit measures of rho-squared and likelihood ratio were very similar for both models at 0.372/0.384 and 281/292 for the standard logit and latent class models, respectively. While the latent class model had a slightly higher rho-squared, both values were acceptable.

![Figure 4.6: Predictive accuracies of estimated latent class and binary logit models](image)

Using the estimated models, Monte Carlo simulations were performed on the initial data set to evaluate the ability of the models to examine how well the models reproduce the observed choices. The two were found to have very close predictive ability at a moderate level, slightly over 70%, with the standard logit model producing marginally better, but not noticeably different results (Figure 4.6). For such a dynamic situation, with significant variability between individuals, the lack of a high predictive ability is not surprising.

The similarity of goodness-of-fit and predictive ability of the two models appears to imply that there is no practical benefit in the use of the latent class model over the standard binary logit for the stair-vs-escalator choice situation, given its added level of complexity. Segmenting the population into two classes did not increase predictive ability, at least where the same revealed preference data was used for both the latent class and standard binary logit models. Whether the availability of panel information (time series data) available at the various points of the decision making process would improve the result is an open question; however, acquiring such information would necessitate a stated preference survey, impractical for such a dynamic situation. Nevertheless, the latent-class model does still serve an informative purpose, with the weight parameter indicating that the choice process is not particularly influenced by choices of other pedestrians.
4.5 Binary and Mixed-Logit Models

The final stage of examining pedestrian vertical circulation choice in subway stations involved the development of a full set of discrete choice models incorporating a range of factors. This examination consisted of the following steps: data collection and processing of the relevant factors, development and estimation of the models, and incorporation of the models into the pedestrian simulator MassMotion to evaluate their effectiveness during their intended application.

4.5.1 Motivation

The motivations behind this final stage of the vertical circulation study were three fold. The first followed from the same impetus behind the original aggregate model, namely to develop a better understanding of how pedestrians are likely to use vertical circulation facilities in stations. The second was to develop a model of vertical circulation facility choice from the perspective of an individual rather than an aggregate model. The final motivation was to understand the applied accuracy of the model when combined with a pedestrian walker model found in a pedestrian simulator; for this study, the 3D pedestrian software MassMotion, developed by Oasys, was used.

4.5.2 Data Extraction

The data used for these final models was sourced from the same data collection effort as the first two stages, but with significant changes in how the information was processed. Unlike the latent-class examination, where information was re-extracted from the source video, the data used was taken from the individual measurements that were aggregated for the aggregate models (Section 4.3). This was due to differences in camera angles and different fields of views at the various locations, which made consistent boundaries for consideration of items like queuing levels and total flows difficult to grab this data through spatial analysis of the video. As a result, the same information taken at virtual cordon lines at the entrance of the facilities were used in this section of the study.

This source information comprised of the following, measured at the moment each individual reached the entrance (beginning of escalator or stair landing):

- Time of arrival at entrance of facility
- Mode of facility taken (stair or escalator)
- Direction of travel
- Location (Vertical element and station)
- Physical dimensions of the facility
  - Width of stair and escalator
  - Height
  - Number of steps for the stairs
  - Landing lengths for the stair
  - Offset between stair and escalator entrance
• Mobility group
  – Normal or restricted (luggage, disability)

• Approach direction

4.5.3 Data Processing

The models of vertical circulation (VC) choice were developed based on the assumption that individuals decision of which facility to take would be dependent on the physical characteristics of the facility, the PRM group of the agent, overall flow, the decisions of other pedestrians (resulting in levels of use and queuing of the facility when the pedestrian makes his/her choice), and how the facility is approached. As only the time of entrance, the mobility group and choice of facility were reliably extractable from the video, information about queuing levels and facility utilization had to be extracted from this entrance-time data.

Under these conditions, the proper time frames to use in what should be considered a queue for an approaching agent, the range of preceding stair and escalator pedestrians influencing choice, and the time the choice was made is not easily discernible. As a result, a different approach was used in this study, by simultaneously determining these ranges while estimating the models (described in the later section). To accomplish this, software was written, developed in C#, to allow for on-the-fly calculation of the variables of interest from the individual entrance-time points, paired with batch estimation of the models via the estimation software BioGEME (the same software used for the vertical circulation discrete choice models). The ranges chosen (different for the two directions) were those that produced models with the highest degree of fit. This method was dependent on the assumption that the proper durations to use when determining queuing and stair and escalator usage would be the ones that maximized the predictability of the developed models. As it was expected that pedestrians might consider these ranges differently depending on the direction, these values were calculated separately for the ascending and descending directions.

Using this approach, the following variables were extracted for each person moving in the direction of the escalator (where a choice existed), with persons with restricted mobility (PRM) separated from those without any restrictions, and estimated separately:

**Direction** A boolean value identifying whether travel was in the ascending (1) or descending (0) direction.

**Stair Use Factor - SF** Number of pedestrians who had recently entered the stairs in advance of the pedestrian at the time of choice in the same direction as the pedestrian, multiplied by the percentage of lanes that were stairs (75cm assumed per individual lane width for stairs as per TCQSM guidelines, and the observed number of lanes for escalators). This multiplication by the percentage of relative stair capacity rather than dividing by the number of stair lanes was due to the follow-the-leader phenomenon found in the first stage (aggregate model).

**Opposing Density - OD** Number of pedestrians who would be opposing the pedestrian as they make their way up the stairs. This was assumed to be the number of pedestrians who would have entered the opposite end when the pedestrian began his/her ascent or descent. As no data was collected at this opposite entrance, a time period of aggregation was used based on the average
walking speed up or down stairs as measured in the field. This aggregated value was divided by the number of lanes in the staircase with also a 75cm with per lane assumption.

**Escalator Use Factor - EF** Number of pedestrians who had recently entered the escalator at the time of choice multiplied by the percentage of lanes that were escalators.

**Queue Factor - QF** Number of pedestrians queuing at the escalator and stairs together at the time the choice was made. Without a proper assessment of whether queuing was in-fact occurring based on the time-series data, a queue in this case was assumed to be any pedestrian ahead of the individual at the time of choice. As with the staircase density, this value was multiplied by the percent of lanes that were stairs.

**Stair Approach - SA** A boolean value (0 or 1) on whether a pedestrian accessed the facility from the stair side, compared to either arriving directly or from the escalator side.

**Height - H** A decimal value representing the total height of the facility in metres.

The variables described above were chosen both because it was hypothesized that they would be the possible factors to affect vertical circulation choice, but also to limit the model inputs to variables that could be realistically used when the model was later implemented within the pedestrian simulator MassMotion.

### 4.5.4 Model Framework

The models developed for this final stage were discrete choice models of the standard binary logit and binary mixed-logit variety. With a lack of sufficient data for PRM individuals, binary logit models were only attempted for that type of individual; both binary logit and mixed logit models were estimated for those without restricted mobility (referred to as PRM Free). The theory and structure of the standard binary logit was discussed in a prior section (Section 4.4.1), with the general form as show in Equation 4.7. The mixed-logit is a generalized and more flexible form of the logit model, eliminating limitations with the base model by allowing for taste variation and time-based correlation of unobserved factors over time \[89\]. This ability to introduce taste variation is particularly useful for pedestrian modelling, permitting parameter coefficients to be distribution rather than fixed values for all individuals. The formulation is stated as shown below for the probability of decision maker \(n\) selecting alternative \(i\) with utility \(V_{ni}(\beta)\) and parameters \(\beta\):

\[
P_{ni} = \int \left( \frac{e^{V_{ni}(\beta)}}{\sum_{j=1}^{J} e^{V_{nj}(\beta)}} \right) f(\beta) d\beta \tag{4.15}
\]

As was true with the prior models of vertical circulation earlier in this chapter, \(i\) has two values representing the choice to take either the escalator or the stairs; in addition, individual models were estimated for the two directions, ascending and descending. The utility functions are also similarly structured, but with slightly different variables, most noticeably either a normal or lognormal distribution for the queue density parameter; the former assumes that it is possible that pedestrians might be attracted to queues, while the latter assumes that they either generally ignore or are repulsed. The two utility functions for the stairs (\(str\)) and escalator (\(esc\)) are in the equations below. One key item to mention is the lack of alternative specific constant; this was chosen to be 0, with the height variable...
Chapter 4. Modelling Vertical Circulation Choice

Determining the base preference for the escalator and allowing for the logical equal probability in a situation of zero values for all parameters.

\[ V_{str} = \beta_{OD} OD + \beta_{SF} SF + \beta_{SA} SA \]  
(4.16)

\[ V_{esc} = \beta_H H + \beta_{EF} EF - \beta_{QF}(\mu, \sigma)QF \]  
(4.17)

While variation in other parameters were also a possibility, they were maintained at constant values for this model for a few reasons. First, these other variables, stair and escalator use, stair-side approach and height, were believed to be less likely to have significant variation between individuals; queuing, on the other hand could have a range of either positive or negative values, depending on if its presence caused a follow-the-leader or repulsive effect. Second, initial estimations showed lack of parameter significance when additional variation terms were introduced; part of this might be attributable to insufficient data for too many distributions in the model. Lastly, with the large range of data aggregation combinations used (detailed in the following section), there was a computational feasibility problem with attempting to estimate models with too many parameters with distributions due to the slow speed of the method of simulated log-likelihood required for mixed logit model estimation.

Lastly, for the standard binary logit models, the two utility functions were identical to those in Equations 4.16 and 4.17, but with a \( \sigma \) value of 0 (i.e. no distribution) for the queue density parameter.

### 4.5.5 Model Estimation

The process of model estimation was simultaneously performed along with data extraction through a developed C# program, with estimation handled by the discrete choice modelling software BioGEME, developed by Professor Michel Bierlaire from EPFL in Switzerland. The software permits estimation of a wide range of discrete choice models, including mixed-logit. As the mixed-logit uses an arbitrary distribution for specified parameters, the software uses a simulation process to minimize the log-likelihood function. Each step in the software involved generation of the variable data based on the combination of time ranges for the round; this was passed to BioGEME for estimation. After all runs, the resulting model estimation output was processed to consolidate results and extract goodness-of-fit data and final parameter values. The goodness-of-fit measure used was \( \rho \), defined as shown in Equation 4.18, where \( LL \) denotes the log-likelihood value of the null model and \( LL_{final} \) denotes the log-likelihood value of the final model at the estimated parameters:

\[ \rho = \frac{LL_{final} - LL_{null}}{LL_{null}} \]  
(4.18)

The final models for use in the next step were those that maximized \( \rho \), with one selected for each type (binary or mixed logit) and direction (ascending and descending), for a total of 6 models (4 for the PRM Free pedestrian group and 2 for the PRM pedestrian group). A step-wise backward elimination process was conducted for non-significant variables (\( \alpha \) of 0.05) to result in the final models.

The corresponding parameter values, with p-values, are summarized in Table 4.5 for the 6 models, 3 in either direction. As shown in the table, the models for those without restricted mobility were found to be sensitive to a wider variety of parameters, nearly the full set estimated save the approach direction for the ascending models. The time aggregation of data parameters and the goodness-of-fit for each of the models are provided in Table 4.6. A queue aggregation of 3 indicates that the queue level used was
the number of individuals ahead of the decision maker when he/she was 3 seconds before entering the facility. Also indicated is whether the queue parameter followed a normal or lognormal distribution. The $\rho$ values show a reasonable degree of individual fit, with better results in the ascending direction, where the variables chosen were able to better predict the choice taken. This was as expected with pedestrians known to distinguish less between escalators and stairs in the descending direction due to the reduced effort of using the stairs down versus when walking up; this was also consistent with the other models described earlier in the chapter.

### Table 4.5: Summary of Final Disaggregate Vertical Circulation Model Parameter Estimation

| Parameter          | Estimate | t value | Pr(>|t|) |
|--------------------|----------|---------|---------|
| **Ascending, PRM Free Model (Mixed Logit)** |          |         |         |
| SF                 | 0.313    | 8.86    | 0.00    |
| OD                 | -0.275   | -9.28   | 0.00    |
| H                  | 0.452    | 25.11   | 0.00    |
| EF                 | 0.584    | 9.52    | 0.00    |
| QF ln $N(\mu, \sigma)$ | -0.663, 0.944 | -7.78, 14.13 | 0.00, 0.00 |
| **Ascending, PRM Free Model (Binary Logit)** |          |         |         |
| SF                 | 0.151    | 8.49    | 0.00    |
| OD                 | -0.170   | -12.79  | 0.00    |
| H                  | 0.392    | 32.78   | 0.00    |
| EF                 | 0.286    | 11.62   | 0.00    |
| QF                 | -0.486   | -17.07  | 0.00    |
| **Ascending, PRM (Binary Logit)** |          |         |         |
| H                  | 0.654    | 11.03   | 0.00    |
| QF                 | -0.776   | -6.59   | 0.00    |
| **Descending, PRM Free Model (Mixed Logit)** |          |         |         |
| SF                 | 0.198    | 5.59    | 0.00    |
| OD                 | -0.279   | -5.42   | 0.00    |
| SA                 | 0.879    | 6.08    | 0.00    |
| H                  | 0.333    | 14.21   | 0.00    |
| EF                 | 0.246    | 3.98    | 0.00    |
| QF ln $N(\mu, \sigma)$ | -1.57, 1.68 | -7.00, 7.35 | 0.00, 0.00 |
| **Descending, PRM Free Model (Binary Logit)** |          |         |         |
| SF                 | 0.141    | 4.91    | 0.00    |
| OD                 | -0.151   | -6.96   | 0.00    |
| SA                 | 0.664    | 5.92    | 0.00    |
| H                  | 0.277    | 17.21   | 0.00    |
| EF                 | 0.099    | 2.95    | 0.00    |
| QF                 | -0.254   | -7.69   | 0.00    |
| **Descending, PRM Model (Binary Logit)** |          |         |         |
| OD                 | -0.275   | -2.07   | 0.04    |
| H                  | 0.351    | -4.25   | 0.00    |
| QF                 | -0.539   | 5.75    | 0.00    |
### Table 4.6: Summary of Final Disaggregate Vertical Circulation Model Settings and Goodness-of-Fits

<table>
<thead>
<tr>
<th>Logit Type</th>
<th>Direction</th>
<th>PRM</th>
<th>Aggregation Settings</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Queue (s) Stairs (s) Escalator (s)</td>
<td></td>
</tr>
<tr>
<td>Mixed</td>
<td>Up</td>
<td>N</td>
<td>2 6 4</td>
<td>0.394</td>
</tr>
<tr>
<td>Mixed</td>
<td>Down</td>
<td>N</td>
<td>2 4 4</td>
<td>0.176</td>
</tr>
<tr>
<td>Binary</td>
<td>Down</td>
<td>Y</td>
<td>2 6 4</td>
<td>0.287</td>
</tr>
<tr>
<td>Binary</td>
<td>Down</td>
<td>N</td>
<td>2 4 4</td>
<td>0.171</td>
</tr>
<tr>
<td>Binary</td>
<td>Up</td>
<td>Y</td>
<td>2 4 8</td>
<td>0.492</td>
</tr>
<tr>
<td>Binary</td>
<td>Up</td>
<td>N</td>
<td>2 6 4</td>
<td>0.385</td>
</tr>
</tbody>
</table>

### 4.5.6 Parameter Analysis

**Stair Use Factor - SF** Existing use of the stairs was found to be a factor in all PRM free models, with positive parameter coefficients. This positive value, while counterintuitive, is consistent with prior research, which showed follow-the-leader behaviour for pedestrians changing levels. Its value relative to the escalator coefficient was diminished for the ascending direction models, which would be expected given the higher level of effort required.

**Opposing Density - OD** The opposing staircase density was found to be significant in almost all of the models, save for the ascending PRM model where pedestrians heavily favoured the escalator so were unaffected by opposing flow.

**Escalator Use Factor - EF** Like with the stair use, existing stair use was present in all PRM free models, with positive coefficients, with the rationale being the same for this positive value. The draw presented by other pedestrians having taken the escalator, however, was significantly diminished compared to the pull of the stairs, partly attributable to the already high base preference to take the escalator.

**Queue Factor - QF** The queue was found to be significant for all models, with an expected negative coefficient, increasing the chance of an individual to take the stairs.

**Stair Approach - SA** As mentioned earlier, the direction of the approach was not found to play a role in the majority of the models, only affecting pedestrians without mobility restrictions in the descending direction. The most likely explanation is that the additional effort required to ascend the facility, resulting in high base escalator preference, counteracted the proximity of the stairs.

**Height - H** The height variable provides the base preference for the escalator, and as expected the parameter value was positive and significant for all models. Also as expected, this parameter was noticeably higher for individuals with restricted mobility, increasing escalator preference, and greater in the ascending compared to the descending direction.

### 4.5.7 Validation

The method of validation of the developed models followed a similar approach to the other VC choice models in this chapter. The preliminary validation step involved analysis of the individual choice predictions of each of the models. The main validation for the purpose of this thesis, however, was in their
Performance within their intended application within the Nexus platform, when incorporated within the pedestrian simulator MassMotion.

**Performance in Individual Choice Prediction**

The initial validation step involved examining the ability of the models to recreate the input data, converting the values to a more tangible measure, and in particular examining the spread of performance of the model across observed decision makers. Generally logit models are validated by setting aside a certain percentage (up to 50%) of the data, called a holdout sample, estimating on the rest and validating against the unused set [90]. The rationale provided for this method is to avoid ‘over-fitting’ of the model, where it is specified to work well with the provided data, normally by producing a reduced likelihood value, and acquire a gauge of how well the model does when applied against an alternative set of data [90]. There are generally two methods used for determining this hold-out sample, either by using a data set dealing with a separate situation, or extracting a random sample from the overall data set. The latter method can be problematic and may be more indicative of the quality of the sampling method used; theoretically, a perfect un-biased sample of the larger data set should reflect the same behaviour. When appropriate, like with small data sets or limited situations observed, validation can also be performed by estimating with all of the data and examining how well it all is predicted; this should involve more detailed examination beyond a single goodness-of-fit measure and ensuring normal distribution of residuals. Issues observed in the residuals, for example a bimodal distribution, could infer that specific classes of populations exist or the need to add additional parameters.

In specific relation to the model developed here, the entire data set was used for both estimation and initial validation on the ability of the estimated models to predict the input data. This was performed for two reasons. The first was due to the prior segmentation of observed individuals into two classes (normal and those with restricted mobility or PRM); with PRM individuals already a small fraction of the overall pedestrian population, further division would have greatly reduced an already small data set. This segmentation also removed what was believed to be a main originator of possible issues in applying a single model across the entire data set. The second reason was the limited situations observed; as a proper hold-out sample would involve isolating at least a quarter of observed locations, the resulting 6 stations for estimation was believed to be too small. Instead, using Monte Carlo simulation, a choice was made for each observed pedestrian based on the models, and the success of prediction was calculated for the entire data set. This was repeated around 700 times for each model to produce the accuracy distributions shown in Figure 4.7.

The figures show that the models had moderate success in predicting individual choices, with significantly better performance in the ascending direction (mid to high 70% correctly predicted vs mid 60% in the descending direction). As expected, choice in the descending direction is more random and less sensitive to conditions at the time the choice is made, consistent with the results of the prior models in this chapter. Next, the mixed-logit models performed marginally better than the binary logit models in modelling the choice of individuals without restricted mobility. This improvement in performance was attributed to a significantly better ability to correctly predict stair usage, with escalator use prediction within 1%. Stair use prediction as a whole was somewhat disappointing, with the binary logit models able to predict 27% correctly in the ascending direction, 36% for the descending; this improved markedly to 35% and 41% in the ascending and descending directions, respectively, for the mixed-logit models. Lastly, with the stronger affinity of PRM individuals to taking the escalator and reduced sensitivity to
environment variables, the PRM models performed better than their PRM-free counterparts. All plots exhibit normally distributed residuals, providing support to the developed models as appropriate for the population of observed pedestrians.

Performance within Pedestrian Simulation

While goodness-of-fit measures of the individual choice performance measure are a good first step in model evaluation, for dynamic situations where interaction between decision makers is continuous, a more practical test of model performance is a necessity. This is particularly true when dealing with a situation where a behavioural model is developed using passively observed data (by necessity, due to the inability of more direct survey methods (e.g. stated preference) to collect accurate choice information [76]). As a result, the way the situation can be modelled is limited, and may not be consistent with the way the model could be applied within the context of a pedestrian simulator. In particularly, the performance of the discrete choice model in correctly predicting crowd flow splits between the vertical circulation elements is unclear without it being linked up with the simulator’s agent walker and route choice model. In addition, as this is a system that evolves over time, when the choice is made, how the variables are calculated and whether this is updated (and at what frequency) would all play a role under the same choice model.

Incorporation into MassMotion

As this study was performed as a collaboration with Arup, the pedestrian simulator MassMotion was the software used to gauge the performance of the developed models. The partnership allowed for access into the code base of the software, therefore granting the ability to modify the base behavioural models for the specific situation being analyzed, a level of access not permitted in the programming interfaces of other commercial pedestrian simulators.

MassMotion is a 3D pedestrian simulation software, where routing of agents is performed automatically and updated dynamically as situations change, with a particular emphasis on congestion. Agents, before beginning their journey, determine their initial path as a series of checkpoints to pass through. Once an agent has arrived on a floor, either through a link from another floor or from a portal as they enter the simulation, they will search for their next waypoint, a link to another floor or an exit. Links are a single connection between floors, which can be adjacent or on different levels, the latter of which would necessitate a structure like a stair, escalator or elevator. They can can be banked to act as a main connection point, rather than being treated separately in the routing process, triggering a separate choice process between the elements of the bank.

To allow incorporation of the models into MassMotion, the software needed modification to its routing choice model. To accomplish this without needing to modify the rest of the agent routing model, the code was inserted in a way to only activate in one particular situation, where an agent had decided to pass through a banked stair-escalator pair. As a result, the only agents who would be considered would be those who have already committed to using the stair or escalator at the specific location, which was consistent with the data that was extracted, also only dealing with those committed individuals. The choice function was programmed to calculate the variables of interest for each agent, and then to apply the generalized mixed-logit model (with all examined parameters included) to choose between the stair and escalator. The ability to have two types of agents with differing models, for the PRM and PRM-free groups, was also incorporated. Lastly, The API developed for incorporation of MassMotion into the simulation system was modified to allow for the vertical circulation model parameters to be specified
Figure 4.7: Predictive accuracy of individual stair or escalator choice for the estimated vertical circulation discrete choice models.
externally, to allow for batch simulation during validation testing.

For the logit model variables, there was one main issue, namely how to translate the time-based aggregation used for the queueing, and stair and escalator occupancy to a simulation environment where these values could be calculated only spatially (e.g. the queue as the number of agents within a certain distance). This was handled by setting two values for each variable, both a time (from Table 4.6) and distance value. The distance value was calculated by multiplying each time value by the expected average walking speed. The number of agents used for the model variables was then set as the minimum between the number of agents within the corresponding distance parameter, or the time parameter multiplied by the total number of lanes based on the assumption of a 1 second minimum between pedestrian entry for each lane as seen on the recorded video from the field survey.

In addition to the logit model parameters, parameters were also input that specified how the model would be applied. These all dealt with the timing and frequency of the choice made by agents using the model. The first was the distance at which the choice model would first be applied (either 3, 5 or 7m from the facility entrance); this was paired with a minimum distance to the facility where the choice could not be changed (0.5 or 1m). The number of times the choice could be updated (1, 3, 7) and the time between choices (2, 5) were the last two parameters. All possible combinations of these parameter values were used during validation. The values selected for each parameter were determined based on the results of a preliminary simulations to investigate the sensitivity of the parameters to the stair-escalator splits.

**Process of Validation** To perform validation of the developed models within MassMotion, three steps were followed:

1. 3D models of the observed VC locations were built in MassMotion.
2. Software was written to automate the process of randomly introducing agents into the simulation under the various models, parameter sets and locations.
3. The simulation output data was processed to compare their results against the collected field data to determine model performance.

The 3D MassMotion models of the observed locations were built by using the dimensions of the facilities (Table 4.1) as a guide to have accurate facility widths, heights and offsets. In addition, the placement and general layout of the platform was mimicked in order to have agents approach from realistic directions.

The software to automate validation was written in C# and performed both the task of running simulations under the various parameters and analyzing model performance. First, it was determined that validation would be based on the performance of the models in predicting 10-second splits of stair-escalator use. This period of aggregation was chosen to be long enough to allow for a sizeable number of agents to enter the facility, but still short enough to not significantly average out time-variant flows. In addition, for practical purposes, it allowed the use of the 10-sec aggregate data processed for the aggregate models. As a result, the simulation needed to reproduce the various combinations of input variables for the 10-sec flow, namely the total incoming flow, the total opposing flow, and the % of agents who were in the PRM groups. As agents could only be inserted into the simulation individually, these could not be easily set. Instead, the software randomly introduced a varying number of agents, of both
those with and without restricted mobility, as a pulse, repeating the process several hundred times for each direction. Models were tested in pairs, with one set for regular agents, and the second for agents with restricted mobility. The same procedure was also performed without the models enabled, using the existing choice models within MassMotion.

The processing of the results involved aggregation of the individual choices into 10-sec crowd flows, followed by comparison of these aggregated results against the corresponding data collected in the field. The performance of each model pair in predicting the proper stair-escalator split was evaluated for each combination of total flow, total opposing flow and % PRM for each facility. Each situation was examined individually by using a paired t-test, comparing the model and real-world results where at least 3 data points were available for both. A success rate for each model pair with model application settings was determined as the percentage of situations where a statistical difference ($\alpha$ of 0.05) was not found between the model and real-world splits.

**Validation Results** The ability of the model pairs to predict the 10-sec flow situation split was generally good at over 80% (see Table 4.7). There was also a significant range of predictive success based on the application settings. Table 4.7 summarizes the validation results, with the accuracy of the model pairs for the worst and best settings combinations. Also provided are the corresponding accuracy of the existing MassMotion models for comparison. Two key findings were arrived at based on the results. First, the way the choice models were applied had significant impact on their ability to properly predict vertical circulation crowd flow splits, with a difference of around 10% between the lowest and highest accuracy. Second, while poor individual predictability was found in the prior validation step for the descending direction, the aggregate flows balanced out to provide good predictive ability of the models.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Regular</th>
<th>PRM</th>
<th>Min Accuracy(%)</th>
<th>Max Accuracy (%)</th>
<th>MassMotion (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>Binary</td>
<td>Binary</td>
<td>77</td>
<td>87</td>
<td>84</td>
</tr>
<tr>
<td>Up</td>
<td>Mixed</td>
<td>Binary</td>
<td>79</td>
<td>89</td>
<td>84</td>
</tr>
<tr>
<td>Down</td>
<td>Binary</td>
<td>Binary</td>
<td>78</td>
<td>87</td>
<td>69</td>
</tr>
<tr>
<td>Down</td>
<td>Mixed</td>
<td>Binary</td>
<td>82</td>
<td>91</td>
<td>69</td>
</tr>
</tbody>
</table>

Overall, this study of developing a disaggregate vertical circulation choice model showed the importance of incorporating dynamic variables and mobility group in place of traditional demographic characteristics, as well as the need to carefully consider how such models are transferred to and implemented within pedestrian simulation to provide the best results.

### 4.6 Application

Application of these vertical circulation models, in particular the aggregate logit and final discrete choice models, require some knowledge of the population under study. Because, in both cases, separate model types were estimated for passengers with and without restricted mobility (PRM), having an understanding of the makeup of the mobility group composition of the target predictor population is needed. One example of how this can be done was provided as part of the validation portion of the aggregate model study. Namely, this consisted of using field data collection to categorize the population
at the target environment, and then assuming those same percentages when applying the combined model. This can be viable where the situation being predicted is expected to occur within a relatively close time-period to when the initial data was collected, so as to assume that the population composition would remain fairly constant.

For more expansive evaluation where passenger flow through the entire network is being modelled, as is the case for the expected use of the Nexus platform, a separate study would be needed focused on assigning agents a specific mobility group, with information passed along with the agent as it travels through each station simulator. For more permanent mobility groups (disabilities), this information should be more readily available. Classification into other transient groups like luggage carriers could either be performed by a separate model based on destination and demographic factors, or based on a straight percentage. For the purpose of the case study presented later in the thesis, all agents were assumed to have no restriction (fall under the 'normal' PRM group).

4.7 Conclusions

The research presented in this chapter had a few objectives. The first was to further the body of pedestrian movement first principles knowledge by focussing on how pedestrians transition between levels. The second was to develop models that could be incorporated within the pedestrian simulator MassMotion to improve decision-making by agents at a key station bottleneck, and allow for the examination of the impact of this improvement at a network-level, a key intended use of the developed network platform.

Based on an extensive data collection effort at co-located stair-escalator transport facilities in six subway stations in the Toronto subway network, two major types of models were developed. The first were aggregate models meant to be used by transit practitioners, while taking into account some previously unaccounted factors. The second were a set of discrete choice models to be incorporated within MassMotion as an improvement over the existing method.

To this end, a set of logistic regression models were developed to explain the split of escalator versus stair use by transit users. Separate models were constructed for the ascending and descending directions, which were further divided by the presence of PRM individuals. In all models, increased incoming channel utilization increased the use of the staircase; opposing flow on the staircase and the flow approaching from the escalator side were found to increase escalator use. A strong preference for the escalator was found in the ascending direction, with a lack of sensitivity to facility height, while in the descending direction, this preference was strongly dependent on height. Lastly, for models involving the presence of restricted mobility individuals, an increasing composition of these individuals strongly increased the affinity of the group to use the escalator.

In between this aggregate study and the final discrete choice models, an intermediate examination was performed, focused on capturing the highly dynamic nature of the decision to take the stairs or escalators. To accomplish this, the process was modelled by basing choice in part on the dynamic conditions faced by individuals. Two types of discrete choice models were estimated, a standard binary logit and a discrete mixture of binary logits (latent-class), the latter treating the process as a combined decision based on information available before and after reaching the facility. Of key interest was how pedestrians weigh static and dynamic factors when making their choice.

The goodness-of-fits of the models were found to be adequate, with a moderate ability to predict actual splits found, but only marginal difference between the two model types. The ability, however,
of the latent class model to identify the relative importance placed on local, dynamic conditions by pedestrians does hold some value by providing a measure of the degree of stubbornness of pedestrians in sticking with their initial choice. In this study, it was found that the permanent characteristics of the facilities largely dominated the choice of which facility was chosen by individuals. Due to the marginal difference between the two types, however, the complicated latent-class formulation was eschewed for a mixed-logit structure for the final discrete choice models.

This final examination, using the full data set collected and testing its effectiveness after incorporation within the MassMotion pedestrian simulation software, constituted the final stage to expand understanding of pedestrian choice at co-located stair and escalator facilities in transit stations. A set of discrete choice models were developed that showed a promising ability (nearing 90%) to predict aggregate flow splits within pedestrian simulation when time was taken to tune parameters of model application. Nevertheless, the environment studied, while common, was quite spatially localized, albeit at a key station bottleneck. As a result, there remains a research path open towards incorporating these results within a broader understanding of station routing behaviour.
Chapter 5

Modelling Pedestrian Distribution Along Platforms

A proper network model of the subway network requires that the interface between components be correctly modelled. One key interface in the subway network is at the platform, where crowds transition between the subway and the station. Unfortunately, research has been limited on this topic, with a focus on measuring overall dwell times for specific boarding, alighting and through volumes, and basic examinations of the factors affecting the distribution of pedestrians along the platform. A thorough analysis or modelling effort considering all factors, however, has not occurred. As a result, how the layout of a station platform, the flows of pedestrians and the inter-arrival spacing of trains interact to result in observed pedestrian distribution along platforms has not been discerned. This is a required input into boarding and alighting models used to determine train dwell times. In addition, a more accurate model would enable better platform design, both in isolation and in gauging further reaching consequences, and assist in localized crowd management. The main goal of this chapter was, therefore, to develop a model of how pedestrians distribute themselves on platforms under various platform configurations and pedestrian volumes (including arrival patterns). This was performed with the intention of subsequently incorporating the resulting framework and parameters within Nexus.

5.1 Current Literature

Studies that deal specifically with the modelling pedestrian distribution along platforms are few, with the majority of the focus on pedestrian routing and kinematic models (see Section 2.4). The need for collecting data along a platform for platform specific model validation, and the general need for a greater understanding of platform behaviour, however, has been noted in several studies. These have included both studies showing a non-trivial distribution that arises on platforms due to several factors, but also the importance of distribution along platforms on dwell operations, train car occupancy, and overall network performance.

The importance of analyzing platform distributions is well known in the industry with respect to both station and line performance. This has been highlighted in a few studies’ data collection efforts that examined pedestrian flow along platforms as a distinct phenomena from regular flow through passages [91, 92]. When calculating dwell times, researchers normally incorporate a factor relating the peak door
volume to average, resulting in uneven distributions increasing dwell periods [57]. The current Transit Capacity and Quality of Service guide details significant passenger load imbalances in the Vancouver SkyTrain, Toronto Yonge Line Subway, and Seoul Metro line 7 [85]. As a result, a significant underutilization of system capacity results, causing a reduction in overall system performance. This has led to studies that attempt to examine ways to optimize the distribution of passengers throughout a train by playing with factors like where trains stop when arriving at the platform [93].

When originally dealing with platforms, researchers tended to assume uniform distributions for boarding pedestrians [94]. However, several studies have been performed over the years which challenged this assumption. One of the earliest examinations was by Spzlett and Wirasinghe who surveyed platform distributions at two LRT stations in Calgary, Alberta. They found that both the boarding and alighting distributions were not uniform [95]. This lack of uniformity was confirmed by data collection efforts in several studies since in both Europe and China, on LRT and subway platforms [59, 96, 21, 91].

In trying to determine the source of this uneven distribution, much of the focus has been on the characteristics of the origin station. The main factor put forth are the entrance locations, with pedestrians more likely to congregate around them after entering the platform area [95, 59, 96, 21]. In moving beyond the entrance locations, two Chinese studies (Cao et al in 2009 and Wu et al in 2010) attempted to understand how the visibility of the train doors, particularly the angle they form with respect to the entrance position and pedestrian flow, resulted in locations that were more attractive to entering boarding pedestrians (cited in [94]). None of these studies, however, provided models that could be practically be applied in other locations, either only providing the general result of entrance location being important, or providing a complex model that was too specific to the observed location [94].

To deal with this lack of usable model, Wu et al attempted to develop one based on the idea of electric potential energies. The authors postulated that the number of people at waiting areas was inversely dependent on the distance away from entrances and directly dependent on the number of people at each entrance, although not necessarily linearly [94]. This relationship was observed to be similar to the potential energy generated by charged particles. This potential energy equation was adapted for use by defining the potential energy associated with a waiting area as the linear combination of the number of people at each entrance divided by the distance between the waiting area and the entrance raised to a parameter to be estimated [94]. The number waiting was then calculated as the ratio of the potential at the specific waiting area divided by all potentials multiplied by the total number who were boarding in that period [94]. Data was collected on a platform in a transfer station in Beijing with 5 entrances distributed somewhat evenly. The area was video recorded and the number boarding at each door was extracted manually; trains consisted of 6 cars, each with four doors. Based on this data, values for the distance parameter were found for each total volume of people, and an equation was fit by minimizing the sum of squares of the difference between the model values and collected data [94]. There was, however, a significant gap of missing observed volumes between the low volume and high volume scenarios, resulting a large uncertainty on the reliability of the equation produced. Differences between the model and actual counts were greatest for low volumes, improving as volumes increased and pedestrians were more evenly spread throughout [94]. While this study was the only one to date to develop a usable model of distribution, its reliance on data from a single platform with relatively uniformly-spaced entrances brings into question how well it would perform on more varied platform entrance configurations.

To gain a better understanding beyond just factors related to the current station, a wide ranging...
survey was recently conducted by Kim et al. on line 7 of the Seoul Metro. Respondents were selected at random (sample size of 340) and surveyed over 4 weeks. Passengers were asked whether they were targeting specific train cars, and if so, asked questions about their motivation. The key finding was that a strong majority (76.6%) of individuals targeted a specific train car. Of these, 69.7% were motivated by the wish to minimize the walking distance to the exit at their destination station, 16.6% minimizing walking from where they entered the platform at their boarding station, and the remainder (13.5%) looking to pursue comfort (for long trips or to avoid high levels of crowding).

As seen in this section, studies to date have not developed comprehensive models that incorporate all factors that might play a role in determining pedestrian distribution along a platform. Only a single model was found in prior studies that looked to predict distributions, with most examining the influencing factors. In addition, the fact that spreading pedestrians is a time-dependent process, with the distribution at time of boarding dependent on the time of pedestrian flows into the platform relative to the train arrival, has not been considered. Lastly, while it is known that the destination station is a key factor for commuting populations, only the layout of the boarding station platform has been taken into account in existing models. As a result, there remained a need for a more comprehensive model in order to properly model platform distribution.

5.2 Model Structure

As the distribution of pedestrians upon boarding is inherently time dependent, based on when pedestrians entered the platform relative to the train arrival, the realized distribution would only in certain situations be at steady state. Such a process can be handled in two general ways, either an aggregate model where the final distribution is modelled as a function of the possible factors, including the time of flows and positions of entrances, or a disaggregate model where the behaviour of each agent is modelled and the system is allowed to evolve over time. This latter method is what was used in this study, owing both to the amount of data required and the many variable interactions that would need to be collected for an aggregate model, but also to permit easy incorporation into the prototype network simulation system where agents in stations are simulated individually.

Disaggregate pedestrian mobility models generally have two key components that determine how pedestrians make their way through a space, one for overall direction and a second for local movements (or a kinematic model). With respect to platforms, the local movement model handles how people will spread themselves out to avoid being uncomfortably close to other pedestrians once they enter. The second component involves targeting specific portions of the platform, even if the density is higher, in order to better position themselves for their intended exit once they alight. The overall model schematic is shown in Figure 5.1.

5.2.1 Local Movement (Kinematic) Model

The kinematic model acts to naturally space pedestrians to avoid excess crowding as they enter the platform. As detailed in the literature review Section 2.4, while many pedestrian movement models exist, their general focus is on flow of pedestrians on paths, with minimal work done to explain movement on platforms. Within the developed simulation system, where MassMotion is used for the station component, the software has its own waiting model, where agents move as far as needed until their minimum density levels are reached; repositioning can occur if more agents are introduced. This behaviour
was kept intact. However, for this specific study and for the constructed simulator for stations where MassMotion was not used, a different model was needed.

**Molecular Diffusion Theory**

This model was developed by borrowing concepts from fluid dynamics, namely the process of the molecular diffusion of particles in a solution. Looking to fluid dynamics is not a new concept for pedestrian mobility modelling, as shown in Section 2.4, but its application has been solely in explaining the motion of crowds through spaces when density is high. As detailed earlier, in that analogy, people act similar to fluid molecules travelling down a passage, allowing for fluid dynamic equations that explain fluid motion through tubes to be applied to crowd flow. On the other hand, the model developed in this thesis for pedestrian dispersion on the platform treated pedestrians as solute molecules diffusing out across the platform, which is treated as the solvent.

In molecular diffusion, individual solute molecules move through a solvent, spreading out gradually until the concentration of solute molecules is uniform. At a molecular level, the process of diffusion occurs in a chaotic fashion. Molecules move randomly in a straight line until colliding with other molecules (both solute and solvent), with their velocities then reversing. As this repeatedly occurs and individual molecules move in a zigzag fashion, the overall group of molecules slowly diffuse out, with the diffusion speed being much slower than the individual molecular speeds based on the frequency and speed of collisions. The diffusion rate is based on two key factors: the thermal energy of the molecules and the solute concentration gradient. The thermal energy determines the kinetic energy of the molecules, and therefore their speed. The concentration gradient, on the other hand, provides the driving force for the overall direction of diffusion with the solute molecules diffusing from high to low concentration (with fewer collisions occurring in the direction of decreasing concentration). [98]

The rate of diffusion is described in terms of the molar or molecular flux \( J_A \), which is the number of moles or molecules moving through a cross sectional area in each time period. The flux of a solute A through a solvent B can be determined using Fick’s law by multiplying the concentration \( c_A \) gradient along direction \( z \) by the diffusivity coefficient \( D_{AB} \), which is related to the thermal energy and the characteristics of the solute and solvent (size and viscosity). Mathematically, it is expressed as the following for a 1-dimensional gradient:

\[
J_A = -D_{AB} \left( \frac{\partial c_A}{\partial z} \right) \tag{5.1}
\]

The diffusivity coefficient is normally measured empirically for each solute-solvent pair and varies for
when the solvent is a gas versus a liquid. For molecular diffusion in liquids, which acts as the inspiration
for the developed model, diffusivity can also be calculated using the Stokes-Einstein equation:

\[ D = \frac{kT}{3\pi \mu d} \] (5.2)

where \( k \) is the Boltzmann constant, \( T \) is the temperature, \( \mu \) is the liquid viscosity and \( d \) is the
diameter of the diffusing molecule. The temperature of the fluid is also a measure of the kinetic energy
of the solution particles; kinetic energy in turn is, therefore, directly proportional to the square of
particle’s speed. The viscosity is a measure of the 'thickness' of the fluid, or the internal resistance
of its molecules to movement; the greater the viscosity the more difficult diffusion becomes.

This flux is generally expressed in mass transport as the flow rate moving through a cross section
area. As a result, in a small enough time period where concentration is relatively constant, the flux can
also be expressed as:

\[ J_A = c_A V_A \] (5.3)

where \( V_A \) is the average velocity of solute particles in the direction of flow.

**Applying Diffusion Theory to Pedestrian Dispersion on a Platform**

The way that pedestrians spread across a platform where there is no preference on platform location
shares many similarities with the process of molecular diffusion. As with solute particles:

- Pedestrians generally spread out at a speed less than their average free flow walking speed when
  they have no preferred location on the platform to stand
- This speed will be reduced as the pedestrian density increases and will increase away from regions
  of higher density towards more unoccupied regions as pedestrians prefer to be spaced out
- Given enough time, pedestrians have been found to space themselves fairly regularly [20], the same
  phenomenon that occurs in diffusion

Nevertheless, as with the treatment of pedestrians as liquid molecules when fluid dynamics equations
are used to model crowd flow, there are differences and approximations made when considering using
diffusion principles to model the spreading out of pedestrians on platforms.

- While solute particles diffuse out slowly at their average velocity through continuous collisions at
  a consistently higher speed, for pedestrians the added social forces that they experience prevent
  these collisions, instead resulting directly in slower overall movement
- Solute concentrations and flux are normally expressed in mols/m-sec, which are orders of magnitude
greater than realistic counts of pedestrians on platforms
- While pedestrians do space themselves fairly regularly (except in the case of groups), unlike with
  solutes, this generally does not result in equal spacing for low volumes of people; instead they will
  move as far as needed to maintain a comfortable distance
- While equilibrium concentrations are eventually reached with solute particles, they remain con-
nstantly in motion and will continue changing places; pedestrians will eventually settle at a final
location after spreading out for some time with newer entrants to the platform often walking by
them to reach lower density locations

- The process of diffusion, calculated at each time step and location using Ficks Law (Equation
5.1) considers molecules in aggregate and solving for the concentration at each location; when
simulating pedestrians in an agent framework, the process expressed in the law has to be adapted
so as to operate on individual agents while achieving the same overall result

As a result of these differences, the developed model adapts, rather than blindly copies, diffusive
principles, and aims to arrive at a proper overall distribution of pedestrians along the platform, but
realistically will not accurately position individual pedestrians.

A Diffusion-based Model of Pedestrian Dispersion on Platforms

In the diffusion-based kinematic model, pedestrians are assumed to be autonomous individual agents
that naturally spread out away from regions of high density to more open spaces. By combining the two
equations (5.1 and 5.3) that define flux, the average velocity of particles in a short time period can be
expressed as shown in Equation 5.4, with a positive $z$ for agents moving right.

$$V_A = -\frac{D_{AB}}{c_A} c_A \left( \frac{\partial c_A}{\partial z} \right)$$  \hspace{1cm} (5.4)

This equation provided the means with which to update the velocity of individual agents based
on the surrounding local density of neighbouring pedestrians, the pedestrian density gradient and the
diffusivity coefficient. The diffusivity coefficient, as explained in Equation 5.2, for molecular diffusion is
proportional to the temperature (and square of particle speed) and inversely proportional to the viscosity
and diameter of the diffusing particle. Applying this to the pedestrian-platform situation, the diffusion
coefficient can be thought of as to be directly proportional to the square of the maximum speed $v_m$ of
pedestrians; the viscosity of the medium would be the friction experienced between walking pedestrians
and the platform floor, and so is constant. The change in concentration in direction $z$ is the difference in
the number of agents between the regions to the left and right of the pedestrian divided by the area used
during calculation. The equation for pedestrian speed $u_{pd}$ on a platform, with the positive direction
defined to the right, under the diffusion model (notated as $d$) can, therefore, be stated as:

$$u_{pd} = -\frac{K_d v_m^2}{\rho} \left( \frac{\Delta N_l}{Wl} \right)$$  \hspace{1cm} (5.5)

where, $\rho$ is the density of pedestrians in the direction of diffusion, $N_l$ is difference in the number
of pedestrians between the regions to the right and left of the pedestrian, aggregated to a distance of
$l$ metres (with $l$ to be estimated), $W$ is the width of the platform, and $K_d$ as the pedestrian diffusion
coefficient to be estimated.

The use of this equation in updating pedestrian speed ignores any consideration of acceleration,
instead assuming that it is rapid enough to not be of concern, allowing a change in speed to be relatively
instantaneously. This is an assumption that is often made in discrete pedestrian models, such as cellular
automata; given the low maximum speed of walkers and the approximate nature of the presented model,
this was believed to be reasonable.
5.2.2 Active Targeting Model

The second part of the platform model is a component that handles the behaviour of pedestrians targeting a location along the platform. This makes an assumption of familiarity resulting in habit formation with respect to transit users positioning themselves at their destination exits to minimize their overall travel time. As a result, it was hypothesized that this would be most likely typical of commuters who take the same route. On the other hand, those using the subway during off-peak hours would most likely be visiting unfamiliar stations, and therefore, would be less likely to know where they should board the train. However, even for commuters, it was expected that only a portion would actively position themselves for any number of reasons, or that there would be a sub population of users that were not particularly familiar with the destination.

It was believed that the probability that an individual would target a section of the platform would not have any significant relationship to their personal characteristics; in any case, this information could not be feasibly recorded given the process used, as described in the following section. As a result, a single probability parameter was used for all agents for each platform, a parameter that would be estimated along with the other model parameters.

For those pedestrians who are targeting, the next step requires selection of a location along the platform that is the intended destination. In situations where the destination station is known, and especially if the target exit is known, this is a straightforward exercise to perform for each traveller. Such situations would include the case where the model is being applied within a network simulation system where agent paths are known exactly, or in the improbable case where the survey data has that level of detail. When field data on how pedestrians spread along a platform is collected, it is generally infeasible to simultaneously record the target destination station and associated exit of all individuals. In this case, an alternate source is needed to provide this information. One such source was used for this study, and is detailed later in the chapter. The end result is a targeting distribution for each origin station; this distribution provides the probability of a pedestrian targeting a specific section of the platform to board, denoted in the model as $T_{is}$ where $i$ is the section and $s$ is the origin station.

Once the destination is known, the targeting model can now be specified, describing how pedestrians make their way towards their preferred location on the platform. To allow for compatibility with the diffusion model described in the prior section, the result of the active targeting model was set to also be the speed of the pedestrian, in the absence of diffusion effects. Developing an equation to explain this movements required consideration of two key factors:

- The walking speed of pedestrians as a function of surrounding density
- The degree of motivation of pedestrians to move to their desired location

Determining the speed of pedestrians as a function of surrounding density is not a trivial process. A number of studies have been performed over the years that have examined this relationship, but they have focussed on situations of crowd flow. The results from these studies were joined together by the following equation developed by Miguel et al [99]:

$$u(\rho) = \begin{cases} 
  u_0 & 0 \leq \rho \leq \rho_{fs} \\
  u_0 - \frac{1}{\tau} \left( \sqrt{\rho_{fs}} - \sqrt{\rho} \right) & \rho_{fs} < \rho < \rho_{rp} \\
  u_{rp} + \frac{2}{\tau} ln \left( \frac{\rho_{pk}}{\rho} \right) & \rho_{rp} \leq \rho \leq \rho_{max}
\end{cases} \quad (5.6)$$
where $u$ is the pedestrian speed, $u_0$ is the desired speed, $\rho$ is the pedestrian density, $\rho_{fs}$ is the density at free speed, $\rho_{rp}$ is the density where other pedestrians start becoming obstructions and "repulsive" forces are felt, $u_{rp}$ is the corresponding speed when these forces are first felt, $\rho_{max}$ is the max density, $\tau$ is the relaxation time and $\gamma$ is an estimated coefficient. In between $\rho_{fs}$ and $\rho_{rp}$, pedestrians can still walk freely without bumping into each other, but occasionally reduce speed to maneuver. Miguel et al in determining an appropriate set of densities to use, came up with the following based on an average of three studies (by Ando et al, Togawa and the Green Guide): $u_0$ of 1.34 m/s, $\tau$ of 0.733s, $\gamma$ of 1.003 m/s, $u_{rp}$ of 0.78 m/s, $\rho_{fs}$ of 0.8 persons/m$^2$, $\rho_{rp}$ of 2.0 persons/m$^2$ and $\rho_{max}$ of 5.5 persons/m$^2$. [99]

As mentioned, while representing a comprehensive study of the speed-density relationship in moving crowds, this equation ignores the influence of standing pedestrians as seen on platforms. Only one study was found that examined their influence. It was performed using a cellular automata model, but provided no final equation explaining the relationship between speed, moving density and stationary density; however, a general 20% increase in travel time was found[92]. Under the assumption that stationary pedestrians waiting on the platform would result in greater friction compared to when pedestrians are flowing in the same direction at moderate to high densities, the following modifications were made:

- The 20% observed reduction in speed was applied through a reduction in $u_{rp}$
- $\gamma$ was scaled to reduce the pedestrian speed to 0 as maximum density was reached (under the assumption that the vast majority of pedestrians would be standing still on the platform rather than trying to move as with crowd flow at very high densities)
- To maintain continuity in the last two density regions ($\rho$ of 2.0), $\tau$ was multiplied by a constant
- The density boundaries were maintained with no basis to modify them; however, it is likely that these would be modified once waiting pedestrians are introduced

As a result, the final set of speed equations under for a fully motivated pedestrian were as follows:

$$u_m(\rho) = \begin{cases} 
1.34 \text{m/s} & 0 \leq \rho \leq 0.8 \\
1.34 \text{m/s} - \frac{1.277}{0.733} \left( \frac{\sqrt{\rho} - \sqrt{0.8 \text{pers/m}^2}}{\sqrt{(0.8 \text{pers/m}^2)\rho}} \right) & 0.8 \leq \rho < 2.0 \\
0.624 \text{m/s} + \frac{1.234}{2} \ln \left( \frac{2.0 \text{pers/m}^2}{\rho} \right) & 2.0 \leq \rho \leq 5.5
\end{cases}$$

(5.7)

This set of equations provides an approximation for the speed of pedestrians moving through a platform towards a destination, while avoiding a mix of standing and moving pedestrians. The motivation itself to move, however, depends on the relative position of the pedestrian to his/her goal. As it was believed that pedestrians would naturally have a higher motivation to reach their goal the further away they were, with some tolerance of precise standing location, a function $m(d)$ was selected that had the characteristic of having the value of 1 when distance $d$ was large, but rapidly dropping off to 0 as $d$ approached 0. This would mimic strong motivation when far away from the desired point, but a region of tolerable destination, without a requirement of being exactly at the desired point. The functional form selected was $\frac{\omega}{\sqrt{1+(\gamma x)^2}}$ which maintains a value of 1 until a threshold dependent of the value of $\gamma$ which is to be estimated, with an example shown in Figure 5.2.

With $x_t$ denoting the target position and $x_P$ the current pedestrian position, the motivation function for the model of a pedestrian located at $x_P$ is defined as:
Putting these two pieces of the active targeting model together from equations 5.7 and 5.8 results in the governing equation for the targeting component of the walking speed of a pedestrian $u_{pt}$ at density $\rho$ and location $x_p$ as follows:

$$u_{pt}(\rho, x_p) = \frac{\gamma |x_p - x_t|}{\sqrt{1 + (\gamma(x_p - x_t))^2}} u_m(\rho)$$  \hspace{1cm} (5.9)

## 5.2.3 The Overall Platform Model

The final platform model combines the diffusion and targeting models. The targeting model is specific to those pedestrians deemed to target, so will apply to only a portion of all pedestrians. Even targeting passengers, however would be affected by diffusive forces; while the targeting speed component would be expected to dominate far away from their preferred platform location, as they near, diffusive forces caused by the positioning of other passengers may play a larger role. As a result, to form the final equation of pedestrian speed under the proposed model (Equation 5.10), the two equations governing each component (5.5 and 5.9) are combined, with the overall speed capped at the maximum for the individual.

$$u_P = \min(\max(u_{pt}, u_{pd} + \beta u_{pt}))$$  \hspace{1cm} (5.10)

where $\beta$ is either 1 or 0, depending on if the agent is or is not targeting a platform section, respectively.

### Handling Alighting Agents

While Equation 5.10 deals with the speed of pedestrians who have entered the platform to board a train, it does not handle alighting pedestrians. While they may have minor influence on the boarding distribution, their incorporation is required for a complete platform model and to account for them if
present in the data during model estimation. These pedestrians are treated more simply in the model, with movement determined in a similar fashion to the targeting model, but with motivation to walk maintained until the agent has reached the exit, and diffusion ignored. In addition, with alighting pedestrians essentially flowing together towards exits, their movement should be less restricted than pedestrians having to navigate around waiting individuals. As a result, the speed was set to be updated in the model using Equation 5.6.

5.3 Data Collection and Processing

Two sets of data were required in order to develop the proposed model. The first was data collected in the field to capture how pedestrians distributed themselves along platforms. To take into account that commuting populations will often learn their routes and target platform sections, the second set was knowledge of where pedestrians were headed on their journey.

5.3.1 Pedestrian Distribution Field Data

The goal of field data collection was to record how pedestrians distributed themselves along platforms under various conditions (physical configurations, volumes, train arrival patterns) to both help in developing and act as a benchmark for the platform distribution model. In addition, it was believed that differences would occur in pedestrian behaviour based on the level of familiarity with the station and the network, and as such examining periods with high and low levels of commuters was also a goal. There were, however, limitations due to the locations that collection was conducted that greatly influenced the method used and the processing and modelling that could be performed.

Locations

The sites selected for data collection were several subway platforms scattered along the east-west Bloor-Danforth line of the TTC subway system. A total of 8 stations were chosen to provide a variety of entrance/exit locations, while maintaining a consistent length (around 140m) across all. The list of all chosen stations and the location of the entrance/exits for each are shown in Table 5.1. To avoid ambiguity of pedestrians able to board multiple trains, all platforms were only associated with a single train (no centre platforms serving multiple lines or directions). Only the platform in the direction of the downtown core was observed, divided equally between eastbound and westbound directions, to isolate a population that was most likely to be commuters. Additionally, observations were made both during the peak AM period and off-peak period to record the behaviour of non-commuters who would be less likely to have familiarity of the layout of their destination, and therefore less likely to position themselves near a specific section of the platform.

Collection occurred across 6 days spread between two weeks in November of 2012. Each week dealt with a separate set of 4 stations; the first involved observation of the east-end stations (Victoria Park, Main, Broadview and Sherbourne), while in week two the west-end stations (Spadina, Bathurst, Runnymede and Dundas West) were observed. Data was collected during the morning commute AM-Peak period (between 7:45-9:15AM) for two of the three days each week (2 of the 4 stations per day), and during off-peak periods (between 12-4PM) on Saturday.
Table 5.1: Platform Data Collection Locations

<table>
<thead>
<tr>
<th>Station</th>
<th>Entrance Locations (m)</th>
<th>Width (m)</th>
<th># Trains (Peak/OffPeak)</th>
<th>Boarding Volumes (Peak/OffPeak)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bathurst</td>
<td>52, 68, 137</td>
<td>2.4</td>
<td>15/9</td>
<td>471/505</td>
</tr>
<tr>
<td>Broadview</td>
<td>6, 47</td>
<td>2.4</td>
<td>15/8</td>
<td>1121/308</td>
</tr>
<tr>
<td>Dundas West</td>
<td>137</td>
<td>2.4</td>
<td>16/9</td>
<td>1019/408</td>
</tr>
<tr>
<td>Main</td>
<td>19, 60</td>
<td>2.4</td>
<td>13/8</td>
<td>988/303</td>
</tr>
<tr>
<td>Runnymede</td>
<td>85, 118</td>
<td>2.4</td>
<td>17/8</td>
<td>964/317</td>
</tr>
<tr>
<td>Sherbourne</td>
<td>8, 16</td>
<td>2.4</td>
<td>16/9</td>
<td>589/390</td>
</tr>
<tr>
<td>Spadina</td>
<td>0, 22, 137</td>
<td>2.4</td>
<td>15/8</td>
<td>403/536</td>
</tr>
<tr>
<td>Victoria Park</td>
<td>0, 74, 107, 126</td>
<td>4.5</td>
<td>16/7</td>
<td>1627/398</td>
</tr>
</tbody>
</table>

Method

As detailed in prior sections, the collection of pedestrian tracking data in closed spaces, particularly when dealing with dense crowds is an on-going challenge in the pedestrian research field. While several attempts have been made for automatic collection (electronic sensors or automated pedestrian detection from video), these methods are still not highly accurate at high crowd densities. In addition, they all require either access to or the ability to install collection equipment on site. The location of collection chosen for this study made this impossible, with the TTC not allowing access to security feeds, and no ability to have any installed for the research due to security reasons. As a result, an alternative method was required that could extract sufficient information on site, while respecting available resources and funding limitations.

To make collection feasible without the convenience of recording and off-site processing, a method of collection was devised that utilized a team of collectors to keep track of the changes along a discretized representation of each platform. As detailed in Figure 5.3, each platform was divided up into 8 sections, with the division lines occurring every three door-spacings at the midpoint between two doors. This resulted in each section being 16.5m in length based on the dimensions of the Toronto subway train used on the line. Observers were positioned at each of these divisions, as well as the entrances/exits to/from the platform.

Those at section divisions recorded three items:

1. The time pedestrians went left across the division
2. The time pedestrians went right across the division
3. The number of pedestrians who did not board a train from the left-side section (right side as well for the right most observer)

Observers at the entrances/exits of the platform recorded three items:

1. The time that pedestrians entered the platform
2. The time that pedestrians left the platform
3. The times of train arrival and departures

In place of traditional counting devices (or pen and paper tallying), to be able to properly handle this amount of information, recording was performed and assisted by the use of pre-existing smartphone
applications. The applications were chosen to ensure that they had the ability to record the time that buttons in their user interface were pressed, and the ability to configure multiple virtual buttons. With those collecting data providing their own devices for the study, one application each was chosen for the two popular smartphone platforms, Tio for iOS and Advanced TallyCounter for Android. Both applications log the time of all button presses and allow for data to be emailed for processing.

Processing

The goal in processing the collected data was to convert this raw information into time series information on the occupancy and distribution of pedestrians along the platform. As the boarding and alighting numbers from each train could not be feasibly collected without video access or a large number of volunteers (to cover all 24 doors), this information had to be deduced from the measurements made at the observed locations. Specifically, the boarding and alighting volumes could be assumed based on the flows into and out of each of the sections, while taking into account the locations of the entrances and exits. Because of the various situations that could occur that could not easily be automatically deduced using a computer script/code, this processing was done manually. To facilitate the process, all flows were grouped into 5-second time segments and were placed in order on a single Excel spread sheet. Three situations arose in the processing, and are shown in Figure 5.4.

The easiest case to handle, shown in (a), involved sections that had entrances/exits only in one direction. For these situations:

\begin{itemize}
  \item The number boarding was assumed to be all pedestrians in the section at the time of train arrival minus the number counted to remain by the observer
  \item The number alighting was assumed to be the difference between the total pedestrians out of the section in the direction of the exit and the total pedestrians into the section from the opposite side (also moving in the direction of the exit)
\end{itemize}

The second more problematic case (b), were where exits were in other sections in both directions. In these cases, it was often possible to examine the time of flows into and out of the sections to deduce their originating section (where alighting occurred). This included following flows to the exits, which allowed for filtering out flows that were not from the train (pedestrians entering during train dwell or crossing over sections to board in an adjacent section). In situations where it was still ambiguous, pedestrians were assumed to move towards the closest exit. In all cases,

The final case (c) was one where the exits were in the same section as that which was under examination. This was the most problematic to handle, with flows from adjacent sections coming into play, and was additionally complicated by flows to other exits in stations where multiple choices existed. To best approximate the probable origin-destination of pedestrians, the timing of flows were once again used; exiting flows out of the platform which were closest to the train arrival were most likely to have

![Figure 5.3: Observation locations, dividing platform into 8 sections.](image-url)
originated in the same section (given the 16.5m width of sections); flows to other exits were determined based on total exiting flows at those other exits and the time flows measured from other sections.

As a result of these realities, there was some uncertainty surrounding whether exact alighting section and target exits for every individual were accurate. This, however, was not of particular concern with respect to model development, which was focussed on the incoming pedestrians and how they arranged themselves before boarding. Therefore, what was important was correctly accounting for the overall impact of flows of alighting agents on section densities as they moved to the exits, rather than the disaggregate choices made by the agents.

Through this manual processing, the following set of data was extracted for use for both determining the model parameters (described in the subsequent section) and for validation of results:

- Total # of boarding pedestrians in each section upon train arrival (used for validation)
- The number of pedestrians in each section of the platform in 5-sec time increments
- The number of pedestrians entering the platform in 5-sec time increments
- The times of train arrivals and departures
- Alighting section and target exit origin-destination matrix

During the processing, fatally inconsistent errors were found in the collected data for one set of platform data, the peak period counts for Broadview Station. As a result, that data was discarded, leaving 7 stations for the peak period model and 8 for the off-peak period set.

### 5.3.2 Platform Section Targeting Data

This stage of data collection involved collecting information that could inform the platform-targeting component of the platform pedestrian distribution model. Specifically, three pieces of data were required in order to build the platform section targeting profiles for each station observed.

- An origin-destination matrix from the observed stations to others along the line
- The locations of exits at the destination stations
- The percentage split of use of each of the exits at the destination stations
As field data collection of pedestrian movements did not involve any verbal communication with those observed, both as to not affect results and due to their unfeasibility because of the volume of people, this data could not be simultaneously collected. As a result, a different source or combinations of sources were required.

**Origin-Destination Data**

The first source of information was the Transportation Tomorrow Survey (TTS) for 2011/2012. The TTS collects a wide variety of trip information from travellers across the GTHA, including those taken by transit users. The most recently collected data was across fall of 2011 and 2012, the second time period corresponding with the observed platform distribution collection effort. As part of the information collected in the survey is the starting and ending TTC subway station where travellers used the subway system as part of their trip. As only the start time of the overall trips is asked of respondents, when the subway portion of the trip occurred was not known. Therefore, the counts of subway boarding and alighting locations were taken for the entire morning peak period (6-9AM) combined rather than having it split for each time period.

Based on these volumes, an origin-destination matrix was constructed for each of the observed stations. This construction involved some processing; the destination station in the TTS data was the final alighting station (skipping any mention of transferring stations if the occurred), however the numbers using each side of the U-shaped line on either side of Union station was extractable from the TTS system. The Bloor-Danforth subway line, where all stations observed are situated on, has three key stations (Spadina, St. George and Yonge) that connect to the U-shaped Yonge-University-Spadina line (Figure 5.5). Of these three, two are connections that passengers would reasonably make if they have knowledge of the system, with Spadina requiring a lengthy walk (over 5 min) to perform the same transfer available at St. George. As a result, an assumption was made during processing that all travellers performed transfers at the other two stations. Volumes of alighters at these two stations were then calculated by apportioning all travellers transferring to the Yonge portion of the line to Yonge station, and the University-Spadina portion of the line to St. George. In addition, Yonge station values were calculated separately for the northbound and southbound directions. While St. George station has relatively equidistant level change locations to transfer between the two subway lines, Yonge station has transfers between the lines at opposite sides of the platform (west-side for transfers to the southbound platform and east-side for transfers to the northbound platform).

**Entrance and Exit Locations**

The second set of information, the locations of all of the entrances and exits, was collected manually by visiting all of the stations along the line. Part of this effort was undertaken during the field data collection at the 8 stations where platform movements were observed. The rest was performed with the assistance of an undergraduate student, David King, who noted locations of all entrances/exits to from the platform. As precise measurement in metres was not feasible in a reasonable length of time, the locations were recorded relative to their position with respect to the train doors. As trains were observed to stop at approximately the same location and the resolution of data collection along the platform was not particularly high, this was deemed appropriate for the purposes of this study.
Usage Splits of Exits

The last set of the required data was the percentage split of each exit used by those alighting at each station. This data was generously provided by the TTC, which undertakes studies periodically to gauge how their facilities are used. Data was collected which recorded the volumes of passengers going to and coming from the platform level at the various entrances and exits across the first half of 2011 at all stations in the network in 15-minute segments from 6AM to midnight. The split of alighters was then determined for all stations along the Bloor-Danforth line for the two directions, eastbound and westbound, separately. Most stations on the line have two separate platforms for each direction, allowing for exit splits to be calculated separately. For stations with single platforms serving both directions, an assumption was made that the splits were identical for both.

The amount of data and the breadth of locations made collection of all required data from the same period impossible for this thesis. Therefore, although the sourced data was collected in different periods across two years from the platform distribution field data, it was what was possible given the resources available. In their use, an assumption is made that the commuter population behaviour and destination did not significantly change over the two years, and was fairly consistent during the weekday AM commute.

Generating Targeting Profiles

Using the station O-D matrix, and the location and usage splits of the exits at destination stations, platform location targeting profiles for morning peak commuters were generated for each of the observed platforms. As mentioned in 5.2.2, the targeting profile is the probability distribution of a given pedestrian to target the various sections of the platform given a specific entrance. The targeting probability to a platform section at door $i$ in station $s$ ($T_{is}$) was then defined as the probability that an agent would target a certain spot on the platform to await boarding, given that he/she had knowledge of the alighting station and would target. The base targeting probability from each origin station was calculated using:

$$T_{is} = \frac{B_{is}}{\sum_j B_{js}}$$ (5.11)
where $B_{is}$ is defined by the following formula with $O_{sj}$ the number of people boarding at station $s$ and alighting at station $j$, and $S_{ji}$ the percent of passengers alighting who use the exit in station $j$ at door $i$ (0 if an exit is not at door $i$).

$$B_{is} = \sum_j O_{sj} S_{ji}$$

(5.12)

While this provides a method of determining the probability of pedestrians waiting for trains at specific sections, given that they would target a specific platform section, it does not distinguish between pedestrians entering at different entrances. Realistically, however, pedestrians who are aware enough of their commute to strategically position themselves would also be aware of the entrance to use to minimize walking distance. As a result, it would be expected that the probability that a pedestrian that had chosen to wait at a section had come from a specific entrance would have some inverse relationship with the distance between the section and the entrance. Therefore, to apportion the total section probabilities ($T_{is}$), an influence term ($I_{ei}$) for an entrance $e$ and platform section $i$, $d$ metres apart was first defined as:

$$I_{ei} = \frac{1}{d_{ei}^\alpha}$$

(5.13)

where $\alpha$ is a distance impact parameter that would need to be estimated. The probabilities could then be apportioned based on the ratio of the influence of an entrance at a waiting location to the sum of all influences from all entrances at that same location. Mathematically, this was combined with Equation 5.11 to result in the final probability $P_{eis}$ for a pedestrian entering at entrance $e$ in station $s$ to target waiting at the platform section $i$ as shown below. For final application in the model, the probabilities were normalized for each entrance.

$$P_{eis} = \frac{I_{ei}}{\sum_f I_{fi}} T_{is}$$

(5.14)

### 5.4 Model Estimation

Model estimation was performed using a genetic algorithm (GA), via a developed program that also handled the simulation of the platform and all agents. The estimation process is shown in the flow chart in Figure 5.6, and consisted of 4 key stages:

1. Generation of Initial Parameter Sets
2. Creation of GA Agents for each parameter set
3. Running of platform simulation for each GA Agent and evaluation of performance
4. Pruning and Mating of GA Agents

#### 5.4.1 Genetic Algorithms

Genetic algorithms are the best known class of evolutionary algorithms, which use mechanisms inspired by biological evolution to solve optimization problems [100]. To solve a problem or find the optimum
set of parameters, a population of agents is created, each with a set of parameter values encoded in a chromosome. Agents evaluate the success of each chromosome as it is passed to a fitness function, a function or a process that is run to produce a value that measures the success of the parameter set defined in the chromosome. Based on the performance of each agent’s chromosome, similar to the idea of natural selection, a portion of the agents die off, while the rest "mate" to replace them for the next generation; random mutations are also introduced to try to avoid premature convergence [100]. As this process is repeated for each generation, the population evolves, becoming better overall in solving the given problem. Convergence is eventually reached once a certain threshold is met with respect to the performance of the chromosomes when run through the fitness function.

There are two main types of genetic algorithm, binary and continuous, where the latter deals with real-valued parameters [100]. Given that the parameters being optimized for the proposed platform model are continuous, this form of the genetic algorithm was used in this study. It was implemented in custom software to use in estimating the optimal parameters of the platform model. There are many methods possible for selection, mating and mutation of agents, and those used in this study are detailed in the following sections.

**Figure 5.6:** Platform model estimation process
5.4.2 Parameters for Estimation

Based on the proposed model in the prior sections in this chapter, several parameters required estimation. In addition, a few more were included that would modify how simulation agents behaved and calculated variables of interest. All chosen parameters are summarized here with the ranges of each tested during the estimation. These ranges were set using results from preliminary runs examining proper orders of magnitude and expected range of values.

**Probability of Targeting -** $\beta(s)$ (0.0-1.0) The probability that an agent will be targeting a platform section (versus just moving via diffusion). This parameter was only estimated when peak-period data was used, and was allowed to vary between platforms from the 8 stations.

**Pedestrian Diffusion Constant -** $K_d$ (0.0-3.0) A parameter that controls the influence and speed of the diffusion component of the movement model as defined by Equation 5.5. As the value increases, agents would increase the speed at which agents walk in response to a density gradient. It was kept constant for all platforms, but allowed to vary between the peak and off-peak times.

**Motivation Parameter -** $\gamma$ (0.2-1.0) A parameter which dictates the tolerance of agents who are targeting in moving towards their preferred platform location. Mathematically, it defines the shape of the motivation function (Equation 5.8) by setting the value (representing distance in the model) below which the motivation would rapidly drop. Higher values would result in agents more stubborn about exact positioning on where they wait.

**Entrance Distance Influence Parameter -** $\alpha$ (0.0-2.0) A parameter that determines the drop in likelihood of an agent targeting a section from a given entrance as the distance between them increases. As the value increases, agents are less likely to target sections far away from an entrance.

**Minimum Required Density -** $\rho_{\text{min}}$ (0.0-1.0 person/m$^2$) The minimum local density required for an agent to be subject to diffusion.

**Maximum Density -** $\rho_{\text{max}}$ (3.0-5.0 person/m$^2$) The maximum density into which an agent will move.

**Distance of Aggregation -** $l$ (0.5-5 m) Distance used to aggregate density values (Equation 5.5)

5.4.3 Initial GA Agent Generation

As explained earlier, the continuous form of the GA was used for this study. As a result, the *chromosome* for each GA agent consisted of a set of real values, one for each of the parameters listed in the prior section. For the initial GA agent population, a size of 100 was chosen both for reasonable computation time, but sufficient in size to cover a range of values for the 13 parameters. The parameter set for each agent was generated randomly using uniform distributions for all ranges noted in the prior section.

5.4.4 Platform Simulator

In order to evaluate the performance of each of the sets of platform model parameters, an agent-based simulation model was developed. The structures used and the process of simulation are detailed in the sections to follow.
Platform Structure and Properties  In-line with the proposed model of pedestrian movement, platforms in the simulator were modelled to only permit movement along a single axis. For simplicity, each platform was assumed to have a uniform width, true for all stations observed except one (Victoria Park). Entrances to and exits from the platform by agents were permitted at the entrance and exit locations, respectively, placed at their real-world locations. Based on the specifications of the train cars used by the TTC, boarding and alighting was permitted at the 24 doors, each spaced 5.48m apart, across the 6 cars of the train. As the exact door of alighting could not be recorded for all pedestrians during data collection, simulation agents were assigned randomly to alighting doors in each section. In terms of properties, the platform stored distributions for targeting platform sections for each entrance, a collection to store the population of agents (both boarding and alighting), a collection to keep track of alighting agents reaching their respective exits, and a list of the divisions to use when tabulating the distribution of agents (to match the locations of observers during data collection).

Agent Structure and Properties  Agents were modelled as point entities (ignoring body size), each with a position, preferred max speed, actual velocity and targeting section, if applicable. Agent speed was determined based on the proposed diffusion and targeting model, as specified in Equation 5.10 for boarding agents, and Equation 5.6 for alighting agents.

Steps of Simulation  A summary of the steps taken for platform simulation is shown in Figure 5.7. As shown, first, the platform was first populated with the corresponding number of agents in the first 5-second frame of the collected data. Following this initial step, the main simulation process began. Each second of simulation involved first introducing any agents either entering at platform entrances or alighting from trains, if present. Boarding agents were removed at the end of the dwell period of the train. All agents then had their positions advanced, after updating their velocity as defined in the proposed platform model (Section 5.2.3). Finally, exiting agents were processed, using the exit times in the collected data to remove them from the platform simulation; agents whose time had not yet come were left at their position. After each 5-second period, the number of agents in each section (with divisions shown in Figure 5.3) was recorded for evaluation purposes. This simulation process was repeated for each platform in the set upon which the model was being estimated, with peak and non-period platforms handled separately.

5.4.5 Evaluating Simulation Results: Fitness Function Definition

Following each simulation run, the results were scored against the collected data. Due to issues in data collection (the precise start and end times of collection for each observer were not identical), time periods before the first train and after the last were ignored. Scoring was performed on all other 5 minute segments using the fitness function defined in Equation 5.15, which was based off a standard sum of squares of residuals between the model and collected data for all platforms $p$, periods $r$ and sections $i$.

$$FF = \sum_p \sum_r \sum_i \frac{(actual_{pri} - predicted_{pri})^2}{\Delta r_T} \quad (5.15)$$

In addition, while the overall process was being modelled, predicting the distribution of agents at time of train arrival was the target; therefore, the score for each time period was weighted based on its proximity to the next train arrival ($\Delta r_T$), with those closer weighted much more heavily. This was
also done to minimize the influence of periods directly following train dwell, as boarding and alighting processes were not modelled in detail in the simulation. For the same reason, the periods of train dwell were also ignored.

Figure 5.7: Platform model simulation steps
As the platform simulation involved drawing from the target distribution, a single simulation run was not sufficient to properly evaluate the performance of the parameter set. As a result, each set of parameters was evaluated using the average score of 10 simulation runs, both as a common value used when dealing with stochastic systems and for computational speed reasons.

5.4.6 Subsequent Population Generation Creation and Convergence

The production of each successive generation of agents in a genetic algorithm involves three steps: selection of agents of the current population based on the fitness function, mating of agents to produce offspring, and random mutations of chromosomes.

Selection  The selection of agents is the process of choosing which agents should survive for the next generation. Many methods have been proposed for selection, but they can generally be divided into two main types, stochastic and truncation selection. In stochastic selection types, each agent is given a probability of surviving based on their fitness function value relative to the population, and agents are selected randomly based on those probabilities. For truncation selection, agents are ordered based on their fitness function score, and a specified portion of the best scores are selected. For this application, the truncation selection method was used with half of the agents retained for mating and mutation.

Mating  The mating of agents involves combining the chromosomes of parent agents in specific ways to produce new agent chromosomes for their offspring. For a binary GA, the most common way of combining is crossover, which like their biological analogue, involves splicing the parent chromosomes at random points and exchanging sections in between. In the continuous form of GA used here, blending the values is more appropriate to result in new values for the offspring [100]. The values of each parameter value $\theta$ in the chromosome for the offspring are calculated as a linear combination of the parent’s parameter values as shown in the equation below, with $\delta$ replaced by $1 - \delta$ for the second child. While $\delta$ can be held constant for all parameters, to allow for more variation, new values of $\delta$ were chosen for each child parameter.

$$\theta_{\text{child}} = \delta\theta_{\text{parent}} + (1 - \delta)\theta_{\text{parent}}$$

(M.16)

Parent selection for mating was performed using a random process, with each agent paired randomly with a second until all agents were paired. As a result, after mating, the population consisted of the original truncated parents, and their two offspring per pair. These agents were then fed into the final step of generation production, mutation.

Mutation  The mutation of agents, analogous to genetic mutation in biology, involved the random change of parameter values based on a specified mutation rate. It attempts to prevent the reaching of local minima in the GA optimization process by random searches outside of the values generated by the parent agents. For this application, a 5% mutation rate was used, corresponding to a random change of one parameter for around every 1.5 agents. To avoid loss of the most optimized agent for each generation, the best performing agent was excluded from the mutation step.

Convergence  Convergence of the GA was chosen to be stabilization of the best performing set of parameters over a significant number of generations (20). At this point it was assumed that a better
solution would not be found with further iterations. These optimal parameters values were used as the starting point for a more focussed second GA run, where search of parameters were limited to within the neighbourhood of these values (± 10%) and mutation step was skipped. This was performed on the chance that mutation resulted in a better solution being passed over and not revisited within the 20 generation threshold.

5.4.7 Estimation Results and Parameter Analysis

A summary of the final parameters is shown in Table 5.2 for the three models estimated: peak period, with and without targeting for agents, and diffusion-only off-peak period. The second model, using the peak period data but only having agents diffuse, was estimated to gauge the benefit in introducing the ability of agents to target platform sections. As shown in the table, there was a decided improvement in the evaluation of the fitness function when the targeting component of the model was enabled. In addition, a marked increase in the diffusion coefficient occurred with only agents able to diffuse, most likely to compensate for the inability of the diffusion model alone to account for agents who spread out further than would be explained by the volume of agents entering along side them at their entrance. The maximum density above which pedestrians would not move into a section was also marginally higher in the diffusion-only model. Given, however, the relatively low overall pedestrian volumes for the observed platforms, even during peak periods and particularly in off-peak periods, the likelihood of the model producing densities even at the lower level of the peak-targeting model was low. As a result, this difference is not believed to be of particular significance, and in both cases the values are higher than Fruin’s highest level of service F at 2.17 pers/m² [19].

<table>
<thead>
<tr>
<th>Model</th>
<th>(\beta) (s)</th>
<th>(K_d)</th>
<th>(\gamma)</th>
<th>(\alpha)</th>
<th>(\rho_{min})</th>
<th>(\rho_{max})</th>
<th>(l)</th>
<th>(FF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak-Targeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bathurst: 0.511</td>
<td>1.724</td>
<td>0.468</td>
<td>0.1</td>
<td>0</td>
<td>3.3</td>
<td>4.5</td>
<td>35140</td>
<td></td>
</tr>
<tr>
<td>Dundas West: 0.750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Main: 0.203</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Runnymede: 0.987</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sherbourne: 0.388</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spadina: 0.756</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Victoria Park: 0.828</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak-No Targeting</td>
<td>—</td>
<td>2.596</td>
<td>—</td>
<td>—</td>
<td>0</td>
<td>3.5</td>
<td>5</td>
<td>57149</td>
</tr>
<tr>
<td>Off Peak</td>
<td>—</td>
<td>1.479</td>
<td>—</td>
<td>—</td>
<td>0</td>
<td>3.0</td>
<td>2.5</td>
<td>25952</td>
</tr>
</tbody>
</table>

Compared to the peak-period diffusion only model, the off-peak parameter values showed some significant differences. While the maximum density term \((\rho_{max})\) was comparable, but again not believed to be of particular significance, both other parameters were significantly different. The initial motivation behind collecting off-peak data was the assumption that the pedestrian populations would behave differently from the commuter population during the peak periods. As a result, it was assumed that pedestrians travelling off-peak would be less likely to be familiar with their destination station and, therefore, less likely to target specific waiting areas. The lower diffusion coefficient \(K_d\) found during the GA estimation process is consistent with this hypothesis, with a reduced need to force spreading even at low density gradients to compensate for the lack of agent targeting. The much lower density aggregate distance \(l\) was also of note; this would act to reduce the range of influence of each agent in
affecting the spread of their neighbours, making agent diffusion more dependent on those nearby. Small pockets of high density would therefore not have far reaching effects and, therefore, should result in more concentrated agent populations.

The peak-period model with agent targeting had three model specific parameters. The motivation parameter $\gamma$ of 0.468 illustrated an inclination of agents to wait quite close to their preferred location, with motivation falling only once they were within 5 metres. The entrance distance influence parameter $\alpha$, on the other hand, had an optimal value of 0.1, leading to quite a gradual falloff of the influence of an entrance based on distance when determining targeting probabilities. Both of these relationships are illustrated in Figure 5.8.

![Figure 5.8: Effect of estimated values of $\gamma$ and $\alpha$ on target motivation and entrance influence.](image)

Lastly, the probabilities of agent targeting for each station showed a distinct spread of values over the observed stations. While no studies have been done in North America, let alone Toronto, of the percentage of subway riders who do have preferred waiting locations, the sole study conducted in South Korea provided a measure of comparison of 76% of commuters targeting train cars for boarding, and nearing 60% who target to minimize walking distance at the alighting location [97]. The targeting probabilities generated by the GA produced values that were similar to this Korean study for three stations (Bathurst, Dundas West and Spadina), in the neighbourhood for two (Sherbourne and Victoria Park), and two stations with extreme values (Main and Runnymede).

While uncertainty surrounding the myriad of sources used as input into the platform model, and the need to have used one set of targeting profiles for the entire peak period, are likely factors in this wide variability, a portion is believed to be due to the different entrance placements across the set of observed platforms. Namely, it would be expected that more regularly spaced pedestrian entrance flows would allow for the diffusion model alone to be better at modelling the spread of pedestrians, as individuals are more likely to be able to enter near their preferred waiting spot. To examine this effect, a pedestrian entrance volume centrality (VC) index was defined as shown in Equation 5.17, and calculated for each platform.

$$VC = \sum_e V_e C_e$$

(5.17)
where \( V_e \) is the percentage of pedestrian volume entering at entrance \( e \) and \( C_e \) is a measure of centrality of entrance \( e \), ranging from 0, if the entrance is at the centre of the platform, to 1, when it is located at either end. Plotting the probability of an agent to target \( \beta \) against the VC (Figure 5.9) shows a rising relationship, but significant spread, possibly due to the time of flows not being considered, in addition to the aforementioned issues of accuracy in the sources used to generate the targeting profiles.

![Figure 5.9: Targeting probability as a function of the spread of pedestrian entrance volume](image)

### 5.4.8 Validation

Validation of the final models was conducted against the pedestrian distribution at train arrival, the main reason for the development of the model. To accomplish this, the following measure was used: the ratio of deviation (\( \epsilon \)) between actual and predicted counts for each section \( i \) over the total volume of agents (\( Q \)) at the time of train arrival (Equation 5.18).

\[
\epsilon = \frac{\sum_i |actual_i - predicted_i|}{Q}
\]  

(5.18)

This measure was calculated for all periods just prior to train arrival for all simulation runs of the models shown in Table 5.2. Analysis of this measure was performed in two ways, by looking at the distribution of \( \epsilon \), and examining the values of \( \epsilon \) at different volume levels, with an expectation that model performance would improve at higher levels.

The validation results for the three models are shown in Figures 5.10-5.12. The peak-period model with agent platform section targeting (Figure 5.10) showed moderate ability at an aggregate level to predict platform distribution, with an average \( \epsilon \) of 0.48. Removal of the targeting component of the model resulted in a noticeable degradation of predictive ability (Figure 5.11), reinforcing the importance of its inclusion. The diffusion-only model behaved better with the off-peak data (Figure 5.12), even with the lower overall volumes, supporting the hypothesis that non-commuter passengers would be either less aware of their destinations in order to target or less interested in being time efficient.

How \( \epsilon \) translates to predictive accuracy at a more micro level is not easily apparent with this aggregate measure. A cursory glance at the deviation values produced by the peak-period targeting model does indicate that significant numbers of distributions are not being corrected predicted. There are, however,
several mitigating factors that need to be considered before determining whether the modelling method holds promise.

Figure 5.10: Performance of the peak period model with agents permitted to target locations, showing overall boarding distribution deviations

Figure 5.11: Performance of the peak period model with agents only diffusing, showing overall boarding distribution deviations

First, the collected data contained a high number of low passenger volume measurements, even during peak periods. As a result, in contrast to the other known study of platform distribution modelling [97], the passenger volumes analyzed fell mostly in a region of low to moderate density that which would naturally be expected to result in higher values of $\epsilon$. The reasons for this are both due to the equation used to calculate $\epsilon$ (Equation 5.18), which would result in large changes even for the misplacement of a few individuals, and a larger chance of a draw from the targeting profile that did not correspond with the particular train. At higher volumes (>50), a noticeable improvement is seen in $\epsilon$. As changes in
pedestrian platform distribution are more likely to affect train dwell at higher volumes, the inability of the model to consistently predict distribution at lower volumes is not of significant concern.

This was further supported by examining the performance of the complete peak period (with targeting) model on its ability to predict the volume of pedestrians in the highest section. As this volume is the limiting factor in setting train dwell time, the model’s performance in this specific aspect is key in understanding it’s usefulness when used in understanding crowd effects on line operation. To enable this, a new measure, $\epsilon_M$ was defined as follows to focus on the max volume section $M$:

$$
\epsilon_M = \frac{\text{actual}_M - \text{predicted}_M}{\text{actual}_M}
$$

The values for all simulation runs and for all stations are plotted in Figure 5.13. As shown, the model was able to accurately predict the number in the maximum volume section to a high degree; however, significant spread was observed, particularly at lower volumes. Finally, as shown in the histogram, the model was more likely to under predict, with an average rate of -6%. This small deviation on average does, however, lend some promise to the model to account for the average effect of platform distribution on train dwell over a reasonable length of time.

The second mitigating factor was the diverse sources of data used in the estimation of the model. As mentioned earlier, aggregate peak-period subway O/D data from the Fall 2011/2012 TTS survey was combined with aggregate peak-period splits in entrance use collected in Spring of 2011 provided by the TTC in developing the targeting profiles. This profile was then used in the estimation of the model, optimized against platform distribution data collected in Fall of 2012. While this is significantly more data than has been available in prior studies, the disparate time periods do present a problem.

Lastly, major sources of error that could not be properly quantified, but would act to raise the average value of $\epsilon$, were both in issues in the only feasible method of data collection, and in properly modelling the collection environment within the simulation model. As data was collected manually by periodically spaced human counters, errors inevitably occurred that would affect the section counts against which
estimation was conducted. While attempts were made during processing to correct them where possible, it is highly likely that a number of them remained, and would increase the value of $\epsilon$. In addition, given limited resources and to avoid unduly influence pedestrian movement, the resolution of collection was not particularly high with the number of observers used; more frequent spacing might have helped in better tuning the model.

Of more significant impact was in handling pedestrians who entered close to train departure. While data collectors were instructed to keep track of those unable to board, they were not always consistent in keeping track; this was handled by clearing off all who remained in sections at the time of door closure in both the processed data and the simulation model. Unfortunately, situations arose where this resulted in differences between the total numbers of people on the platform in the data compared to the simulation model, artificially increasing the value of $\epsilon$. This would be particularly problematic at low volumes, where differences in overall numbers would be magnified in the deviation value.

Next, issues arose in properly translating the position of data collectors on the platform to divisions used to section the platform for analysis in the simulation. With data collectors instructed to stand halfway between doors at specified locations, these locations were calculated precisely using TTC train car specifications; however any deviation from this precise value by those in the field would not be capture, neither would any variation in train stopping location. In sections with higher numbers of people, variations in collected versus simulation section divisions could result in significant differences in realized pedestrian distribution.

Finally, as a simplified and consistent rectangular platform was assumed for all locations modelled in this study, variations in that structure could affect model results. This would be particularly true in situations where entrances were positioned at the extreme ends of the platform, but could have also played a role in affecting the exact time of entrance and exit into the platform at other locations.
5.5 Conclusions

Given all of these issues, the reasonable parameter values obtained, and the expected behaviour of the model under the various conditions under which collection occurred lends promise to the developed model in predicting platform pedestrian distributions. It is believed that it provides a more comprehensive modelling framework than has previously been attempted, and could produce better results if more time consistent input data was available.

Research paths remain in further refining the model, particularly in dealing with more complex platform layouts, such as varying width and for platforms serving multiple tracks, and for higher volume situations. Overall, however, the proposed model provides a more comprehensive modelling framework than has previously been attempted to better enable platform design and assist in better predicting the effects of station crowd management on line operation. The latter use requires linking of the platform model with a subsequent model explaining the dwell process; this includes better understanding of the alighting process of passengers, and how passengers positioned along the platform by the platform model will make their way onto trains. In addition, the framework is amenable to implementation within Nexus, to provide clear linkage between local platform behaviour and the overall network paths of each agent. This method of incorporation is detailed in Section 7.3.2, with its overall impact analyzed within the Toronto case study in Chapter 8.
Chapter 6

Nexus Development: Stage I

To examine the feasibility of Nexus, an initial prototype was developed. The main goal was to show proof-of-concept of the feasibility of the services-oriented approach of distributed computing for transit systems modelling. Its focus was to show its capability in both accelerating simulation of large-scale networks and permitting the creation of a network simulator out of separate specialized simulators (line, station, surface/street).

With the focus on examining the feasibility of the approach at a structural level, priority was given to the computing architecture and the development of the coordination and data handling components; the individual modules (line, pedestrian and station simulators, and transit assignment method) were built as simplified constructs, to be replaced later with more sophisticated and realistic implementations. As a result, the system described in this chapter is a partial implementation, leaving development of a comprehensive control system to instigate disruptions and actuate response, as well as inclusion of comprehensive crowd simulation for the case study presented in Chapter 8.

In the following sections of this chapter, the method of prototype implementation of each of the key components in the framework is detailed. This is followed by description of two test networks that were used to evaluate the computational feasibility, and first case example of how the system might be used. The key aims of this initial prototype were, therefore, to illustrate the ability of the framework to (1) interface disparate simulation components, (2) improve simulation performance across multiple computers, (3) provide detailed trip and pedestrian level of service information, and (4) measure network-level effects of the propagation of delays resulting from overcrowding and disruptions.

6.1 System Components

The prototype was built leveraging the Microsoft .NET Windows Communication Foundation (WCF) to handle inter-application communication; WCF is a services and distributed computing architecture that permits and simplifies communication between clients and services running both on the same computer or spread across computers over a network. WCF communications are specified in terms of service interfaces and data contracts, allowing for a standardized method of communication between components independent of their internal workings. It also permits two-way communication between clients and services in a multi-threaded fashion, necessary for system feedback and intra-pulse transfer of vehicles between simulation modules. Data handling was accomplished using MySQL server, leveraging its ability
for rapid reading and writing of simulation data, and quick searching and processing. Transit network structure, including routes, stops and schedule information, is stored within database tables, and is read into object structures for use during simulation.

The client/services structure of the prototype was as follows. Individual simulation modules (station, line and surface) were developed as WCF services to which the coordination engine connects as a client in a reverse hub-spoke topology to push data and run the simulation. The coordination engine in-turn also runs as a service for connection by the transit analyzer user front-end to indicate a network to load and simulation to perform. Transit assignment was incorporated within the coordination engine to allow for direct access by the engine to all pedestrian and network structure data and to avoid transferring large amounts of pedestrian data. During simulation, all pertinent data (such as vehicle locations and sensor data) is relayed from the modules at each synchronization pulse through the coordination engine to the analyzer for in-run visualization, while simultaneously being logged within the database. The basic process followed for running a simulation is provided in Figure 6.1.

![Figure 6.1: Process model for running a simulation for the stage I system](image-url)
6.1.1 Transit Structures and Basic Operation

Geographic Structure

For this initial implementation, map coordinates were simplified using a x-y coordinate system with respect to an arbitrary point in each map. As the test networks used (detailed later in the chapter) were not real world locations, this sufficed. All locations were assumed to be point locations. For subway stations, doorways were placed as the same coordinate as the station, with platforms spaced 20 metres apart from each other. Routes were assumed to follow along stop locations using straight lines, without specific paths specified.

Transit Operation

As with geographic data, transit service was kept simple. Each line for both surface and underground routes consisted solely of a single branch; vehicle units were kept dedicated to that route and limited based on the number required to provide service at the specified headway. In addition, each run of a vehicle assumed visits to all stops in both directions before returning to the original starting point. Vehicles were released at the starting stop of their route based on their schedule, being held if they arrived too early for their next cycle.

6.1.2 Coordination Server

The coordination engine acts as the main controller of the system. To allow for system logging and pedestrian processing, and to avoid excessive communication points of failure, all communication between network modules occurs via the coordination engine; the database is also only accessed and updated through this component. Within the prototype implementation, the engine is specified via two separate interfaces. The first defines how clients (such as the network analyzer) interact with the engine, while the second is a call back interface necessary for data exchange with component services.

The client interface exposes three main functions, one to load network data, one to specify simulation parameters, and one to control simulation. All network configuration data (both specifying how system components interact and the structure of the transit network) and simulation parameters (population, simulation period, agent settings) are stored within the database. The simulation control interface signals the simulation to be run for a specific duration (allowing for simulations to be run piecemeal if needed), taking in any control signals to be relayed to components, and indicating whether visualization is to be enabled, which triggers relaying of all sensor data to the connecting client. While running the simulation, the engine requests from the transit assignment module the set of agents that depart from home during the current cycle and pulses all component simulators to advance for the cycle duration and return sensor information when complete. Synchronization of all component simulators was set at a frequency multiple of the internal clocks of the component simulators, but not so infrequent as to excessively delay transfer of pedestrians moving between components not on a vehicle. To avoid excessive calls to transit assignment, the fetching of agents occurs in super cycles.

The call back interface exposes functions to relay vehicle arrivals and departures (through transfers of complete vehicle data and their passengers), indicate expected vehicle arrival and departure times, update agent trip information and allow for general system logging. The vehicle-related functions are critical for components to be aware of vehicle movements that might have effect in the current simulation cycle and to provide the capability of the system to inform pedestrians of when pertinent vehicles might
arrive. Agent trip information is updated by relaying pedestrians to the transit assignment module to retrieve the next trip leg anytime they move through doorways, board or alight. The engine handles all issues in determining which service component should receive pedestrians or vehicles who are transferring based on internal tables, allowing connected components to not concern themselves with how the transit network is distributed amongst simulation services and how transfers occur.

6.1.3 Transit Assignment

The transit assignment module used a simplified estimated travel-time based assignment method in this implementation, without consideration of transit schedules. A choice set of attractive paths is first determined by considering all transit paths with a maximum of two transfers between considered origin and destination stops (those within a walking distance of 500m). Travel times were estimated based on link speed, with transfer penalties added; the quickest route by this method was then chosen for each pedestrian. Origins and destinations were determined in the prototype networks through generation of a generic population with origin and destination points uniformly distributed across the region. In this implementation, routes are fixed, and any communication with the module during simulation is strictly for providing route information to agents and logging of trip performance.

6.1.4 Surface Simulator

The surface simulator is responsible for the simulation of all surface routes, boarding and alighting at stops, and for handling the movement of agents from their starting points to bus stops or station entrances, and to their final destinations. In the course of simulation, it is also responsible for properly informing all connected stations whether a vehicle should be expected in the current cycle to maintain synchronization. In the implementation used in the stage I prototype, pedestrian motion was not modelled, but instead agents were attached to their target doorway or bus stop with their arrival time indicated in a discrete-event fashion. Vehicles, conversely, did have their motions modelled, and were advanced forward by the appropriate distance based on their speed until they reached the next stop, accelerating and decelerating as needed. When a vehicle arrived at a surface stop, boarding and alighting was performed. The set of alighting agents were determined explicitly based on the trip itineraries of each agent. Boarding was set to occur in two steps, first to board those who have arrived at the stop by the arrival time of the vehicle and repeated to additionally capture those who arrived during this initial dwell period when the vehicle was ready to depart. Those unable to board due to capacity limitations were left on the platform and logged the event in their trip memory. No consideration was made of order of vehicle arrivals at each surface stop, allowing for implicit passing of vehicles. Upon reaching a bus platform at a station, the vehicle data was converted to the appropriate format for communication and passed to the coordination engine; vehicles remained there until processed by the station and continue to the next stop after receiving the updated list of passengers.

6.1.5 Line Simulator

The line simulator is similarly responsible for the simulation of all underground routes, and for maintaining all information on agents on-board the vehicle to allow for their transfer to transit stations for boarding and alighting processes. Within the simplified implementation, and comparable to the surface simulator, train movements are specifically modelled. Unlike the surface simulator, the line simulator
does not handle any boarding or alighting processes, which occur at transit stations. Additionally, train passing was set to not be permitted. To mimic signal blocking present in train control systems, vehicles were only allowed to move if at least two hundred metres behind the train ahead, and maintain a minimum of one minute station headways, allowing for some propagation of delays through the train network. In contrast to the surface simulator, trains can avoid direct interaction if the lines are independent and vehicles do not share routes, allowing for multiple line simulators to be used to simulate service.

6.1.6 Station Simulator

The station simulator is responsible for the simulation of passenger agents as they transfer between bus platforms, train platforms and station entrances. The original impetus for the presented framework centred around being able to incorporate the simulation of large crowds of people at many stations to understand both pedestrian flow throughout the network and the resulting impact on transit service performance while considering feedback. At this stage, however, the developed station simulator was greatly simplified, jumping people between origins and destinations within stations and dealing with boarding and alighting from vehicles in a similar fashion to the surface simulator. One key difference, however, is that the trains consist of four transit units each, and agents randomly choose a unit to target prior to boarding. Dwell time is then taken as the maximum time for dwell processes to complete across all transit units. Prior to leaving, the module checks whether the calculated departure time is permissible by the line simulator, or if the vehicle should be held; in the latter case, control of the departure time of the vehicle is handed to the line simulator. This mechanism allows for complete two-way control as necessary to simulate real-world situations where trains can be held beyond their dwell processes. As the station has to be able to handle multiple vehicles coming into multiple platforms from multiple applications, each simulation round proceeds in two steps, first waiting for the expected arrival times of the next vehicle at all connected platforms, and then proceeding with any dwell processes. As vehicle arrivals are occurring on different threads from the main simulation code, cross-thread signalling is used to force vehicle dwell processes to occur in the proper sequence.

6.2 Test Networks

To evaluate the described prototype simulation system, two hypothetical networks of varying size and complexity were developed. The first consisted of 10 stations, two independent subway lines and a surface network with six non-connecting bus lines, with two-minute headways on all routes (Figure 6.2). The second consisted of 40 stations, three subway lines (two of which were integrated) and 17 bus lines with transfer points, with two-minute headways on trains and five-minute headways on buses, to test the ability of the system to handle larger and more complex networks (Figure 6.3).

6.2.1 Performance Evaluation

First, the networks were evaluated by measuring simulation run times for varying population sizes to examine the scalability of the architecture in moving many people around between components. To focus the analysis on any issues with network latency due to the time needed to pass data between services, the simulation was performed on a computer capable of running the networks below its computational capacity. High frequency service (2 min subway, 5 min bus) was used to increase overall system capac-
ity to enable testing of higher population sizes. Nevertheless, the networks both saturated at around 120,000 agents being able to reach their destination over the course of a three-hour simulation. The resulting run times for both network sizes, and distributing the larger network onto three computers, are shown in Figure 6.4. Given that additional processing time is necessary for the transit assignment and surface simulator components to handle increasing numbers of people, the communication system itself appears fairly insensitive to population size for large networks. A 10-fold increase in population (20 to 220 thousand agents) resulted in an only 50% increase in simulation run time when run over three computers. As expected, distribution of the under-capacity simulation onto multiple computers degraded performance due to latency; however, the effect was minor and diminished percentage-wise at higher populations.

Next, the benefit to simulation speed of the distributed architecture was gauged by evaluating performance as the number of computers was increased for a 1.5-hour simulation with a fixed population size of 100,000 agents on the large network, and a synchronization pulse period of 8 seconds (Figure 6.5). To mimic the behaviour of the system under more detailed crowd simulation, sufficient busy cycles were added to the station simulator to overburden the main testing computer. Due to the asymmetric nature of distributing network components across computers, the values shown are the times obtained for the best configurations. It should be noted that the added three computers were of lower computing power from the main test system, and components were shifted only until the computing processor of the lead system was running below its full capacity. As seen in the results, the use of multiple computers can significantly decrease the overall run times. The decrease is more pronounced when moving from one computer to two, with diminishing returns for larger numbers of computers as the overall computing capacity becomes sufficient to run the simulation. As the framework is in part targeted at expediting large-scale simulations, with extensive and detailed pedestrian movement, one would expect greater improvements once a full pedestrian simulator is incorporated at the next stage of system development.
Lastly, the effect of synchronization period on run times was examined by varying the synchronization period between 4 and 24 seconds for the large network while keeping internal simulation periods for the surface and line simulators at 2 seconds (Figure 6.6). As one would expect, longer synchronization periods generally allow for faster run times, but the relationship is not directly inverse with a clear trade-off occurring, resulting in run times eventually levelling off and starting to rise. It is believed that inter-component vehicle transfer signalling time begins to dominate any efficiency gained from reducing pulsing frequency in the higher range. This is not, however, believed to be of any real consequence with synchronizations with excessively long periods presenting practical challenges in properly transitioning pedestrians between stations and the surface streets.

6.2.2 Transit Network Performance Examples

The value of a simulation system is in its ability to produce relevant performance measures to evaluate the conditions of the network being analyzed in both normal and unexpected situations. The vehicle arrival/departure logging, agent trip logging and sensor feedback mechanism of the framework enables such data collection by collecting network movements of agents (vehicles and pedestrians) and component-specific performance measures exposed by each simulator. Examples of sensors of interest at the network level could include platform volumes, flow through key passages or escalators, headways, dwell times and operating speeds; one example is shown for the changing dwell times of vehicles for

Figure 6.3: Schematic of the large-scale test network
Figure 6.4: Scalability in population size and number of computers for stage I system

Figure 6.5: Improving simulation performance with increasing number of computers for stage I system

different passenger loads (Figure 6.7). Agent-centred measures of performance could involve measuring overall trip waiting and travel times, as well as inability to board vehicles due to capacity limitations.

In addition, the goal of the framework in providing a mechanism to interconnect and synchronize disparate simulators was to go beyond current commercial software and allow for the use of more sophisticated simulators to examine the effects of operational changes on the overall system. This might include, for example, varying forms of line control or schedule recovery strategies following unexpected disruptions. While the focus of this prototype system was the main coordination engine, initial components of the control and detect subsystem were also developed to illustrate the capability of the system in capturing the network-level impact of subway disruptions, both on the transit service and its agent passengers.

To this end, a temporary breakdown or emergency disruption was simulated by holding a train for
Chapter 6. Nexus Development: Stage I

106

Figure 6.6: Diminishing returns with increasing synchronization periods for stage I system

30 minutes at a platform in a key station at the meeting point of two of the train lines in the large prototype network. Next, three line control responses were considered. First, doing nothing, but still maintaining 200-metre train spacing and a minimum of 1 min headways at stations; second, doubling this spacing during the period of the disruption to better space trains once service is restored, third, holding all trains in place for the duration of the disruption on routes that would pass through the platform (not a good response). The more conventional response of turning back trains and deploying shuttle buses was considered, but was left for the full case study implementation (Chapter 8).

To gauge the effect of these three response strategies, simulated sensors were placed within the station simulator to report back platform volumes, agent trip logging was used to examine the consequences to overall agent travel times, and vehicle event logging was used to examine the effect on bus and train dwell times. Figure 6.8(a), which shows the platform volumes over time, marks the disruption period as a shaded region. As seen in the figure, from the base condition (blue) and no response after disruption (red), significant and persistent average platform volumes across the network are observed well after the end of the disruption, with nearly two hours passing before volumes back down to normal. The method of response is also key with significantly worse crowding and platform volume build-up found where trains were held in place (green) through the disruption period. These effects are paralleled in the agent overall travel time distributions (Figure 6.8(b)), where increased average platform volumes translate to longer commutes due to higher waiting times (Figure 6.8(c)). A

Lastly, significant train disruptions can have far reaching consequences, going beyond just the subway service to spread to connecting surface routes. Therefore, the ability of the prototype to capture these dynamics was also examined by looking at the 5-minute average dwell times of surface and underground vehicles (Figure 6.8(d)). As shown in the figure, train dwell times are persistently higher well after the disruption has cleared. The effect on bus dwell times is more muted, both showing marginally reduced dwell (as riders are trapped in the subway network) and marginally higher later on as they emerge into the bus network. Overall, the system appears to produce results consistent with what would be expected under these conditions.
Chapter 6. Nexus Development: Stage I

107

(a) Sub-Capacity Volumes  
(b) Over-Capacity Volumes

Figure 6.7: Vehicle dwell time distributions for the large stage I network

6.3 Conclusions

This chapter presented the initial prototype implementation of Nexus, implementing the conceptual frameworks defined in Chapter 3. A prototype system and computational architecture were constructed to illustrate feasibility. Through tests on two hypothetical networks of varying size, it was shown that computation time could be decreased by distributing components across multiple computers without running into communication bottlenecks for large numbers of agents.

This prototype represented the first stage in the development of Nexus. As a result, many steps remained in operationalizing the framework to enable testing of response strategies to alleviate congestion due to peak demand or unexpected disruptions. In addition to computational efficiency improvements, the temporary prototype modules needed to be replaced with more robust models. Of particular importance was the incorporation of MassMotion for detailed pedestrian movements at busy stations. In addition to improvements in individual components, to accurately model the influence of local changes on network performance, the interface between the subway and stations needed to be more properly modelled. As a result, the models developed in the prior two chapters (4 and 5) needed to be incorporated. These modifications, including computational and structural improvements, were handled in the second stage of the prototype development, described in the next chapter.
Figure 6.8: Effect of line control on network performance for the large stage I network
Chapter 7

Nexus Development: Stage II

In order to both allow for the main test network (the Toronto transit network) to be implemented in Nexus, and to incorporate the two pedestrian models developed in this thesis, some significant changes were required to the initial prototype (Chapter 6). This chapter details modifications that were made, from both a computing and modelling standpoint.

7.1 Structural Changes to Transit Service Data

The way network information was organized required some structural changes that led to operational changes in how transit service was simulated. This was driven by the move from the hypothetical test networks to real-world networks, and a desire to allow the system to be easily adapted to these networks. For this reason, compatibility was sought with the Google Transit Feed Specification (GTFS), a standard way that transit agencies now deliver their service structure data.

7.1.1 Google Transit Feed Specification

The GTFS is a set of tabulated data files that specify the various characteristics of transit service; it is also the input method for transit service into Google Maps. The GTFS standard specifies a minimum set of required data, with many fields left as optional (a full set of the specification can be found at: https://developers.google.com/transit/gtfs/reference). In general, the format specifies agency info, the list of all stops and their locations, the transit routes of the service, and all vehicle trips (sequence of stops and stop times, and grouping of trips assigned to each vehicle). In order to have sufficient data to be able to construct the network within Nexus, it was assumed that agencies provided the following in addition to the minimum:

- Trips Table: Head Sign, Block ID, Shape ID in order to combine trips into route branches
- Shapes Table: To draw routes and calculate trip distances

7.1.2 Supplemental Data

The data contained within the GTFS data structures was supplemented by two additional sets of information. The first were the locations of all the entrances of stations (the doorways), which act as
the transition points between the street and station simulators. The second set were the locations of entrances and exits for each of the platforms, required for modelling of platform operations. At this stage of development, the percentages of transit users using each of the entrances and exits were also assumed to be an input, rather than being determined endogenously.

7.1.3 New Structure of Transit Data

The network structure data in the first prototype was simplified to allow for easy creation of the test networks. Mainly, this revolved around the use of cartesian coordinates for locations, each route having a specific set of stops with no branches, and vehicles each limited to a single route, circulating continuously after their initial release. Moving ahead from this simplified system towards one capable of handling real-world networks, changes were required. These were executed to comply with the need to be compatible with the GTFS standard, and to allow for a more realistic method of how structural and service data would be represented. It also resulted in a network structure consistent with the framework specification (Section 3.4.2 and Figure 3.6). The modifications to the initial prototype system are summarized below:

- Replacement of cartesian coordinates with WGS64 (longitude, latitude)
- Route groups replace routes in the prototype (with main route number, name and agency)
- Route groups contain all related route branches (same route number, different stops sequence)
- Route branches are defined to collect all vehicle trips visiting the same sequence of stops
- Sets of trips replace vehicles circulating on the same route, specifying the scheduled departure times at each stop
- Vehicles are now disassociated from routes, allowing them to service trips on multiple routes
- Basic station data was added containing distances between doorways and platforms, and platform entrance/exit layouts

Based on these modifications, the structure of the SQL tables to store this information on disk, and C# data objects for use during simulation were finalized. Conversion of the GTFS and supplemental data into these tables was performed using a separate piece of written software. The method used is heavily dependent on the specific data formats used by transit agencies when distributing their GTFS data. As a result, explanation of this processing is presented in Chapter 8 in the context of the GTHA case study.

7.2 System Components

All system components were modified to accommodate the changes in transit network structure and operation, and in preparation for the much larger scale of the case study network compared to the test networks implemented in the initial prototype.
7.2.1 Transit Assignment

Some of the more significant changes involved the method used to assign agents to routes. To begin, the agent population was switched away from being automatically generated as a hypothetical population. It was, instead, assumed to be set as an input, with specification of agents’ origins, destinations and departure time. As with the initial prototype, the method of transit assignment assumed that agents chose their quickest route based solely on time with a set transfer penalty, ignoring capacity considerations of the chosen routes. Unlike the initial implementation, however, schedules as sourced from the GTFS data, were used as part of this travel time calculation. In the course of finding this quickest path, however, a set of 10 possible paths were generated for each agent. This was done to later allow for a more sophisticated transit assignment method that would involve possibly choosing other paths to iteratively balance transit users through the network.

Without load balancing in this prototype implementation, the transit assignment process was essentially reduced to a network path finding operation. As before, in order to limit the computation time, number of transfers was limited to two, which allowed for more efficient searching. However, while the algorithm used in the initial prototype was sufficient for the size of the network used at the time, the much larger network used in the case study could not be quickly searched, taking on average 1 sec per agent. While computational performance of transit assignment was not a key goal of this thesis, this was deemed infeasible for use for the approximately half-a-million agent population, requiring several days to perform. As a result, the solution was reformulated to reduce as much as possible the search space of routes examined when performing path finding. The general steps followed are illustrated in Figure 7.1.

As before, the end result of this process was a set of possible paths to take. This path include information not only of the routes taken and stops visited along the trip, but also specific station doorways used when transition between the surface and train stations. The entire network path of each agent was stored only within the transit assignment module; information for individual segments, and sufficient info of the subsequent segment is passed to each simulator component as the agent makes its way through the network. This allows for complete control of the trip to be contained within the module, allowing for any updates as needed if a change in path needs to occur (e.g in the case of a disruption).

The final major change in the transit assignment module was the development of the ability of the module to reroute a specific set of agents when prompted. This was essential to enable the disruption simulation and response strategy management features of Nexus. This involved allowing for specifying disrupted segments (as a series of affected stops) and implementation of the Update Agent Path Choice function specified for the transit assignment module in the framework (see Section 3.4.1). Disrupted segments were taken into account by checking for them during the travel time calculation portion of the path finding algorithm (disallowing trips that went through that segment). The Update Agent Path Choice function, in-turn, was used to trigger a new path choice routine for a subset of agents that would be directly affected by the disruption, based on their current location. This function was only called where the response strategy allowed for agents to re-route.

7.2.2 Surface Simulator

The surface simulator was modified to work with the new format of trip-centric transit service, and to take advantage of scheduled times at stops and route paths from the GTFS data. Even with these modifications, however, the surface simulator was coded with the assumption that it was present as a
Preparation Stage: Route Network and Trip Transfer Network Graph Creation

- Transit Network Structure
  - Load Trips Within Simulation Period
  - Construct Route Graph
  - Construct Route Transfer Lookup Tables (how each route connects to other routes at each stop)
  - Construct Trip Transfer Time-Based Graph (links each trip with its connecting trips at all transfer points)
  - Construct Schedule Structure (list of trip arrivals at each stop for each route)

Routing Stage

- Buffer Stops Within Walking Distance of Origin and Destination
- Build set of Routes departing from Origin Stops and arriving at Destination Stops
- Look for common routes in the origin and destination route lists
- Using Route Transfer Lookup Table, look for origin routes that connect with destination routes
- Create agent paths with 1 transfer and calculate travel time based on trip schedules
- Sort paths by travel time, store 10 quickest, select fastest
- Create agent paths with 2 transfers and calculate travel time based on trip schedules
- Using Route Transfer Lookup Table, look for origin routes that connect with a destination route via a third route
- Create agent paths with 0 transfers and calculate travel time based on trip schedules
- Sort paths by travel time, store 10 quickest, select fastest

Figure 7.1: Transit assignment path finding algorithm

stand-in for the purpose of the prototype, eventually requiring replacement with a more established and existing software.

Surface Transit Structure

As Nexus has a key aim of enabling a dynamic representation of the network, allowing for on-the-fly modification of routes, the structure of the transit network is not fixed as is often the case in other software. As a result, instead of a priori creation of the network, creation of all routes, stops, vehicles and vehicle schedules are performed on-the-fly during the initial data loading stage using the common network structure data provided by the coordination server. This creation step involves first going through all routes of the network, and creating internal route structures for those dealing with both fixed and variable surface routes. Next, stops are created to be placed along these routes that are either surface stops, bus platforms at stations or station doorways. Finally, the vehicles belonging to the component are identified and are added to a pool of unreleased vehicles.
Treatment of Vehicles

To operationalize this new method, vehicles were treated as individual agents, moving through a set of trips, possibly from different routes, as they made their way through the simulation period. At each time step of the simulation, any vehicles that were to begin their initial trip of the period or start a trip on a new route were added to the appropriate route and set to reach the initial transit stop of the trip in the next time step. All vehicles on all routes were then processed, updating their location and dealing with dwell operations for any vehicle that had reached a stop. Speeds for vehicles were set based on the scheduled times at stops sourced from the GTFS data; these times are normally refer to departure, with no data provided on assumed dwell values. For this simplified system, these averages were assumed to be exact with no spread, resulting in surface vehicles were forced to keep on schedule. Vehicle movements were assumed to occur in uniform steps based on these average speeds, and no consideration for interaction between vehicles (allowing for vehicles to pass freely). The path taken by each vehicle is set by the associated path information provided in the GTFS data. If the vehicle is at the end of the trip and its next trip begins at the same stop, it will be attached to the new route; if not, the vehicle will be temporarily removed from the simulation until its next trip release time. Vehicles that have completed all of their trips are removed from the simulation after dropping off their final passengers.

Dwell Operations for Surface Stops

Dwell operations for surface stops were performed within the simulator. The list of alighting agents were retrieved from the vehicle and their route information updated with the transit assignment module to advance them to the next segment on their journey. Boarding passengers were then added to the vehicle on a first-come-first-serve basis up to the capacity limit of the vehicle. Dwell time was calculated based on the number of boarding and alighting agents, with the assumption that boarding could only occur through the front door, with either front or rear door used for alighting. With the simulator lacking a detailed model of the distribution of agents within the vehicle, it was assumed that one third of alighting agents would use the front door. Each agent was assumed to take 3 seconds to board and 1.5 seconds to alight [15]. Finally, dwell time was calculated as the higher of the two durations of agent movement between the front and back doors.

Dwell Operations for Station Platforms

Dwell operations for station platforms were performed within the station simulators. As a result, when a vehicle was determined to enter a station during the simulation step, an appropriate vehicle transfer data object was created with all relevant vehicle data, including information for on-board agents. This vehicle information was then sent to the coordination server for transfer to the appropriate station simulator. When the vehicle is ready for departure (determined by the dwell time set by the station simulator), the updated vehicle data is received and the vehicle moves towards the next stop.

Treatment of Agents

All agents begin and end their trip in the surface simulator. With the focus of the prototype system on mass crowd movements in stations, agent movement for the surface simulator was greatly simplified. Movement was handled in a discrete-event based manner, with agents jumping between positions based
on their average walking speed, and an assumption that they move in a straight line between their origin and destination. This also assumes that agents do not run into any congestion as they make their way, and ignores the effect of traffic lights in delaying their journey. Transitions between the street and station doorways were set to occur at each simulation cycle, while transitions between the street and station platforms occur at the moment of arrival.

7.2.3 Line Simulator

As with the initial prototype, the final prototype implementation of Nexus maintained a consistent approach with that used for the street simulator for network creation and handling of vehicles. As a result, the same modifications made to the street simulator were applied to the line simulator with respect to having vehicles make a series of trips, possibly on different routes. As before, agent boarding and alighting was not handled by this component, instead occurring within the confines of the station simulator. In addition, in handling vehicle movement, the line simulator continued the approach of maintaining a buffer distances between vehicles approaching the same platform to mimic a signal block system, with this spacing set as a parameter. In contrast to surface vehicles, as the focus of the prototype was in understanding the effects of crowds on train service, train movements were not forced to remain on schedule. Instead, vehicle movements remained as in the Stage I prototype, with acceleration up to a max speed (20 m/s) and deceleration directly modelled.

One key modification that was implemented surrounded the ability of the line simulator to control dwell times at stations. In the initial prototype implementation, generally dwell time was set by the station based on the duration required for boarding and alighting processes. This was overwritten only in situations where a vehicle had not yet reached its release time at the beginning of each trip, keeping train doors open until departure time. Where the line was simulating non-frequent service commuter rail, vehicles were also programmed to be held if they had not yet reached their specified departure time for that stop. In other cases, such as where the vehicle was held up due to another vehicle head or told to hold due to a disruption, a vehicle at a station would leave its doors closed after the initial dwell process. This unrealistic behaviour was corrected in the final implementation, with the ability of the line simulator to have trains remain at their stations with doors open for an arbitrary period in cases where they needed to be held by the line controller.

Finally, modifications were made as part of the expansion of the system to be able to implement a small number disruption situations to illustrate proof-of-concept. These disruption events were translated into two actions:

- Holding of a train at a station for a length of time (e.g. if a passenger assistance alarm is triggered)

- Removal of train after dumping all passengers (train going out of service)

Given the complexities of instigating on-the-fly service changes, particularly the need to reschedule and reassign trips to vehicles (an entire area of research on its own), response strategies implemented for the prototype did not involve these types of modifications. These were left as future work. Instead, more easily implemented responses (e.g. holding trains along the line or modifying train spacing as a whole) were programmed to examine how they affected the network’s disruption response.
7.2.4 Basic Station Simulator

Modifications to the station simulator were minimal between the initial and final prototypes. As before, the station simulator was responsible for handling movement of agents within stations, including during dwell operations for both buses and trains. Movement of agents in the simplified station simulator was performed the same way, by jumping them between entrances and platforms at their average speed based on a matrix defining distances between each of the various station end points, therefore ignoring interactions. To simplify operations, it was again assumed that platforms could handle only one vehicle at a time, with any additional vehicles queuing until space was available. Due to the workings of the line simulator, such conflicts were limited to platforms serving surface vehicles.

One key difference between the initial and final prototypes, however, was the handling of platform operations and setting of dwell times. Platform operations for train platforms were modified to incorporate the diffusion model detailed in Chapter 5 to produce the distribution of agents at each time point; how this was performed is explained later in the chapter. This distribution of agents was then used for loading train cars; the individual dwell times for each car was calculated using Weston’s formula (Equation 2.1), and the overall dwell time was set as the maximum across the transit units making up the train.

As before, once boarding and alighting are complete, a request is made to the coordinator to check whether the vehicle is ready to depart. This decision is made by the line and surface simulators for trains and surface vehicles, respectively. If the vehicle is requested to be held at the station, doors are set to remain open until informed by the appropriate simulator that departure is permitted. At this point, a vehicle data package is constructed to send back to line or surface simulator which contains information about the newly boarded agents, including how they might be distributed along the vehicle.

7.2.5 Integration of MassMotion

The simplified station simulator is aimed at stations deemed not significant enough to warrant detailed simulation. For more complex and congested stations, the application of full pedestrian simulation might be required. For these cases, the 3D pedestrian simulation software MassMotion (an overview is presented in Section 2.4.3) was chosen. In order, however, to have the software act as a station simulator within Nexus, modifications and additions had to be made to the software to enable it to properly interface.

MassMotion currently has no publicly accessible application programming interface (API), which would allow for on-the-fly control of MassMotion simulations. The framework and prototype, however, requires connecting pedestrian software to have the ability to dynamically control events like the opening and closing of doors, introduce agents and controlling their targets, and retrieve information about overall agent flow or about specific agents after they have made their trips. The presence of the development team close to the university allowed for an unparalleled level of access to the inner workings of the software and ready support, enabling significantly easier and more customizable integration of the software with the system. Through this collaboration, it was possible to produce a custom version of MassMotion with the needed abilities.

MassMotion is written in standard C++ and so the interface between the Nexus prototype and MassMotion required a few layers. The first was a C++ class that acted as the main API for the MassMotion software. This API exposed a set of functions to perform a broad range of operations.
These included functions to load MassMotion project files, advance the simulation by a frame (default period of 0.2 seconds), create agents with a specified set of tasks to perform, and retrieve agents who had exited in that frame. Also included were functions that queried the simulation to be used as sensor data in the prototype, including the number of agents on specific floors (such as platforms) or flow through key locations like vertical transition elements. Lastly, a set of functions were created to improve the ability of MassMotion to model trains and passengers waiting on platforms. Specifically, these included the ability to treat certain sets of floors as train cars and platforms, allowing for actions such as emptying all cars when the trains depart, opening all doors associated with a train, and indicating the arrival of a train to waiting agents. How trains were modelled within MassMotion is described in the context of the case study in Chapter 8.

In order to interface this API-modified version of MassMotion with Nexus, two additional steps were required. As Nexus was developed in the C# programming language, the first step involved developing an intermediate C++ .NET class wrapper to mediate between Nexus and the standard C++ MassMotion API. This intermediate wrapper handled all data type conversions that differed between the Microsoft .NET libraries and the standard C++ libraries.

The second step was developing the C# application that would act as the station simulator component, handling all data transfer and communication with the coordination server, but relying on MassMotion for the simulation of the movement of agents. This application was structured in a similar fashion to the simplified station simulator described in the prior section. This included implementing the same interface functions and order of signalling of other system components in order to properly work within the prototype, but using a time-step simulation approach to be compatible with MassMotion.

At the beginning of simulation, the application is first responsible for preparing MassMotion to run the specified station model. This is sent to the component in a compressed data stream, extracted into a temporary folder and specified to be loaded within MassMotion. Also sent with these files are a mapping of which MassMotion agent entrance and exit portals correspond to the stop identification numbers in the network structure. This information is sent to the API-modified version of MassMotion to allow for easy reference of entrance and exit portals during simulation.

The method used to handle agents and vehicles are detailed in Figure 7.2. Agents arriving through the doorways from the street or alighting at vehicles are introduced at the corresponding portals in the MassMotion station model. They are then given a destination. For targets consisting of a single portal, like a bus or doorway, the assignment is direct. On the other hand, when the target is a train with several transit units, a method is required to distribute the agents across the portals associated with train cars. Two methods were put into the application for this purpose. The first was having an agent target a specific portal or set of portals (choosing the closest), in front of which it would wait until the doors were open to board. The second used the platform distribution model of targeting waiting locations along the platform; this method is detailed later in this chapter.

Dwell operations begin upon train arrival as indicated by the line simulator. To simulate arrival, a set of commands are sent to MassMotion; these involved indicating that a train had arrived to the appropriate set of agents and opening the train doors. MassMotion allows for doorways to have a priority direction; using this method, alighting agents are first allowed to leave the train cars, followed permitting boarding agents to pass. Boarding is allowed to continue until either the number boarding causes the train car to reach capacity or no agents are present within 4 metres of all train doorways. Once dwell operations have completed, a command is sent to MassMotion to clear all agents from the appropriate
Agents are retrieved from MassMotion either when they exit through doorways, or when cleared from train cars or buses. At this point, information about the agent’s journey through MassMotion is recorded, including entrance, walking, waiting and exit times and the level of congestion experienced. As with the simpler station simulator, agents leaving through street exits are processed at every synchronization point, while agents leaving on vehicles have their information transferred along with the vehicle data on departure. The number of agents that are unable to board are also noted.

During the simulation period, the application provides feedback of the conditions inside the Mass-
Motion simulator. This information includes volumes of agents on platforms and flows through vertical circulation elements, particularly stairs and escalators. At the end of the simulation period, a command to finalize simulation is sent to the MassMotion engine to finalize its output files, and these files are compressed and transferred back to the coordination server to be stored to be available for later analysis.

### 7.2.6 Network Analyzer

The network analyzer is the main user interface for Nexus. As mentioned in Chapter 3, the analyzer has a few key responsibilities in determining the network and population to load, any disruption scenario or response strategy, and to visualize and analyze results. In the initial prototype, the analyzer was limited to loading a pre-specified model with scenario and response built into the software. Visualization was also limited to displaying the test networks (stations and routes) and vehicle locations.

In the final prototype, significant changes were made to this component. The first change involved a complete overhaul of the visual design of the user interface, including a move to a multi-page format with menu navigation, allowing for the software to handle loading of the network, running simulation and presenting analysis results. An example of the interface is shown in Figure 7.3; details of its construction, however, are not critical to this thesis so have been omitted.

![Network Analyzer](image)

**Figure 7.3:** User interface of network analyzer

The second major change surrounded modifications to the network and vehicle display system to allow it to handle real-world networks. These changes included the use of a third-party open source map UI C# control (XAML Map Control) to allow easy plotting of WGS 64 spherical coordinates (longitude/latitude) and zoom and pan features to navigate the map. Modifications were made to the control to enable it to handle hundreds to thousands of vehicles simultaneously, and present live feedback of simulation results on mouse-over on vehicles (showing capacity and basic info), routes or stations (displaying platform and station volumes). In addition, the control allowed for map tiles to be loaded behind overlaid shapes to provide context on the location of various stops, stations and routes.

Finally, an analysis section was inserted to allow for quick processing and presentation of results from individual simulations. This was required due to the size of the data output from each simulation and the need to look up information from multiple data sets in order to make sense of the simulation...
output. For example, vehicles when arriving and departing from a stop log their specific vehicle ID, the stop ID, and information about the dwell process (arrival time, departure time, number boarding, number alighting, etc); information about vehicle route and stops to make sense of the data is stored in a separate database. Code was written to automatically process and produce graphs on stop, route and agent metrics. These are detailed within the context of the case study in Chapter 8.

7.3 Incorporation of Pedestrian Models

A key driver of Nexus is the general idea that research carried out on individual transit network components would benefit from the ability to examine whether there were any far reaching effects across the network. In addition, there exist specific situations and phenomena at the interface between transit system components that have a clear and direct ability to impact multiple facets of transit system operation. As a result, it is of key interest to incorporate models that better describe behaviour at interfaces and in other major locations to examine their influence on network performance.

To this end, the two models of pedestrian behaviour developed in this thesis (Chapters 4 and 5) were built into the station components of Nexus. The behaviour of passengers on the platform has a more obvious ability to influence line operation and system capacity. On the other hand, whether modifications to how pedestrians make their way between levels would have a noticeable impact on operations beyond the station was an open question. In addition to examining their individual impact, it was also of interest to determine their combined impact. The following sections detail how these two models were incorporated for testing.

7.3.1 Vertical Circulation Choice

The vertical circulation choice models, specifically the final disaggregate discrete choice set developed in Section 4.5, were implemented only within the MassMotion pedestrian simulator (detailed in Section 4.5.7). With the models specific to software where crowd flow is being simulated in detail throughout the station, they were not implemented within the simpler station simulator.

7.3.2 Platform Dispersion

In contrast, the platform dispersion model was adapted for use, with some modifications, in both the simpler station model and within MassMotion. As detailed in Chapter 5, the model involves two components, one that spreads agents out based on surrounding density through a diffusion like process, and a second optional one for agents to target specific waiting locations. These two components were handled differently when applied in the two applications.

For the simpler model, which used a simplified approach of walking time based on distance between the origin and destination locations within the station, the method of implementation was similar to that used during model estimation (Section 5.4). Agents are assigned an entrance to their target platform when added to the simulation; this is performed randomly based on feasible entrances to the platform from their current location. Next, based on an input probability, it is determined whether an agent will target a preferred waiting location. If so, based on the specific platform at their destination, and the position of possible exits that could lead to the point on their trip (platform or doorway), a waiting location is randomly selected for the agent.
Chapter 7. Nexus Development: Stage II

The initial simplified station simulation model was discrete-event based, jumping agents between locations rather than individually simulating their steps, the platform distribution model works in a time-step fashion. For easy incorporation, the discrete-event nature of the software was maintained; the simulation to update the platform distribution was performed at two time points, on train arrival and at expected train departure (based on a preliminary estimation of dwell time using agents on the platform at train arrival). This simulation update step involved updating the model in one-second time-steps by first adding any agents who had entered in that second, and then advancing the simulation based on the process described in Chapter 5. The final distribution of agents at the end of the update procedure was relayed back to the dwell operation function to calculate dwell times.

For the MassMotion simulator, the diffusion model was eschewed for the existing model of agent waiting behaviour within the software. In addition, as MassMotion automatically performs pathfinding for agents between origin and destination, the choice of entrance into the platform was also left as is. What was incorporated was the waiting section targeting portion of the platform model for agents who were determined to be targeting. This was accomplished using an assumption that a series of evenly-spaced waiting-location portals (one for each door) were present in the model for the length of each platform. Based on the known target station of the agent, a specific exit was selected (using the same method as with the simpler simulator), and this resulting location was translated into a target waiting platform. Agents were then commanded to spread out an wait for 2 seconds, and then hold their location. For agents who were not assigned to be targeting, the default spread out method of waiting was used.

7.4 Conclusions

The expansion of the initial Nexus prototype into its final form was significant. The modifications included performance improvements to enable Nexus to handle large networks, changes in service structure and geographic coordinate systems to be compatible with real-world transit systems, successful interfacing with MassMotion to enable detailed crowd simulation at key stations, and incorporation of the developed pedestrian routing and behavioural models. These changes were initiated in order to prepare Nexus to model the Toronto transit network; this case study is presented in the next chapter.
Chapter 8

Large Scale Case Study - Greater Toronto Transit Network

8.1 Objectives

This chapter presents a case study of the Toronto Transit Network to illustrate the capabilities of Nexus. As the first major test case of the system, the scope of the study was limited to feasibility and sensitivity analysis, with a fully validated network left for future work. The specific goals were, instead, to demonstrate its ability to:

- Import a real-world multi-agency network into Nexus
- Handle the number of agents and vehicles which would exist in an AM peak-period situation
- Coordinate multiple instances of MassMotion station simulators
- Examine the effects of passenger distribution along platforms on network performance
- Simulate disruption events via runtime commands
- Simulate response strategies set at runtime
- Provide detailed analysis of system and agent performance

The Toronto network was chosen both for geographic reasons (given this thesis was conducted in Toronto), for its importance as Canada’s largest network, its well known issues with overcrowding, and the wealth of data available both in describing the network as well as its users. This case study focuses on modelling and conducting tests of the system during the AM peak-period rush, where significant crowding issues are present.

Given its size, the use of MassMotion, and the current state of computational optimization of Nexus, this case study requires significant processing and memory power. The analysis in this chapter was conducted on a computer with a 6-core 3.6 GHz Xeon processor, 32GB memory (normally well under 16GB actively used), and a solid-state hard drive.
8.2 Background

The Toronto Transit Network is Canada’s largest transit network, serving over 1.6 million passengers daily, 900 thousand of whom use the subway network. Service is provided along 3 subway lines, 1 intermediate capacity rapid transit line, 11 streetcar and 141 bus routes. It is also characterized by having an extremely high level of integration between its surface routes and subway network, with bus or streetcar service generally having their origin at one of the 69 stations, and seamless transferring between the modes. Additionally, the subway network is known to be currently running at capacity, with issues of severe crowding at key interchange stations [1]. As a result, it presented itself as an ideal real-world network for a system aimed at better taking into account these various interactions between adjacent modes and the effects of crowds.

The focus of this initial version of Nexus was to better understand the interaction between crowds at stations and the subway service. However, the bus network in the Toronto transit network plays a primary role in providing feeder routes into the subway network. As agent trips are followed from origin to destination, this necessitated modelling of all surface routes in addition to the subway system. In addition, due to the high level of inflows into the City from surrounding regions (see Section 8.3), some method was required to introduce these flows into the Toronto network. To test Nexus’ flexibility to handle multiple agencies and multi-agency trips, many of these flows were introduced by direct modelling of the transit systems of the surrounding municipalities and the GO train network. For systems where insufficient public data was available (mainly lack of GTFS), auto access to GO transit stations was assumed.

With these additions, the final network consisted of several agencies, over a dozen rail lines, over a hundred stations and several hundred bus lines. The characteristics of each agency are shown in Table 8.1.

<table>
<thead>
<tr>
<th>Agency</th>
<th># Surface Routes</th>
<th># Train Routes</th>
<th># Stations</th>
<th># Stops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toronto Transit Commission</td>
<td>177</td>
<td>4</td>
<td>69</td>
<td>10922</td>
</tr>
<tr>
<td>Brampton Transit</td>
<td>44</td>
<td>0</td>
<td>0</td>
<td>2389</td>
</tr>
<tr>
<td>York Region Transit</td>
<td>133</td>
<td>0</td>
<td>0</td>
<td>4659</td>
</tr>
<tr>
<td>MiWay (Mississauga)</td>
<td>96</td>
<td>0</td>
<td>3065</td>
<td></td>
</tr>
<tr>
<td>GO Transit</td>
<td>0 (omitted)</td>
<td>7</td>
<td>63</td>
<td>268</td>
</tr>
</tbody>
</table>

The period of simulation analysis was the weekday morning peak-period from 6 to 9AM. To allow time for vehicles to enter the network and have an existing population of agents at 6AM, the simulation period was set from 5-9AM.

8.3 Data Sources

As would be expected, a significant amount of input data is required in order to construct a large scale simulation model. This section details the various sources used, and any processing that was required in order to get data into a format usable by Nexus.
8.3.1 Transit Network Structure

The data used to build the transit network structure and service information was mainly sourced from public GTFS files (see Section 7.1.1 for description) provided by the transit agencies in the GTA region. As the process of distributing public GTFS data is still a relatively new concept in the region, limitations existed in being able to acquire GTFS data from the several agencies in the region corresponding with the same time period. While this is not ideal, as a calibrated model was not a specific goal for this thesis, these time disparities were not believed to be a critical issue.

While GTFS files are the standard method of input of transit data into Google Maps, as mentioned in Section 7.1.1, the standard is not particularly strict, with many tables and fields optional, and no consistent format provided for text fields. As a result, the conversion of the GTFS data into a format suitable for use by Nexus was not a direct or easy effort, because of gaps in data or inconsistency in naming or numbering conventions, even within the same data set.

To convert the data within the GTFS files into a format compatible with Nexus, a program was written that could accomplish this goal. In preparation, a visual examination of the GTFS files of the various agencies was conducted to determine any patterns in naming conventions that could be exploited in extracting network structure. For example, it was found that it was possible to automatically extract TTC station names, and categorize the two types of stops (street stops and platforms). The occasional gaps or errors in data were reconstructed where possible or taken from other sources where information could not be automatically filled in (such as missing route path data for the GO train network). The list of agencies for which GTFS files were available, the corresponding date range, and any key gaps or alternate sources are provided in Table 8.2. One key issue was the low quality of the GTFS data provided by GO Transit. In addition to the missing shape data, specific platforms used by trains within a station were not named, and while train trip information was provided, the assignment of these trips to vehicles (their block) was omitted. As a result, as vehicle assignment to specific trips was not part of the scope of the work, and unnecessary for simulation, it was assumed that each train only served one trip.

<table>
<thead>
<tr>
<th>Agency</th>
<th>Date of GTFS</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toronto Transit Commission</td>
<td>Sept 2013</td>
<td>Incorrect distances between stops (reconstructed)</td>
</tr>
<tr>
<td>Brampton Transit</td>
<td>Aug 2013</td>
<td>Missing distances between stops (reconstructed), misplaced stops</td>
</tr>
<tr>
<td>York Region Transit</td>
<td>May 2014</td>
<td>None</td>
</tr>
<tr>
<td>MiWay (Mississauga)</td>
<td>Sept 2013</td>
<td>Missing distances between stops (reconstructed)</td>
</tr>
<tr>
<td>GO Transit</td>
<td>June 2014</td>
<td>Missing run blocks, shape info (sourced via CanMap and manual additions for new stations, route extensions)</td>
</tr>
</tbody>
</table>

In addition to this data, two additional sets of information were manually collected: the locations (longitude and latitude) of all station doorways, and the positions and associated doorways of all platform entrances and exits (with respect to leading edge of the associated subway trains) for the TTC subway network.

The software developed to convert this raw GTFS and supplemental data to the structure used within Nexus consisted of the following steps:
1. Reading in all GTFS and supplemental data files

2. Constructing main system components for all agencies (a common street network, one rail network per agency, and all rail stations)

3. Constructing all train and surface vehicle route groups and branches (and their associated vehicle trips) using a new set of unique IDs

4. Extracting stops for all agencies (physical stops shared between agencies were assumed to be separate) using a new set of unique IDs

5. Detecting stop transfer clusters (stops between which agents will transfer routes, both inter and intra-agency)

6. Writing all final stops, routes, trips, components, etc. following the table schema of the Nexus input database

7. Constructing station layout files for all station components (both simplified and MassMotion), and storing the compressed project files within the database

8.3.2 Agent Origins, Destinations and Departure Times

The Nexus prototype takes, as input, three items at minimum in order to generate a set of agents: the origin and destination of the agent trip and the departure time. For this case study, this information was sourced from the 2011/2012 Transportation Tomorrow Survey.

The Transportation Tomorrow Survey (TTS) is a wide ranging traveller behaviour survey conducted every five years in the Greater Toronto and Hamilton region. The survey is one of the largest of its kind in North America, randomly sampling 5% of the population in the region. The resulting data contains both detailed demographic information as well as a travel itinerary for a typical weekday. A section of the survey is dedicated to transit users, collecting data like the routes taken, access and egress modes, and boarding and alighting subway stations. Also collected is the type of trip, describing the origin and departure locations (e.g. home-based work trip). In presenting the data, the region is divided up into over 3500 zones, with their size based on density (smaller zones are used for areas inside the City of Toronto and its immediate neighbours).

As privacy is paramount for the survey, information acquired from access to the survey data generally has to be in an aggregate and anonymous format, and requires special permission. For the purpose of this case study, information was extracted that provided the surveyed number of users (scaled to a full population) that commuted from each origin zone to each destination zone, in departure time divisions of 10 minutes. This provided one of the required pieces of information (the departure time), but a method was then needed to synthesize a population of agents to correspond with the origin and destination zones.

This method of generation of precise starting and ending points involved a few steps. As a calibrated model was not a goal of the case study, one possible method would have been to randomly choice locations within each origin and destination zone. This, however, was found to present problems for the transit path finding algorithm of the platform; as transit stops are strategically placed to maximize accessibility of the population, blind generation throughout each zone often resulted in points in uninhabited areas,
too far from any transit stop. As a result, the method below was used to generate more realistic origin and destination points.

1. For zones inside the City of Toronto, the publicly available tables of Toronto address points (accessed via the City of Toronto Open Data portal) was used; this table provides longitude and latitude coordinates for the street addresses in Toronto, and identifies their type (e.g. high density residential, restaurant, office building, etc.). These addresses were joined with the TTS zone files to link each address to its underlying TTS zone.

2. For zones outside the City of Toronto, addresses were not available; instead, the following steps were followed:
   (a) CanMap (produced by DMTI Spatial) shape files were accessed via the University of Toronto Map and Data Library for the Province of Ontario; these shape files contain comprehensive mapping data. While the CanMap Address Points data set would have been ideal, this was not available. Instead, the Land Use shape files from the Route Logistics package for 2013 was utilized.
   (b) Inhabited land use regions were isolated from this shape file (removing regions such as forests, parks, etc)
   (c) The resulting shape file was intersected with the TTS zones ex-City of Toronto shapes to produce a new shape file consisting only of occupied land-use regions with a field indicating their TTS zone
   (d) Points were randomly generated within each land use zone

3. The address (City of Toronto) and randomly generated (outside City of Toronto) points were exported to a comma separate values file for import into the platform for agent population synthesis

4. Origin-Destination matrices were extracted from the TTS data server divided into 10-minute departure times (from 5AM to 9AM), and by their trip-type: Home-based Work, Home-based School, Home-based Discretionary, and Non-Home Based

5. Code was written to then perform the population synthesis, and consisted of the following steps:
   (a) Load in all O/D data by time period, and all address points; store by zone index
   (b) For each origin-destination pair and 10-minute time segment, do the following based on the number of individuals travelling between the origin and destination zones:
      • Randomly choose an address point with the appropriate type (e.g. for home-based work, an origin with a residential type and a destination with a commercial or office type)
      • Randomly choose a departure time in the 10-minute time segment
   (c) Create a list of objects containing this agent data (representing around 510,000 individuals) to be sent to the system

The final distribution of the origin and destination points created for the entire morning period is shown in Figure 8.1. As expected, agent origin points are much more widely distributed, with destinations concentrated in the central city and other known growth hubs. As local transit networks for Ajax and
Figure 8.1: Distribution of origin and destination points for the case study
municipalities southwest of Mississauga were not modelled in this case study, trips that both began and ended locally within these regions were excluded to avoid unrealistic network assignment (e.g. driving to a GO station, only to travel one station and drive back).

### 8.4 Agent Network Path Calculation - Transit Assignment

The transit assignment procedure, essentially an all-or-nothing schedule-based path-finding process in the Nexus prototype, was detailed in Section 7.2.1. This approach was applied to the agent population generated for the GTA case study, using a full representation of all of the transit routes detailed earlier in the chapter. While an accurate and calibrated transit assignment was not an expected outcome of this process, requiring an iterative process to account for capacity considerations, some effort was made to have agents make reasonable choices. The following assumptions were made as part of this effort:

- Agents were assumed to have a walking limit of 500 metres for a surface stop, and 1000 metres for a train station
- If stops were not found within the specific limit, the access mode was assumed to be driving, and the range of search was extended up to 20km, with the nearest 6 stops selected
- Transfer penalties were arbitrarily set as two minutes per transfer, with a limit of 200 metres to transfer between surface stops 600 metres for subways stops of the same agency, with that search distance doubled for inter-agency trips
- While fares were not incorporated as part of the assignment, this was incorporated with an assumed and arbitrary value of 20 minutes per additional agency, to provide some level of penalty but with an awareness that this would need to be more accurately set
- Total numbers of transfers were limited to two, which was sufficient to route 96.5% of input agents
- The fastest route, after transfer and multi-agency penalties were added, was always chosen

The final result was a population of around 492,000 agents, 20% with a direct route, 42% using one transfer and 38% with two transfers. This compared to splits from the source TTS data of 42%, 37% and 21% for direct, one-transfer and two-transfer trips, respectively; this pointed to calibration issues with both the transfer and cross-agency penalties, as well as the lack of incorporation of fares within the path finding method. The assignment process took 9 hours, averaging a full path search (every possible route) speed of 15 agents per second (note that while a faster method from the initial prototype was devised to make assignment feasible, further methods to improve performance were not deemed critical at this stage and left for future efforts).

### 8.5 Building the Base Simulator Network

Acquiring these two main categories of data (transit network and service structure, and agent O/D and departure times) made it now possible to build and simulate the Greater Toronto network within the platform. The system was divided up as follows:

- One street simulator simulating all surface routes for all agencies
• Two line simulators, one for the TTC subway network, and one for the GO Train network
• 69 station simulators for the TTC network and 63 station simulators for the GO network

As detailed in the prior chapter (Chapter 7), the input network structure via GTFS was used to draw all routes and guide vehicles for both the surface and sub-surface simulators, with both simulators taking on a simplified form. As an illustration, the main section of the rail portion of the network, as displayed in the platform interface, is shown in Figure 8.2, with the TTC network in red, the GO network in green, and rail stations as grey rectangles. Stations were predominantly kept to a simplified form as well, except for two key stations detailed in the following section, with a simple assumption of an agent needing to travel 100 metres between any key point (platform, doorways).

8.5.1 MassMotion Models of Bloor/Yonge and St. George Stations

One of the goals of this initial large-scale case study was to test the ability of Nexus to successfully handle mediating agents between multiple external MassMotion station simulations, preferably in locations where they are most appropriate. To that end, two key stations within the subway network were constructed within the MassMotion pedestrian simulation software, Bloor/Yonge and St. George. Their locations (shown in Figure 8.3) are at the two main transfer points between the horizontal Bloor-Danforth line, and the U-shaped Yonge-University-Spadina line; as a result, they are the two busiest stations in the Toronto transit portion of the network (source: 2013 TTC Subway Ridership). Bloor/Yonge station, in particular, is well-known to be over-capacity during peak periods, and is of key interest to the TTC with respect to better crowd management [1].
To construct both stations, layouts of each floor, the physical characteristics of all vertical transition elements (stairs and escalators), and the positioning of trains were required. Due to security concerns, digital CAD files of the stations could not be acquired from the TTC. Instead, measurements were made against provided simplified print-outs, coupled with on-site measurements for verification and for measurements not contained within the drawings (e.g. lengths and numbers of steps).

In general, construction of models in MassMotion involves three stages: building of the 3D architectural elements, assigning each element as a MassMotion actor, and setting train and agent schedules. Often architectural elements can be imported from 3D CAD files into the Autodesk Softimage MassMotion workbench, detailing the physical layout of each of the floors of the station, and any connecting elements. Otherwise they can be built using tools within Softimage.

Once constructed, elements can be assigned a range of types, with the main ones listed below. Each element requires the Softimage geometry to conform to specific characteristic (e.g type of shape, divisions)

- Floors: main walking areas of agents; edges of floors essentially act as walls
- Links: used to connect floor regions on the same level and mark the only locations where agents can move between adjacent floors (even if they are physically touching); links can be gated to simulate controlled flow (e.g. turnstiles) and train doors, specifically allowing for a priority direction to better model boarding/alighting behaviour
- Stairs and Escalators: allow for transition between floors on different levels
- Portals: mark entrance, exit and waiting locations for agents
- Barriers: fixed obstacles that force agents to navigate around them

MassMotion actors can be grouped in some specific ways that influence how agents perceive them. Groups of links can be ”banked”; this causes an agent to ignore the distance of individual links from the
next waypoint along the agent’s route. As a result, the agent will consider only how to most quickly move past the banked set of links, considering their current location and congestion in front of each link. Banking is also possible for stairs and escalators, and was used for all adjacent stair-escalator sets in the constructed models.

Finally, agent schedules are used to set when agents enter and their target (including on-the-way intermediate points if necessary), while event schedules can be used to control the opening and closing of gated links. For the purpose of Nexus, these internal mechanisms of agent and event schedules were overridden through the use of the developed API to allow for both to be set dynamically.

**Bloor/Yonge Station**

The model of Bloor/Yonge station was adapted from work performed in conjunction with a supervised undergraduate student tasked with first developing the model from a crude station schematic, and then using it to conduct analysis on how train arrival patterns would affect station flow. Details on its specific method of construction can be found in [101]. Building of the architectural elements involved tracing a simplified floor plan layout in a PDF format within AutoCAD, importing that model file into MassMotion for automatic conversion, followed by manual fixes of architectural elements. Trains were added as a series of floors (one for each car), containing a portal for the entire car, and four gated linked to represent the doors (Figure 8.4). At this stage, detail at the concourse level was minimal, with the ticket booth area and turnstiles not specifically modelled. This was not deemed to be particularly important in the absence of a higher fidelity model of the surface network (where pedestrian movement was greatly simplified); the focus in this thesis was, therefore, on proper modelling of transferring agents.

While the station is the busiest in the network with high numbers of transit users transferring, it was presumably not built with this level of demand in mind. As a result, the transfer point to the upper-level north-south Yonge-University-Spadina platform from the Bloor-Danforth platform occurs at the northern end of the north-south platform (shown in Figure 8.4). This causes issues in both passenger flow and distribution of pedestrians along the north and south platforms. To attempt to mitigate these effects, the TTC in recent years began placing temporary barriers along the southbound platform during the AM weekday period to force transferring passengers to enter near the middle rather than the end of the platform. As the model was for the morning peak-period rush, this barrier was added.

The model was calibrated against data collected during the AM peak period in July of 2011. This included flow through each vertical transition element (stairs, escalators), train arrival and departure times, and the distribution of boarding and alighting passengers for the upper-level trains. The main tasks of interest to its application in the test-case network during the calibration step were ensuring that routing penalties were appropriate to match observed route choices, and working out any modifications to behaviour or architectural elements to ensure that agent behaviour was appropriate (mainly did not result in unnatural levels of congestion).

The use of the Bloor/Yonge model within the framework was key both as being a critical station in the network, but also provided a demonstration of the ability of the platform to enable easy re-purposes of existing models for broader network analysis. To maintain fidelity, the internal calibration parameters were maintained when adapting the model for use in the network. Modifications, however, needed to be made to make it properly interface with the model. These included removal of all agent schedules, renaming platforms to work within the naming convention assumed by the platform, and the addition of portals along the length of each platform to indicate waiting locations. Data was also included on how
each external stop ID (defined following GTFS processing) mapped to the portals of the MassMotion
model, and a list of waiting portals associated with each platform.

St. George Station

The model of St. George station was constructed through floor plans provided by the TTC; as it was
possible to get these in digital format, manual measurements were made to determine dimensions. This
was combined with field measurements of the station collected in the vertical transition modelling study
detailed in Chapter 4. As measurements were performed by hand, the model architecture was built
using the geometry tools within Softimage instead of import via AutoCAD, as with the Yonge/Bloor
model. Afterwards, a similar process was followed; however, as no data collection was performed for
this station, specific calibration of model agent and routing parameters were not conducted beyond
minor modifications to ensure that unnatural agent flow did not occur (e.g. agents becoming stuck when
walking in opposite directions on the narrow platforms). Figure 8.5 presents the final model, showing
the station’s four levels, and more specifically the two train platform levels (for Bloor-Danforth at the
bottom, and for Yonge-University-Spadina above it), and single bus platform at the street level. In the
figure, portals are coloured green, links are yellow (with green if gated), floors are blue, stairs are orange
and escalators are red.

8.5.2 Operational Settings

In addition to the physical structure of the test case network, several components included variable
processes and settings that could influence network operation and agent behaviour. With this thesis
focused on the impact of operational, rather than physical, changes, these processes and settings pre-
dominantly defined the base model of the case study. Much of these operational settings for system
components were detailed in Chapter 7; this section lays out additional parameters specific to the case
study.
Platform Behaviour

The developed platform dispersion model (Chapter 5), including preferred waiting locations for agents based on their network path, was put in place for all of the TTC subway stations (upon where the data used to estimate the model was collected), with specified modification for the MassMotion models. The effect, however, of enabling this endogenous model of the distribution of passengers on a platform compared to a uniform assumption was examined (see Section 8.6). For GO Train platforms, as the service is relatively infrequent (>20 min) and run based on a schedule, an assumption was made that users would be more likely to distribute themselves evenly, independent of platform layout. As a result, uniform distribution when boarding was assumed. As mentioned in the prior chapter, dwell times were set using Weston’s formula for non-MassMotion stations. For MassMotion stations, doors were kept open until either no agents were within 2 metres of a car door. In both cases, agents were loaded up to a time limit of 90 seconds; this value was arbitrary in the base model.

Rail Operation

To keep to schedule as much as possible, all trains were held until their release time at the beginning of each trip if they completed a prior cycle ahead of time. After initial release, trains were set to move at an average speed based on the distance to the next stop and the difference in the scheduled times between their current stop and next (taken from the GTFS data), with an assumed dwell time of 15 seconds. No control was placed on trains running on the TTC to maintain these scheduled times in between terminals. On the other hand, vehicles on the GO commuter rail network were held at each station until their scheduled departure time, to conform with how GO runs their train service. Finally, in both networks, only one vehicle was permitted to occupy a platform at any given time.

The capacities of each vehicle were taken from published specs for each type of vehicle. The TTC train network has three configurations of vehicles. Trains on the main Bloor-Danforth and Yonge-University-Spadina lines consist of 6 transit units, each holding 167 individuals; this corresponded with the older Bombardier T1 subway trains, in service during the period of the source data. The Sheppard line uses the same car-size, but with only 4 transit units per train. Finally, the intermediate rapid transit, Scarborough line, have 4-car trains, each holding 54 individuals. For the GO network, as detailed
breakdown of the structure of trains on each line were not readily available, an assumption was made
that all trains were at their maximum length and capacity (12 carriage trains, with each carriage able
to hold 162 agents). While it is likely that trains do hold more individuals than their stated capacities
during rush-hour service, without such specific data available, capacities were strictly adhered to within
the simulation.

**Bus/Streetcar Operation**

The focus of this case study surrounded the impact of crowd flow in stations on subway network dynamics.
While the surface network was incorporated to feed agents into the rail system, it was not the primary
concern. As a result, as mentioned in Chapter 7, a simplified mode of vehicle movement was used, with
complete adherence to the schedules specified in the GTFS input.

Vehicles in the network were divided up into two types, buses and streetcars. While multiple agencies
were modelled, buses were assumed to be of standard TTC dimensions, holding a max of 51 agents. For
streetcars which operate solely on the TTC network, the capacity was set at 81 agents, the weighted
average capacity of the fleet of regular and articulated streetcars in the TTC system.

### 8.6 Base Model Results

Before Nexus could be used to analyze the effects of disruptions in the case study network, first, analysis
was conducted to examine the performance of the system under ‘normal’ conditions. Also of interest
was the individual impact of incorporating the pedestrian models developed in this thesis (Chapters 4
and 5) and the effect of having full MassMotion models in place of the simplified versions used for the
rest of the stations of the network.

#### 8.6.1 Visualization of Results

As detailed in Chapter 7, the Network Analyzer component of Nexus was expanded for the purpose
of this case study in order to visualize the movements of vehicles throughout the network, perform
automated processing of results and produce relevant performance graphs. While allowing for analysis,
they also played a key role in identifying modelling and programming issues that needed to be addressed
during construction of the case study network. Also included was the ability to construct comparative
graphs across several runs.

Performance graphs spanned a range of measures, from those providing a snapshot of overall network
performance and agent experience, to ones dealing with specific transit routes or stations. Network
performance measures examined included average station and platform volumes over time, while agent
experience measures included overall agent progress over the simulation period, and histograms of average
travel and waiting times. Specific routes were analyzed using histograms of headways (at specific stops),
on time performance and dwell times, as well as passenger load and boarding and alighting volume
graphs. Stations were evaluated based on platform counts over time.

#### 8.6.2 The Base Model

The base model was defined as administering transit service under normal operation (no disruption),
with all settings as detailed in the prior sections of this chapter. Specifically included were the platform
behaviour and vertical circulation choice models developed in this thesis, and MassMotion models of Bloor/Yonge and St. George Stations.

As mentioned earlier in the chapter, in the absence of a calibrated transit assignment method, particularly with fare integration, while the volume of agents were comparable to the real-world situation, the model was not expected to produce real-world passenger flows. Instead, it was used as a way to investigate the issues that could be faced when implementing a large-scale network within Nexus, and also as a starting point for an accurate full scale model of the Greater Toronto and Hamilton region, the next phase on the Nexus roadmap. Some of the more obvious discrepancies and their possible sources are detailed during presentation of results later in this section.

### 8.6.3 Characterizing the Base Model

The performance of the base model was examined at both the network and individual-route level. As the focus of this thesis was the TTC subway network, all station and platform-level analysis was confined to that portion of the network. As definite results could not be drawn on the TTC network without a calibrated transit assignment, trends were analyzed instead with the simulation run with a single random seed; moving forward with a more complete model would necessitate a more thorough analysis. The passenger volume graphs for train stations (Figures 8.6 and 8.7) and TTC subway service (Figure 8.8) showed an expected and noted increase in passenger traffic throughout the system from 8-9AM, coinciding with the highest volume section of the AM peak period. The double spikes in average platform volume seen in the figure during this period was attributed to sudden increases in agents at the northbound platforms of Union station, transferring from the connecting GO lines. This is expected to be an artifact produced by the transit assignment method, with the lack of consideration of fares resulting in an unnaturally high number of agents transferring to the TTC from GO at Union station. The agent experience graphs (Figure 8.9) show the travel and waiting time distributions for transit users who used the train, and the overall progress of agents making their way through the network. Note that some agents experienced excessively long travel times due to abnormally long waiting times at their initial stop; this was most likely due to incorrect route assignment for early commuters who should have driven to the station rather than wait for a bus.

In addition to these network-level performance measures, the platform also allows for analysis of specific routes. For illustration, graphs detailing performance of the Yonge-University-Spadina subway originating at Finch station (the busiest line of the network) are shown in Figure 8.10. These include the headway distribution at Bloor/Yonge station, the dwell time distribution of the line, and the passengers loads and boarding/alighting diagrams during the highest volume period. These graphs show a system behaving, for the most part, as expected, with large numbers of agents transferring at Bloor/Yonge station, most destinations in the downtown-core (Yonge/Bloor to St.George stations), and the line at or over capacity. However, the limitations of the current model due to the lack of fare considerations during transit assignment and crude models for Union Station and subway operation were also apparent. Excessive numbers of agents are transferring from the GO rail system to the TTC at Union Station, with the majority of these agents travelling to nearby stations (where walking would be more appropriate rather than paying the extra fare). While this played a role in the wide spread of headways seen in Figure 8.10(b) (due to excess dwell times at Union in both directions), the simple subway operations model also was a factor with an inability to compensate.

Finally, a cursory check of the flow through the two stations modelled in MassMotion compared to
TTC counts for 2011 was conducted. While, as with overall agent flow, the numbers were not expected to match without a calibrated assignment model, the values were still reasonable. Bloor/Yonge station was modelled as processing around 39 thousand agents, with TTC counts at around 44 thousand in the corresponding period; St. George station was modelled as relatively less busy, processing around 24 thousand agents compared to around 30 thousand from the TTC counts. Simulation time for the 4.5 hour simulation period (5-9:30AM) averaged around 20 minutes on the test system.
8.6.4 Impact of Incorporating the Developed Pedestrian Models

Both pedestrian behaviour models developed in the course of this thesis were incorporated within the simulation platform. While increasing the fidelity of utilized models is always a worthy endeavour, particularly for analyzing local flow of passengers at individual stations, whether such improvement results in significant impact in the predicted values of transit service and agent experience is also important to
Figure 8.10: Performance of the Yonge-University-Spadina subway line for the base model

discern. Nexus is particularly useful in these cases by enabling this broader study.

As detailed in Chapters 4 and 5, models were created for two situations, the choice of facility in transitioning between station levels (stairs or escalators), and the distribution of passengers along a platform as they wait for the train. The vertical circulation choice models were limited in application
within the case study, only being applied within the MassMotion simulator. As a result, the model was applied at less than a dozen choice locations, and all in the ascending direction (as set during the AM peak period at Yonge/Bloor and St. George stations). Nevertheless, with these two stations being the main transfer locations between the two main subway lines of the TTC, impact was observed throughout the network with an increase in platform volumes, as well as locally at the stations where the models were incorporated (Figure 8.11). Of particular interest is the divergence seen in the figures between the impact the models had at the station at which they were applied compared to the effect overall, difficult to predict without a connected system.

![Figure 8.11: Impact of vertical circulation choice model on platform crowding](image)

The incorporation of platform behavioural model was more obviously expected to impact crowding levels at stations and dwell times of trains throughout the network, given its broader application at all TTC stations. To analyze this impact, the base model results were compared against the model with boarding location assigned randomly for agents for the high volume period of 7:45-9:15 AM. Two measures were examined, the dwell time of TTC trains and average platform counts. As shown in Figure 8.12, impact was observed on both measures; this translated to average dwell time and platform counts being reduced by around 15% and 22%, respectively, for the random platform location assignment model. These results were as expected. More uniform boarding distribution would reduce the number boarding at the peak section due to more even spreading of agents, in-turn reducing dwell time. This would have the added impact of increasing line capacity to reduce overall platform counts.
8.7 Analyzing the Impact of Disruptions

To demonstrate a key piece of functionality of Nexus, this section details the steps to investigate the station and network-level impact of a common disruption in subway network. The situation investigated was the case where a train at a specific station is held for an intermediate (less than half-an-hour) length of time; often this occurs due to the pressing of a passenger assistance alarm or issues in the subsequent track section.

8.7.1 Specifying the Disruption

As mentioned in the chapters dealing with the platform frameworks and implementation, specification of disruptions are intended to involve two main steps. The first is a broad command sent from the analyzer component to the coordination server indicating the overall disruption to be initiated, while the second step translates this general instruction to the appropriate commands for the network components. In the case of a disruption that causes a train to be held at a particular platform, the following information was specified:

- Target platform
- Time and duration of disruption

As this type of disruption solely involves train movement (with no allowance for agents to re-evaluate), the signal necessary to simulate such an event only needed to be passed to the line simulator. It was implemented by extending the dwell time of the first train to reach the platform after the specified time.
of disruption by the specified duration. While the train is held at the station component, the two-way control of train dwell (either station or line) programmed into stage II of the prototype, allowed for holding of the train without the disruption command sent directly to the station component. For the same reason, the disruption could also be handled by signalling solely the station; this method might be appropriate if agents within the station are given the opportunity to find an alternate route.

For the proof-of-concept analysis conducted, the disruption was set to occur for the first train to reach the southbound platform at Bloor/Yonge station after 7:55AM. This platform was selected given its high volume and overall critical importance to the subway network. Disruption lengths were set at 10, 20 and 30 minutes to gauge the ability of the modelled subway network to recover to normal crowding levels without any change to the operation of other trains.

### 8.7.2 Disruption Results

The short-term impact of disruptions was evaluated using two measures, the average platform counts throughout the TTC subway network outside of the location of disruption (Figure 8.13(a)), and the distribution of subway-utilizing agent travel times (Figure 8.13(b)). The figures shows a model network that is able to relatively quickly recover (within 10 minutes) to normal crowding levels on the platform for short-duration disruptions (10 minutes) without any change to train operation, and with minimal effect on agent travel times.

Disruptions of longer duration (20 and 30 minutes) were not as well handled without intervention by the transit service. A disruption of 20 minutes (7:55-8:15AM) needed 25 minutes after the end of disruption to return to normal levels of platform use. At a disruption of 30 minutes, however, the network was unable to recover, maintaining elevated passenger crowding levels on the platforms throughout the remainder of the simulation period. For both of these longer disruption lengths, as expected, the model produced agent travel times that were noticeably longer, particularly for the 30 minute disruption.

Overall, the case study model acted as expected when stressed with disruption events of varying lengths. These situations were also able to be analyzed without requiring modifications to the base model files with on-the-fly commands given to the line simulator. Finally, the results show that the platform has its use in both understanding the impact of specific disruptions, but also in determining the level of stress the network could handle before intervention is required.

### 8.8 Testing Disruption Management Strategies

The final goal of the developed platform was to provide a means for testing response strategies to disruption events. As with the simulation of disruptions, this was enabled by the implementation of the communications framework (see Chapter 3) which mediates passing of control signals to cause changes in the network during run time. As a result, the mechanism used to pass the commands for network or agent response is identical to those used to simulate disruptions; what changes are the targets of these commands and the range of potential effects. While disruption scenarios generally cause a blockage or slowdown at specific stations or segments, response methods can take many forms. With respect to the effect on service, they can include the following:

- Shut down lines or parts of lines
- Re-route vehicles (where possible) or turn back trains
Figure 8.13: Impact of various duration disruptions at Bloor/Yonge station
• Introduce new routes (bus bridging)

• Modify train operation (e.g changing speed)

Also part of response is the behaviour of transit users, particularly whether they remain on their current path or look to re-route. This is determined both by their own characteristics (familiarity with the network for example) as well as by whether they have received information to be aware of any changes in service. The latter factor is an important concern of effective response measures, where insufficient or ineffective information provision of response measures to transit users can hamper efforts. The platform frameworks make allowance for such testing, with control packets that could, for example, control messages on a PA system, while virtual sensors indicate which agents received the information. This functionality was not programmed into the prototype system, but the end result, a subset of agents only considering re-routing, was achieved.

To show proof-of-concept of using Nexus to conduct on-the-fly response strategy testing, a simple response measure was devised for the longest disruption scenario analyzed in the prior section (30 min hold of a train at the southbound platform of Bloor/Yonge station). To maintain simplicity, train operation was left as is; modification to train schedules (e.g turning them back) would require creating new trip schedules for each vehicle both during and after the disruption given the trip-centric schedules used for vehicles in the final prototype. This task is not trivial, and an active dedicated area of research in its own right (see Section 2.6). As such, such manipulation was left as a possible future direction of research.

Instead, the response measure analyzed focussed on provision of information to relevant agents. The goal of the response was to avoid the over-crowding situation at the southbound platform due to a continuous influx of agents from the westbound Bloor-Danforth line. As such, agents who would be projected to make the transfer to the southbound platform at Bloor/Yonge station were made aware of the disruption and allowed to reconsider their route choice. This decision point was set as the time of arrival of their current vehicle at its next stop, with the number of transfers limited to two from the current position (alighting at the stop counting as one).

Figures 8.14 and 8.15 show the impact of the response on the relevant measures. Average platform crowding (Figure 8.14(a)) displayed a noticeable reduction with the response method; the majority of this was, however, attributable to the greatly reduced passenger flow through the southbound platform at Bloor/Yonge (Figure 8.14(b)). As the agents affected were predominantly only those transferring from the westbound direction of the Bloor-Danforth line, and no adjustments to train or bus service was made, the response strategy’s minimal impact on other platforms was expected. The method, in fact, while reducing overall platform crowding, could not return overall levels to normal during the simulation period. There was, however, a benefit on the travel and waiting times of subway users (Figure 8.15). Also of note was the faster run time (26 min vs 30 min) of the simulation, even with the re-calculation of agent routes, due to the reduced agent population in the MassMotion simulation of Bloor/Yonge station.

Overall, this final section showed the final key capability of the prototype to assist in response strategy testing for disruptions. It also displayed an ability of the system to dynamically route agents on-route in response to network changes, which has relevance beyond disruption management scenarios to evaluating more general transit ITS information provision technologies.
Figure 8.14: Impact of response strategy on platform counts
Chapter 8. Large Scale Case Study - Greater Toronto Transit Network

Figure 8.15: Impact of response strategy on agent experience
8.9 Summary

This chapter presented a proof-of-concept case study of the Toronto transit network to illustrate the capabilities of the Nexus prototype, particularly its ability to model a large-scale network, interface with a commercial pedestrian simulator, and be used to simulate disruptions and test response strategies. Using data from a variety of sources, including GTFS and the 2011/2012 TTS, a model was constructed of the Greater Toronto transit network during the AM peak period. This included full 3D models of two key transfer stations, Bloor/Yonge and St. George, simulated within MassMotion. Using this base model, an analysis of the network-level impact on transit service and platform crowding was conducted, as well as illustrating how Nexus could be used to test a common disruption response strategy. This, however, was only a first step, and significant research efforts remain, particularly in vehicle rescheduling, to fully realize the disruption management capabilities of the platform.
Chapter 9

Summary and Conclusions

9.1 Summary

This dissertation focussed on how a better understanding of passenger crowd movements in stations and their impact on mass transit could lead to more accurate modelling of transit service. To fill gaps identified in the literature and in commercially available software, this investigation was conducted with a goal of developing Nexus, a network modelling tool that could be used to analyze the impact of disruptions on transit network performance and the experience of passengers, and more broadly act as a research platform to integrate transit modelling efforts. Key to this effort was improvements in modelling of passenger behaviour at key locations in the station, particularly at the interface with train service (the platform).

This thesis was divided up into 8 chapters. It began in Chapter 1 with an overview of the motivation behind the research, the need to consider the two-way interaction between crowds and transit service and the lack of available tools to model large-scale transit networks to study operational changes. Also summarized were the key objectives of the thesis and the general approach.

Chapter 2 presented an overview of the literature in the various fields that involved within transit network modelling, including urban micro simulators, train and pedestrian modelling and transit assignment. Also discussed are current approaches used by agencies in handling disruptions in the network and research into rescheduling of trains. Highlighted are the gaps in the knowledge in each of these fields, and the overall lack of integration or holistic analysis when modelling these systems.

Chapter 3 presented the frameworks developed to address this lack of integration. First, the research framework that provided the scope and potential applications of the simulation platform is detailed. Next, the computing framework provides the specification of the tasks and inputs/outputs for each transit network component (station, rail, surface, transit assignment, data handler, and analyzer). Finally, the agent and communication frameworks are explained, with the goal of providing intelligent routing to agents and network awareness, and a means to dynamically control and receive feedback from transit service components.

With crowd behaviour in stations key to the network model, particularly at bottlenecks and interfaces, the next two chapters covered models of vertical circulation choice (Chapter 4) and passenger spread on train platforms (Chapter 5). The vertical circulation choice models were developed to predict how pedestrians would make their way between station levels at co-located stair-escalator facilities. This
was performed with the goal of better understanding the actual utilized capacity of such facilities, an issue of particular importance at transfer stations. Two classes of models (aggregate logistic and disaggregate mixed logit) were developed and estimated based on field data collected at several Toronto subway stations. The latter type was implemented within the pedestrian simulator MassMotion with parameters tuned to produce the most accurate flow splits.

The passenger distribution model detailed in Chapter 5 was required to better model the interface between stations and trains. It provides the boarding distribution of passengers, a key input in a dwell time model. The model consists of two components, a diffusion model to spread out passengers based on a density gradient, and a targeting model for passengers with preferred waiting locations. This allowed the model to provide a time-dependent distribution of passengers based on the entries of each passenger at the platform entrances, and the arrival times of trains. The model was estimated via a simulation-based genetic algorithm, and incorporated within the station simulator component of the platform.

Next, Chapters 6 and 7 explain the two stages of development of the Nexus prototype. Stage I (Chapter 6) involved creation of the basic architecture in C# with a focus on coordinating system components and establishing the general feasibility of the service-oriented approach of the computing framework. Also of interest was investigating whether distributing components over multiple computers would indeed speed up simulation even with the possible latency when communicating over a network. Initial proof-of-concept was verified against two hypothetical test networks of small and medium size. Stage II (Chapter 7) greatly expanded the system developed in the initial stage, with modifications to allow for importing of a real-world network, as well as necessary performance improvements to handle the much larger size of networks. Stage II also marked the development of an interface to allow the commercial pedestrian simulator MassMotion to act as a station simulator within the platform. To visualize the network for illustration and debugging purposes, the analyzer component was expanded in this stage to display the network on a map backdrop; an analysis section was also added to process simulation output and provide performance graphs for the entire network or specific routes.

Finally, Chapter 8 presented a large-scale case study of the Greater Toronto transit network to provide a proof-of-concept of the ability of the simulation platform to handle large real-world multi-agency networks and allow for analysis of disruption events. The model network was loaded based on the GTFS data for the available agencies in the region, and an agent population was synthesized based on the 2011/2012 TTS data for the AM peak period. MassMotion models of the two key transfer stations (Bloor/Yonge and St. George) was created, with simplified models for the remainder of the stations in the TTC subway and GO Train networks. Both of the developed pedestrian models (Chapters 5 and 6) were also incorporated. While a calibrated transit assignment process was beyond the scope of the thesis, the base model was characterized (using measures like average platform volume, dwell distribution and agent travel times) to provide a benchmark. Finally, a simple disruption scenario of varying duration was modelled using the dynamic event capabilities of Nexus to analyze how well the modelled network could handle short-term disruptions. Overall, Nexus provided logical results, and demonstrated its ability to realistically model disruptions and response strategies.
9.2 Conclusions

In the course of the construction of the platform prototype, the passenger models developed and the case study, several conclusions were drawn. While the platform frameworks were designed with a potential research roadmap in mind, and ideas of maximizing scalability and flexibility in modelling networks, moving from conceptual frameworks to a realized platform is dependent on technical feasibility. This was of particular concern in this case given the complexity of the architecture and the potentially high number of independent applications needing to communicate. Based on this exercise, the following was learned, potentially helpful for others who might consider such a system, or independently implement the frameworks presented in this thesis:

- Recent advances in programming tools and concepts were essential to its success. For the core simulation engine, these included recently released computing frameworks to handle task parallelization and asynchronous operation, and web-service architectures that allow for independent specification of service interfaces and data contracts, persistent objects supporting concurrent calls and fast serialization and deserialization of data. For visualization, also important were user interface tool developments that allow for map data to be easily displayed and objects to be overlaid using real-world coordinates.

- Text file inputs and outputs have been a common approach in existing commercial transportation network simulators. Nexus instead heavily utilized databases that allow for information to be structured and queries to be made for quick filtering and calculation. This was important both for post processing of the millions of lines of result output produced for each simulation run, and for debugging given the multitude of actors and applications.

- The hub-spoke topology (a central coordination engine handling all data transfer between components) was critical in hiding the complexity of the system from individual components. This allowed individual components to not have to worry about the location of neighbouring components, letting the coordination engine take care of transferring data to the right place. This is also important from a security and robustness standpoint by minimizing the number of connections throughout the system.

- The numerous connections and the significant quantity of transmitted data did not prevent the system from simulating in a reasonable amount of time. Increasing the cycle period of the synchronization pulse also helped to a point; however, as pedestrians transferring at station doorways to and from the street are transferred only on-the-cycle, this increase in speed had to be balanced by the potential modelling errors that could occur. A limit was also found after which the in-cycle vehicle transfers slowed down the progress, negating speed improvements from less frequent signalling.

- While the main reason for separation of network components into individual applications was to allow for external simulation software to be incorporated, the added benefit of being able to run components on different computers was also seen as a potential benefit. While an improvement in speed was observed in the initial Stage I networks, at the time Stage II and the case study were conducted, identical computers were not readily available (given the additional computing requirements necessary to hold the case study network). As a result, only a slight improvement
(less than 10%) was found in moving the less-busy MassMotion station to a secondary computer; increases in overall simulation speed during high agent volume periods were somewhat offset by reduced speeds during lower volume time periods, with the slower computer acting as a hindrance. Care, therefore, must be taken and additional research must be conducted to better understand how to properly load balance simulations for maximum effect. It is expected that this would be less of a concern as more detailed models are added.

Next, two main classes (aggregate and disaggregate) of vertical circulation choice models were developed in this thesis to predict how passengers would utilize adjacent stair-escalator facilities, a key bottleneck in subway stations, particularly in transfer stations. An intermediary study was also conducted to examine the relative importance of static versus dynamic factors in individual passenger choice. The following conclusions were drawn from this set of studies:

• For all classes, separate models were developed for the ascending and descending direction, as well as segmentation of the population or flow situations into those with and without persons with restricted mobility (PRM). This was done on the expectation that sensitivity to all parameters would vary based on these two factors.

• For the aggregate models, a set of logistic regression equations were developed to predict the 10-sec flow splits through the facility. The pseudo-utility functions for these models included variables for total incoming flow, opposing flow on the staircase, approach direction, height and the % of situations with PRM individuals present. Situations were divided into those with and without PRM individuals, on the assumption that their presence would alter overall behaviour.

• In all aggregate models, increased incoming channel utilization increased the use of the staircase; opposing flow on the staircase and the flow approaching from the escalator side were found to increase escalator use. A strong preference for the escalator was found in the ascending direction, with a lack of sensitivity to facility height, while in the descending direction, this preference was strongly dependent on height. Lastly, for models involving the presence of restricted mobility individuals, an increasing composition of these individuals strongly increased the affinity of the group to use the escalator.

• The intermediate study used a latent-class disaggregate discrete choice model to examine the relative weight of dynamic versus static variables on passenger choice of facility, by dividing up the choice structure into two components, one dealing with static variables (base preference, physical characteristics, overall flow) and a second solely for dynamic variables (queuing level, existing numbers of pedestrians on each facility). The static component of the model was found to dominate (73% to 27%) the overall decision, but the model did not have significantly improved predictive ability compared to the base model, indicating that a simpler structure would suffice for the final disaggregate model.

• The final disaggregate set of the models used a combination of mixed-logit (for non PRM individuals) and binary-logit (for PRM individuals) to describe choice. Variables chosen were mainly dynamic in nature (numbers of other individuals already on the stairs and escalators, total queuing numbers, opposing density on the stairs) with approach direction and height (providing the base preference) rounding out the set. Moderate performance (mid-high 70s in the ascending direction
and low-mid 60s in the descending) was found in the ability of the models to predict individual choice. When incorporated within MassMotion to interact with its walker model, the ability to predict aggregate 10-sec flows was quite promising, with nearing 90% accuracy for both directions. The parameters of application (timing and frequency of choice) played a significant role, improving accuracy by 10% when properly tuned.

The platform dispersion model acted as a core piece of research in this thesis, dealing with an area of research where a significant gap in knowledge was identified: how to predict the boarding distributions of agents, as an input into a dwell time model. It was informed by data collected at platforms across several stations in Toronto. The following conclusions were drawn:

- The framework developed to describe platform behaviour during peak periods was a key takeaway of the study. It reformulated the phenomena, in contrast to existing literature, as being dependent not only on the layout of the modelled platform, but also affected by the choices made by passengers in response to the layout of their destination platform.

- The addition of the targeting component to allow for preferred waiting locations for passengers improved model fit during peak periods relative to a diffusion-only model with passengers spreading solely from their entrance location.

- Off-peak period data was better described with the diffusion-only model compared to peak-period data, supporting the hypothesis that non-commuter passengers would be less aware of their destination and thus likely to have preferred waiting locations; however, a higher diffusion rate was found for the off-peak model as it compensated for those passengers who were targeting.

- Allowed to vary in value from each other, the percentage of the population targeting locations on the platform exhibited significant variance between stations. This in-part was attributed to the positioning of entrances and volume of flows, as well as uncertainty in the source data. However, additional study is required to more solidly rationalize the reason for the differences.

- Entrance location had little influence on agents with preferred waiting locations.

- Improved accuracy in prediction was found as the number of agents increased, an important result for application where higher volume conditions are more critical.

- The model, on average, under-predicted the numbers of people in the highest load section, the limiting section for dwell times, by -6%.

- Based on these results, the framework was incorporated into the simulation platform.

The case study of the Greater Toronto area transit network was performed mainly to illustrate the capability of the simulation platform to handle a large-scale real-world network. However, some conclusions could still be drawn both in the modelling of the network and in the results from the disruption managements studies performed:

- The platform was able to handle importing the multiple agencies of the region and allow for agents to transfer between them; however, without fares explicitly modelled, some modes, particularly the GO train network, were over-used with unrealistic transfers occurring between agencies.
• While GTFS is a common specification, it is not a hard standard and several agencies do not put out particularly complete data sets, and often differ from other agencies in their format. As a result, individual converters need to be written, and occasionally gaps in the data need to be filled.

• The path finding of agents through the network is the limiting factor for computational speed rather than the simulation run time. This is not of significant concern where transit assignment is static, but more research and development need to occur to address this issue for more dynamic applications.

• The base model of the regional networks produced TTC platform count profiles that corresponded well with the known surges in demand during the AM peak period; volumes at Bloor/Yonge and St. George stations (key transfer hubs) even without a calibrated transit assignment were also reasonable compared to actual volumes.

• Incorporation of both the vertical circulation choice and platform models showed noticeable and logical impact on platform volumes throughout the network and train dwell times.

• When stressed by the holding of a train at the key southbound platform of Bloor/Yonge station, the modelled subway network was able to quickly (within 10 minutes) return to normal crowding levels on platforms for small 10 minute disruptions, but took almost 30 minutes for a 20 min disruption, and could not recover for a 30 minute disruption. This showcased one use of the platform in analyzing the potential impact of disruption and when intervention might be required.

9.3 Key Contributions

The research presented in this thesis provided some significant contributions in several key areas; they are highlighted in this section. Generally, the dissertation dealt with enabling large scale simulation of transit networks and crowd flows, and the modelling of passenger movements at vertical transition facilities and at train platform in subway stations. Lastly, the foundation was laid for a comprehensive model of the Greater Toronto transit network to enable large scale studies in transit operation and planning.

The first contribution was the development of the Nexus frameworks to enable not only the system-level analysis conducted in this thesis but to introduce a fundamentally different method compared to existing simulation packages of constructing models of transit networks.

Currently, transportation microsimulation software attempt to simulate all aspects of the network. Nexus, on the other hand, calls for constructing the transportation network by linking dedicated specialized simulators for stations, rail and surface vehicles via their individual APIs. This permits the modeller to choose the best software for each aspect of the transit system without having to be confined to inaccuracies in software that attempt to model all components of the network. The inherently parallel and services-oriented approach utilized in this framework also enables individual components to be run across multiple computers to harness additional computing power. Agent and communication frameworks provide coherence to this network of simulators. The agent framework separates out the decision making and network-level intelligence of agents away from individual components. This permits transit assignment to be deeply integrated and dynamic throughout the simulation, allowing agents to respond to changes at any point in the network. The communication framework acts to simplify collection of
relevant data from each system component, and to direct components to make dynamic changes during simulation for unexpected events and in-run service changes.

The second contribution was the implementation of these frameworks into an operational simulation platform prototype. This involved formation of the computing architecture to produce a system that could be run in a variety of configurations, from a standalone self-contained application to multiple applications spread across a network of computers. A hybrid discrete-event, time-step coordination engine was developed to synchronize the various simulators in the network, connected via a common set of interfaces and data structures. The feasibility of integrating external commercial simulation software was demonstrated by interfacing MassMotion with the platform to act as a station simulator where required. Finally, the implementation demonstrated a method of using public GTFS data to construct complete multi-agency transit networks, automatically classifying stations, platforms and stops, logical transfer groups, vehicles and their associate trips.

The third contribution was the development of a set of models dealing with the choice made by pedestrians transitioning between floors at adjacent stair-escalator facilities in subway stations. Separate aggregate and disaggregate models were developed for each direction and the two major classes of pedestrians, those with and without mobility issues. The data collection conducted for estimating both of these models was the largest of any prior effort, with pedestrians observed across facilities of widely varying physical characteristics at several stations in Toronto. In addition, the variables selected, unlike prior studies, were those that could be easily applied by transit practitioners, in the case of the aggregate models, and within pedestrian microsimulation for the disaggregate models. The latter of these models were incorporated within MassMotion, which, in combination with the developed simulation platform, permitted analysis of the potential impact beyond the boundaries of individual stations based on changes to modelling techniques at pedestrian bottlenecks in key stations.

The fourth contribution was the development of a framework to more accurately model the behaviour of commuting passengers on train platforms, as well as a novel time-dependent diffusion-inspired model of the spread of these passengers. This was the first model to consider both the relative timing of the entrance of passengers into the platform relative to train arrivals, as well as the first to incorporate the destination of individual passengers when considering where they might prefer to wait. Modelling of this behaviour is critical in determining how they are spread at the time of train arrival, thus playing a key role in estimating train dwell times. It also allows for the boarding distribution to be endogenous to the model rather than having to be set externally as is the case in current transportation simulators. Finally, as with the vertical circulation choice model, the framework and diffusion model were implemented within the simulation platform. They were found to have a strong impact on platform counts and train dwell times throughout the network, with an expected and noticeable increase relative to the uniform-boarding assumption often used in models.

The final contribution was an initial model of the Greater Toronto transit network, and a proof-of-concept analysis of the impact of disruptions and a demonstration of using the model to evaluate response methods. A key feature of the developed system is the ability to develop large-scale regional models using a piece-meal approach. This initial model, therefore, lays the foundations for a more comprehensive regional model of the Greater Toronto Area, by incorporating all agencies in the area who release public GTFS data, including the GO train network, with stand-in models of the surface vehicle and subway networks, and the majority of the stations developed available data. As a starting point towards a full network model, two key transfer stations (Yonge/Bloor and St. George) had 3D
models built, the former adapted from a model built for a different study. These two stations allowed for explicit representation of the movement of large crowds between transit lines to gauge the impact on service from local changes. Finally, the model showcased the ability of the platform to investigate the impact of disruption events and test strategies to more quickly return service to normal. This is, as far as is known, the first such operational investigation of the Toronto transit network.

9.4 Future Research Directions

While initially focussed on the problem of disruption management of transit networks, the framework and implemented platform prototype presented in this thesis has significant potential for a wide variety of uses in transit network planning and operational management. To conclude this thesis, the following sections detail some potential future research and development avenues that could be explored, and areas of application for the platform.

9.4.1 Improvements to Component Simulators

Improving Surface Transit Simulation

The software used for simulating surface transit is heavily dependent on the project use case. This could range from full multimodal simulation of all transportation modes to only simulating transit vehicles but with more realistic movement, accounting for the effects of signals and other traffic. For full multimodal simulation, commercial software with potential are Aimsun, with its ability to run at different resolutions of detail (macro, meso, micro), or MATSIM with its queuing based traffic model. Full microsimulators like VISSIM and PARAMICS are also viable, however given their speed, would be most useful for non-time-critical studies. In all cases, a thorough examination of the APIs exposed would be necessary to determine which software could be feasibly connected. Also, the framework specification may need to be expanded if multimodal assignment is to occur within the current transit assignment module of the framework. If full transportation simulation is not required, one avenue of research and development could be using publicly available GPS data of bus movements to produce a statistical model of travel times for vehicles along each segment of their trip, rather than relying on posted GTFS schedules as used in the prototype.

Improving Rail Transit Simulation

As detailed in the literature review section of this thesis, rail simulators handle all aspects of train movement, signal control systems, and crew scheduling, and have more complete models of train tracks. As rail simulation is the weakest link within current transportation simulation packages, incorporation of a commercial rail simulator like OpenTrack or RailSIM represents the greatest opportunity for the platform. As with the surface simulator, this would also necessitate investigation of the APIs or formation of a partnership with the makers of the software to allow for the degree of control necessary to interface the software with the platform.

Improved Station Simulation

This prototype interfaced MassMotion to act as its detailed station simulator for key stations in the network. While its possible to build all stations within the software, computing requirements would be
large, and the passenger volumes through most stations may not warrant the effort needed to construct
the models. As a result, there is need for an alternate station simulator for cases where full 3D simulation
is not necessary, to replace the extremely crude model used within the prototype. Ideally, this software
would use a queuing network model approach, while taking into account congestion along passageways.
As part of this effort, research is also needed to better understand when such a model would be sufficient,
and when a complete model would be necessary for more accuracy.

Incorporation of MILATRAS

The framework and prototype were developed with the assumption that it would be a base platform for
MILATRAS, a dynamic transit assignment method previously developed at the University of Toronto.
As the assignment method was not integrated within the scope of this thesis, it remains a key step in
realizing the full potential of the platform. Part of this effort should include expanding the method used
within MILATRAS to take into account agent level of service experienced within stations.

9.4.2 Transit User Behaviour

Routing During Disruptions

MILATRAS was developed for transit user behaviour under 'normal conditions', as is the case with all
current transit assignment methods. There, however, remains a significant gap in existing literature
on better understanding how transit users make decisions during disrupted situations, both in regular
non-panic situations (train breakdown, signal problems) and emergency situations.

Expansion of Platform Passenger Dispersion Model

The platform dispersion model developed in this thesis presented a new way of modelling the spread
of passengers along train platforms, incorporating the tendency of commuters to have preferred waiting
locations. As a first step, the model considered motion in one direction, and had individual targeting
probabilities for each station. Further research paths remain on modelling the platform in two dimensions
for irregular geometries, better accounting for input and output flows at entrances and exits, and better
understanding of how preferred waiting locations develop. Also of interest is examining the model in
the context of centre platforms.

Expansion of Vertical Circulation Choice Model

The vertical circulation choice model developed in this thesis was the most comprehensive and simulation-
focussed model to date. While a broad range of facilities was observed, and a more expansive variable set
was used, locations were all from within the Toronto subway network. How the model would, therefore,
translate to other locations is a question that could be explored with further study. Also of interest is
examining the broader routing of agents within a station, and whether findings from this model could
be extended to understand longer range path finding.

9.4.3 Transit Service Rescheduling During Disruptions

As detailed in the final section of the Toronto case study, as well as in the literature review (Section
2.6), rapid rescheduling of vehicles during disruption is an active area of research in the disruption
management field. Rapid rescheduling is required for any disruption response measure, both during and afterwards, and involves either reassigning trips to different vehicles or creation of new trips in situations where vehicle routes are affected by blockages of route segments or stops. Such capability would need to be either incorporated (if existing methods are sufficient), or could be an area of research to be explored.

9.4.4 Validation of Greater Toronto transit network

This thesis used the Toronto regional transit network as a case study to test the system. However, without the improvements to the modelling capabilities of the system and incorporation of proper transit assignment, the network could not be validated. This remains an important step to take once these changes have been made. Given the scope of such an effort, validation should most likely occur at an aggregate level (route loads, relative station volumes) to look at overall network results. Access to now-being-installed card reader data would make such an effort easier to accomplish.

9.4.5 Applications

Disruption Management Support

A primary goal of the presented framework is to provide a toolkit for transit researchers and agencies to easily test structural changes and operational responses for better preparation in the face of disruptions. Such strategies could include operational modifications, but also involve more long-term changes, such as better positioning of excess capacity and response crews for quicker disruption resolution. While ideally the system would run at sufficient speed to allow analysis to occur on-site, it is more likely that such support would involve the use of disruption response libraries.

The development of disruption response libraries is an offline process where response plans are developed based on various disruption scenarios on a simulated network. This would involve simulating a disruption on a model of the specified network, detecting the presence of the disruption, and implementing various scheduling changes or rerouting to both allow for better functioning during the disruption and expedite service recovery. There are several ways one could envision the use of this framework in disruption management preparation, ranging from an assistive tool to test and optimize plans developed manually by operators to various methods of optimization (e.g. linear programming, genetic algorithms, etc.) which use the simulation suite as a service to provide either more accurate service times, or for optimizing plans while also considering the passengers point of view and comfort.

Platform for Transit Research

Finally, while the focus on this thesis was on integrated crowd dynamics into transit system modelling for the purpose of disruption management, there was a secondary objective of constructing the toolkit in such a way to have much broader potential application. The flexibility gained from dividing up all network components would allow researchers to individually replace modules to examine, for example, the performance of alternate decision-making methods of transit assignment or the impact of more detailed train simulation. Two additional examples of using the platform to examine network-level impact were provided in this thesis by incorporating both the train platform behavioural and vertical circulation choice models. The service-oriented architecture of the entire system also allows for easier linkage with external models in the future, including multi-modal transport choice and land-use models.
Bibliography


