MODELS AND SOLUTION ALGORITHMS FOR
TRANSIT AND INTERMODAL PASSENGER ASSIGNMENT
(DEVELOPMENT OF FAST-TRIPS MODEL)

By
ALIREZA KHANI

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF CIVIL ENGINEERING AND ENGINEERING MECHANICS

In Partial Fulfillment of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

In the Graduate College
THE UNIVERSITY OF ARIZONA

2013
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Alireza Khani entitled Models and Solution Algorithms for Transit and Intermodal Passenger Assignment (Development of FAST-TrIPs Model) and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

_________________________________________________________ Date: July 19, 2013
Mark D. Hickman

_________________________________________________________ Date: July 19, 2013
Yi-Chang Chiu

_________________________________________________________ Date: July 19, 2013
K. Larry Head

_________________________________________________________ Date: July 19, 2013
Ram Pendyala

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

_________________________________________________________ Date: August 07, 2013
Dissertation Director: Mark D. Hickman
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of the requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that an accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Alireza Khani
ACKNOWLEDGEMENTS

My appreciation is given to those who supported me in making this achievement possible.

My advisor, Professor Mark Hickman, would be the first person I would like to thank for his invaluable guidance, encouragement and support during my studies. I would like to express my sincere respect for his supervision, which provided me with all the means to focus on my research.

I would like to thank Professor Yi-Chang Chiu for his guidance and insightful feedbacks in my research. My appreciation extends to Professor Kenneth Larry Head, Professor Ram Pendyala, and Professor Wei-Hau Lin for their instructive feedbacks to my dissertation. I am also grateful to my colleagues at the University of Arizona Transit Research Unit (UATRU) for all the discussions and brainstorming.

I am so indebted to my parents to whom I owe a lot. They did their best for our success and I could have never gotten to this point in my life had it not been for their love and support. I am also grateful to my brother and two sisters for their support and encouragement. I consider myself privileged for having the company of my friends in Tucson, Arizona.

Finally, I would like to express my deepest appreciation with love to my wife for her patience, emotional support, encouragement, and endless love. She rightly deserves the best in her life.
DEDICATION

To my beloved wife Zahra

And to my parents, dearest in my heart
TABLE OF CONTENTS

LIST OF FIGURES .................................................................................................................. 8
LIST OF TABLES ....................................................................................................................... 11
ABSTRACT ................................................................................................................................. 12

CHAPTER 1: INTRODUCTION ............................................................................................... 14

CHAPTER 2: BACKGROUND AND PROBLEM STATEMENT .............................................. 18
  2.1. Preliminary Literature Review ..................................................................................... 18
  2.2. Research Goal and Objectives .................................................................................... 21
  2.3. Fundamental Assumptions ......................................................................................... 23

CHAPTER 3: TRANSIT ASSIGNMENT WITH CAPACITY CONSTRAINTS AND BOARDING
  PRIORITY ................................................................................................................................. 25
  3.1. Review of Capacity Constrained and Stochastic Transit Assignment Models
       ........................................................................................................................................... 25
  3.2. Capacitated Transit Assignment with Passenger Boarding Priority ......................... 27
  3.3. Extension to Stochastic Transit Assignment with Logit Route Choice ................. 35
  3.4. Dynamic Multimodal Assignment Model ................................................................ 37

CHAPTER 4: PATH ALGORITHMS IN SCHEDULE-BASED TRANSIT NETWORKS .......... 42
  4.1. Schedule-based Transit Path Algorithms in the Literature .................................... 42
  4.2. Hierarchical Trip-Based Transit Network ................................................................. 44
  4.2.1. GTFS Data .............................................................................................................. 44
  4.2.2. Transit Network Hierarchy .................................................................................... 46
  4.2.3. Trip-Based Network Structure ............................................................................ 48
  4.2.4. Notations Used in the Path Algorithms ............................................................... 49
  4.3. Trip-based Shortest Path (TBSP) Algorithm ............................................................ 51
  4.4. Trip-based Hyperpath (TBHP) .................................................................................. 54
  4.5. Trip-based A* (TBA*) .............................................................................................. 57
  4.6. Numerical Tests of the Path Algorithms .................................................................. 62
  4.6.1. An Illustrative Example ....................................................................................... 62
TABLE OF CONTENTS - CONTINUED

4.6.2. A Real Case Test ........................................................................................................... 66

CHAPTER 5: SIMULATION-BASED ASSIGNMENT MODEL FOR CONGESTED TRANSIT NETWORKS .................................................................................................................... 73

5.1. Transit Assignment Algorithms in the Literature ......................................................... 73
5.2. The Simulation-based Transit Assignment Model ......................................................... 75
5.3. Numerical Test in San Francisco MUNI Transit Network ........................................... 83
5.4. Application in San Francisco Integrated Dynamic Travel Model ......................... 94
5.4.1. Input Data .................................................................................................................. 94
5.4.2. Integration with the DTA and Activity-based Model ............................................. 98
5.4.3. Results and Validation .............................................................................................. 104

CHAPTER 6: MODELING INTERMODAL TOURS IN A DYNAMIC MULTIMODAL NETWORK ................................................................................................................................. 118

6.1. Intermodal Assignment Models in the Literature ...................................................... 118
6.2. Intermodal Shortest Path Algorithm .......................................................................... 121
6.2.1. Modeling the Intermodal Transportation Network at Park-and-Rides 121
6.2.2. Intermodal Optimal Path Algorithm ....................................................................... 124
6.3. Modeling Park-n-Ride Choice in the Intermodal Tours ............................................ 130
6.4. Dynamic Simulation of Transit and Intermodal Tours ............................................ 135
6.5. Application in the Integrated Travel Model in Sacramento Regional Network ............................................................................................................................... 137

CHAPTER 7: CONCLUDING REMARKS .............................................................................. 151

REFERENCES ...................................................................................................................... 157
LIST OF FIGURES

Figure 1 The model development process .............................................................. 16

Figure 2 Sample trip-based transit network with 4 stops and 5 vehicle trips in a transit route. Letters on each link represents the vehicle trip ID .......................... 29

Figure 3 A sample multimodal network .................................................................. 38

Figure 4 GTFS files. Attributes with common symbol and/or color are used for linking two files ........................................................................................................ 45

Figure 5 Transfer stops in a typical transit network ................................................. 47

Figure 6 The TBSP algorithm .................................................................................. 52

Figure 7 Attractive set of transit paths in a hyperpath .............................................. 55

Figure 8 The TBHP algorithm .................................................................................. 57

Figure 9 The TBA* algorithm .................................................................................. 60

Figure 10 The sample network .................................................................................. 62

Figure 11 Computational test of the trip-based path algorithms in Sacramento network ........................................................................................................ 67

Figure 12 Distribution of the computational time for shortest path algorithms over random set of origin-destination-times ........................................... 70

Figure 13 Distribution of the computational time for hyperpath algorithms over random set of origin-destination-times ............................................. 71

Figure 14 Distribution of the computational time for A* algorithms over random set of origin-destination-times ....................................................... 72

Figure 15 Transit assignment gap by iteration number in San Francisco MUNI network ........................................................................................................ 86

Figure 16 Transit assignment gap by time in San Francisco MUNI network ...... 87

Figure 17 Average passenger travel cost by iteration in San Francisco MUNI network ........................................................................................................ 87

Figure 18 Minimum OD travel cost in the final solution of the proposed model vs. minimum OD travel cost in the one-shot stochastic assignment with no capacity constraints. The differences show the effect of capacity constraints with boarding priority .......................................................... 92
LIST OF FIGURES - Continued

Figure 19 Average vs. minimum OD travel cost in the one-shot stochastic assignment without capacity constraint. The differences show the effect of stochastic assignment ................................................................. 92

Figure 20 Average OD travel cost in the final solution of the proposed model vs. minimum OD travel cost in the stochastic assignment without capacity constraint. The differences show the effect of stochastic assignment and capacity constraints with boarding prior ................................................................. 93

Figure 21 Average travel cost in OD pairs significantly affected by stochasticity and capacity constraints, showing different parts of the travel cost ......................................................... 93

Figure 22 Distribution of passenger boardings and alightings based on APC data ........................................ 97

Figure 23 The integrated travel model framework in San Francisco, CA ........................................ 99

Figure 24 Capacity violation reductions in the iterative transit assignment ........................................ 108

Figure 25 Dwell time gap convergence in the multimodal assignment ........................................ 109

Figure 26 Distribution of the transit trips with a) number of transfers b) travel time ................................................................. 110

Figure 27 Transit vehicles’ travel time in a) APC vs GTFS b) DTA model (2nd big iteration) vs GTFS c) DTA model (2nd big iteration) vs DTA model (1st big iteration) ................................................................. 113

Figure 28 Sample load profile in inbound vehicle trips of route 38 – Gearya) Westbound vehicle departing at 4:37PM b) Westbound vehicle departing at 5:42PM ................................................................. 114

Figure 29 Average number of on-board passengers in different routes ........................................ 116

Figure 30 Average ridership (number of boardings) in different routes ........................................ 117

Figure 31 Average of transit vehicle’s total dwell times in different routes ........................................ 117

Figure 32 Framework of the intermodal network in park-and-ride facilities ........................................ 117

Figure 33 Intermodal network, including different layers of transportation modes, connection between nodes and transit stops, at Sunrise Park-and-Ride, Rancho Cordova, CA ........................................................................................................ 121

Figure 34 A set of intermodal trips to a destination through park-and-ride facilities ................................................................. 124

Figure 35 The structure of the intermodal optimal path algorithm ........................................ 125
LIST OF FIGURES - Continued

Figure 36 Intermodal shortest path algorithm................................................................. 128
Figure 37 A typical intermodal tour with possible paths through park-and-rides 133
Figure 38 High level structure of the dynamic multimodal network model for
Sacramento, CA .................................................................................................................. 139
Figure 39 Integrated model data flow................................................................................ 140
Figure 40 Sacramento regional multimodal transportation network......................... 142
Figure 41 Transit vehicles travel times .............................................................................. 145
Figure 42 Average number of on-board passengers in the LRT lines a) 507-Gold,
inbound b) 507-Gold, outbound c) 533-Blue, inbound d) 533-Blue, outbound ...... 149
Figure 43 Average number of on-board passengers in the LRT Blue line a) inbound
b) outbound ...................................................................................................................... 150
LIST OF TABLES

Table 1 GTFS Required Files ........................................................................................................... 45
Table 2 Notation of the Variables Used in the Algorithm .............................................................. 50
Table 3 The Schedule of the Network Shown in Figure 10 ............................................................. 63
Table 4 The Result of the TBHP Algorithm in the Sample Network ............................................. 65
Table 5 San Francisco MUNI Transit Network and Demand in the 3-hr PM peak ....................... 84
Table 6 Transit Assignment Results for San Francisco MUNI Network ................................. 86
Table 7 Transit Assignment Results with Simulated Travel Times for Transit Vehicles .............. 108
Table 8 Sacramento transit network properties ................................................................................. 142
Table 9 Transit Vehicle Simulation Results in the Sacramento Case study .............................. 144
Table 10 Transit Passenger Simulation Results in the Sacramento Case Study ....................... 144
Table 11 Intermodal Passenger MOEs in the Sacramento Case Study ..................................... 145
ABSTRACT

In this study, a comprehensive set of transit, intermodal and multimodal assignment models (FAST-TrIPs\textsuperscript{1}) is developed for transportation planning and operations purposes. The core part of the models is a schedule-based transit assignment with capacity constraint and boarding priority. The problem is defined to model the system performance dynamically by taking into account the scheduled transit service and to model user behavior more realistically by taking into account capacity of transit vehicles and boarding priority for passengers arriving early to a stop or a transfer point. An optimization model is proposed for both deterministic and stochastic models, which includes a penalty term in the objective function to model the boarding priority constraint. The stochastic model is proposed based on logit route choice with flexibility on the level of stochasticity in route choice. Optimality conditions show that the models are consistent with network equilibrium and user behavior. An extension of the optimization models is proposed for multimodal assignment problem, in which the transit and auto networks interact dynamically.

To solve the proposed models, since the penalty term is non-linear and makes the model an asymmetric nonlinear optimization model with side constraints, a simulation-based approach is developed. The solution method incorporates the path assignment models and a mesoscopic transit passenger simulation in conjunction with Dynamic Traffic Assignment (DTA) models. The simulation model can capture detailed activities of transit passengers and determines the nonlinear penalty explicitly by reporting

\textsuperscript{1} FAST-TrIPs: Flexible Assignment and Simulation Tool for Transit and Intermodal Passengers
passengers’ failure to board experience. Therefore, the main problem can be solved iteratively, by solving a relaxed problem and simulating the transit network in each iteration, until the convergence criterion is met. The relaxed problem is a path generation model and can be either a shortest/least-cost path or a logit-based hyperpath in the schedule-based transit network. An efficient set of path models are developed using Google’s General Transit Feed Specification (GTFS) files, taking into account the transit network hierarchy for computational efficiency of the model.

A multimodal assignment model is also developed by integration of the proposed transit assignment model with DTA models. The model is based on simulation and is able to capture the effect of transit and auto mode on each other through an iterative solution method and feedback loop from the transit assignment model to the DTA models. In the multimodal assignment model, more realistic transit vehicle trajectories are generated in the DTA models and are used for assigning transit passengers to transit vehicles. In an application of the multimodal assignment, intermodal tours are modeled considering the timing of auto trips and transit connections, the schedule-based transit network, and the constraint on park-n-ride choice in a tour. The model can simulate the transit, auto, and intermodal tours together with high resolution and realistic user behavior.
CHAPTER 1: INTRODUCTION

In planning, design, operations and management of transportation infrastructures, the primary goal is to estimate how people use the infrastructures. From an economic perspective, this is important because consumers’ use of transportation supply determines the state of the demand-supply interaction in the system and the users’ benefits and costs. The result is what planners, engineers, and operators use to make infrastructures as efficient as possible. One such problem in transportation planning is called traffic assignment, in which transportation demand is assigned to the road network with specific assumptions and constraints. While the traditional traffic assignment is defined for road networks and considers passenger cars as demand, its extension to transit networks is called transit assignment, dealing with assigning passengers to transit routes or transit vehicles, given their operational characteristics. A more general assignment problem may consider all types of transportation demand and all modes of transportation, resulting in a multimodal assignment problem.

The problem studied in this research is a generalization of multimodal traffic assignment with the emphasis on transit assignment. Moreover, intermodal passengers (drive-to-transit mode) are thoroughly considered in the model. However, taking into account the interaction of the transit system with other modes of transportation is a key feature of the model. More specifically, a schedule-based transit assignment model with consideration of transit vehicle capacity and passenger priority in boarding is proposed, which is capable of modeling passengers with either walk or drive access to transit. Furthermore, an interface is designed between the proposed model and existing dynamic
traffic assignment (DTA) models to capture the effect of road traffic congestion on transit system performance and the effect of transit operation on the traffic level of service. The integration of the proposed transit assignment model with DTA models will result in a *dynamic multimodal assignment* model.

To explain the research effort in this study, Figure 1 shows the model development process. In the flowchart, the first part is the analytical model for the transit assignment problem with the stated properties, and then the multimodal assignment problem, both shown in green boxes. The next part is the algorithms for transit path generation, transit simulation, and intermodal path generation, shown in the yellow boxes. The three algorithms form the cornerstones of the solution method for the proposed optimization models. The simulation-based transit assignment model and the integrated multimodal assignment model are shown in the red boxes, as the third part of this study. In essence, these two models provide the solution algorithms for the analytical models defined in the first part and are the main achievements of this study. The final part is the application of the models in real case studies, including the San Francisco County network for the capacitated transit assignment and its integration with DTA, and the Sacramento regional network for the integrated multimodal travel model. The two applications are shown in the blue boxes in the flowchart.
This dissertation is divided into seven chapters. After this introduction, in chapter 2, the background, objectives, and a brief literature review are provided. The detailed review of the relevant studies in the literature is provided in each chapter as it relates to the context of the chapter. In chapter 3, analytic models are proposed for the schedule-based transit assignment problem and its application in the dynamic multimodal assignment problem, both in the form of optimization problems. Chapter 4 is dedicated to
a set of computationally efficient path algorithms on schedule-based transit networks using the hierarchical trip-based network format. The simulation-based transit assignment model is proposed in chapter 5, as a solution method for the optimization problems defined in chapter 3. The algorithm takes advantage of the path algorithms and of a mesoscopic transit simulation model to solve the assignment problem in congested transit networks. The integration of the assignment model with a DTA model is also considered and the results are presented as a case study using San Francisco County. In chapter 6, a dynamic intermodal assignment model is proposed including an intermodal path algorithm, a park-n-ride assignment for intermodal tours, and the dynamic simulation of intermodal tours. A case study in Sacramento is presented in this chapter. Finally, concluding remarks and possible future works are provided in chapter 7.
CHAPTER 2: BACKGROUND AND PROBLEM STATEMENT

2.1. Preliminary Literature Review

Transit assignment models started in the 1960’s when long-term travel demand models (4-step models) were introduced. The early models were mostly route choice models, capturing the out-of-vehicle travel time as an approximation of the route configuration, and considering the effect of common lines in the transit network (Dial 1967, le Clercq 1972, Andreasson 1976, Jansson and Ridderstolpe 1992). Later, a new variation of route choice models was introduced by Chriqui and Robillard (1975) that considered a set of attractive routes for the users, so they could choose a route among this set while minimizing their overall travel time. This route choice behavior was called a “strategy” in the later study by Spiess and Florian (1989) and a “hyperpath” by Nguyen and Pallotino (1988) and de Cea and Fernandez (1989). The strategy or hyperpath approach was a significant improvement in transit modeling, although this method only dealt with the frequency-based transit systems initially.

Another aspect of transit networks which makes their modeling complicated is crowding and capacity constraints. This motivated researchers to propose a second type of models, incorporating the capacity constraints or crowding in the assignment model (Gendreau 1984). This property has been modeled in one of the following forms:

1- Effective frequency, in which users experience longer waiting time if they fail to board a vehicle (Gendreau 1984, de Cea and Fernandez 1993)

2- Penalty function, in which people on crowded routes experience higher cost (Spiess and Florian 1989)
3- Failure to board, in which people may fail to board a vehicle if it is full (Bell 2003, Kurauchi et al. 2003, and Schmöcker et al. 2008)

4- Prioritized set of choices, so people board the vehicle with available capacity according to a priority set (Hamdouch et al. 2004, Hamdouch and Lawphongpanich 2008).

Many of the approaches were suitable for transit systems with either short headways or unreliable scheduled service, so that the transit service is modeled by its frequency. The frequency-based approach captures the average performance of each transit route and does not provide any variation among different transit vehicle trips.

The newer type of transit network models is schedule-based, which models every transit vehicle trip separately from the others and which captures the dynamics of the system within time periods. One important assumption in the schedule-based models is that information about system performance is provided to the users through different information systems ranging from a schedule-book to online trip planning (or recently, smartphone applications). This modeling approach, although started in the 1960’s by Levin and Hedetniemi (1963) and Cooke and Halsey (1966), has been studied widely in the last 20 years with the most noted early model by Tong and Richardson (1984). The schedule-based network is usually modeled as a time-expanded network (Nuzzolo and Russo 1996, Nielsen and Jovicic 1999, Hamdouch et al. 2004, and Hamdouch and Lawphongpanich 2008). More analytic route choice behavior can be considered for the transit users with more information about the system (Hickman and Wilson 1995, Hickman and Bernstein 1997, and Nuzzolo et al. 2001). The adaptation of the hyperpath
approach to the schedule-based system was also studied by Nguyen et al. (1998), Nguyen et al. (2001), Noh et al. (2012a), and Khani et al. (2012b). In these cases, a set of attractive vehicle trips (a choice set) can be determined for transit users according to the schedule.

Modeling schedule-based transit systems with capacity constraints is a more complicated topic that researchers have studied in the last few years (Nguyen et al. 1998 and 2001, Hamdouch et al. 2004, Hamdouch and Lawphongpanich 2008, Nuzzolo et al. 2012, Noh et al. 2012b, and Khani et al. 2013). This variation of the models is the most relevant approach to the current research and a more in-depth review of the literature is provided in the following chapters. There are other transit assignment approaches in the literature such as the micro-simulation learning-based approach by Wahba and Shalaby (2005 and 2009) that are also reviewed.

The integration of the transit assignment model with the traffic assignment in the auto network introduces a multimodal assignment model, which has received some open discussion in the last decades. Many different approaches were proposed for its modeling, ranging from the full integration of the multimodal transportation network and assigning the users to the optimal path (Abdulaal and Leblanc 1979, Fernandez and de Cea 1994, Garcia and Marin 2005) to partial integration of the modes, in which the assignment is done separately by mode and the network loading is conducted jointly in the auto and transit network (Ziliaskopoulos and Wardell 2000, Abdelghany and Mahmassani 2001, Zhou and Mahmassani 2008).
Despite the fact that the literature is very broad in these multimodal transportation models, usually the transit modeling is completed in a simplified way and the transit user behavior receives less focus than the auto assignment. I.e., some sources of complexity, such as schedules of transit service, capacity of transit vehicles, and preferred arrival/departure times of the travelers, are not considered in the models. One source of the complexity in multimodal models is capturing intermodal trips (drive-to-transit trips) that include the assignment of the trip within both the auto and transit networks and embedding the choice of where and how to transfer from private to public mode (Modesti and Sciomachen 1998, Ziliaskopoulos and Wardell 2000, Lozano and Storchi 2002). The constraint on the park-n-ride location in a tour (i.e. people should return to the same park-n-ride location where they parked their car) and the constraint on capacity of the park-n-ride lot add additional complexity to the model (Festa 2009, Baumann et al. 2004, Nassir et al. 2012). The tour-based approach is now of higher interest to researchers, especially because the tour-based demand models are now starting to replace traditional trip-based models. Again, there are multiple approaches in the literature for intermodal assignment, but it is hard to find a comprehensive set of intermodal and multimodal assignment models that consider the network dynamics, transit capacity constraints, and realistic user behavior.

2.2. Research Goal and Objectives

From this review of relevant published works, schedule-based transit assignment was found to be a mathematically challenging problem and very few models in the literature are capable of solving real-world problems of reasonable size. Furthermore, the
transit assignment is mostly performed separately from auto traffic assignment, while the interaction of auto and transit is very important for finding a system equilibrium state. This interaction is also important for transportation infrastructure design, multimodal system management, and integrated system operations. Therefore, the high-level goal of this research is to model multimodal transportation systems in the most general but practical way. Toward this goal, a practical model for assignment on the transportation system, with focus on transit assignment, is the primary purpose of this study. Additionally, development of an advanced model for transit network modeling is pursued to support multimodal transportation system modeling. To achieve these goals, the following objectives are pursued:

1- The focus of the model is on transit assignment, and the effect of traffic conditions in the roadway network should be considered in modeling transit system performance.

2- The model should be flexible to represent transit users’ behavior in an appropriate way.

3- The transit system should be modeled at the highest resolution, so that more details about system performance are obtained in the results.

4- The solution method should be efficient so that real-world problems are solved in a reasonable amount of time.

5- The use of automated transit data should be considered for modeling the network, specifically for user information, for model calibration, and for model validation.
Because the model must be applicable to real-world cases, findings from these applications will be a good way to evaluate the achievement of these research goals.

2.3. **Fundamental Assumptions**

The main assumptions considered in this study are listed below, including assumptions on data availability and users’ level of information about the system.

1. The transit network is given and contains fixed routes with a predefined schedule. Vehicle deviation from the schedule is possible and can be estimated by considering the effect of traffic congestion and passenger boarding and alighting activities.

2. Transit demand is given, including travelers’ origins, destinations, and modes. For some applications, a complete passenger tour may be required as the input to the model.

3. Either the planned departure time (PDT) from the origin or the preferred arrival time (PAT) to the destination is known for each passenger trip.

4. Transit vehicle capacity is known and can differ among vehicles.

5. The auto network is given and may provide time-dependent link travel times. For simplicity, we assume that a dynamic traffic assignment can be performed using existing DTA software.

6. People are assumed to have perfect information about the transit vehicles’ arrival/departure time at each stop and the available capacity, based on their experience and/or from information provided by information systems such as websites, smartphones, etc. The relaxation of this assumption (users with
imperfect information or variation in perception of the travel cost) will result in a stochastic assignment model. However, the user perception of cost has to be measured in each case, and should be utilized in a stochastic assignment model.

In most cases, relaxation of each assumption may change the model to a more general case with additional complexity. However, even the problem with these assumptions is not trivial and is more complicated than what has been solved in the literature. Furthermore, if this problem is solved properly, it will provide useful information to transportation planners for planning and operations applications.
CHAPTER 3: TRANSIT ASSIGNMENT WITH CAPACITY CONSTRAINTS AND BOARDING PRIORITY

3.1. Review of Capacity Constrained and Stochastic Transit Assignment Models

The proposed models in this study are based on some specific work in the literature. However, they are not all in the field of schedule-based transit assignment. To provide a background about the mathematical formulation of the proposed models, a review of the related literature is provided here along with the analytic models of the transit assignment problem.

Capacitated traffic assignment was widely studied by Larsson and Patriksson (1995 and 1999). In their approach, the traditional optimization model of traffic assignment was modified by side constraints for link capacity, and a lagrangean relaxation method was proposed to solve the model analytically. A common idea for solving constrained assignment models is based on relaxing the side constraints and adding a penalty term to the objective function. The general form of these side constraints was used in the current study to model transit assignment with capacity constraints. More information about capacitated traffic assignment can be found in Patriksson (1994).

A stochastic traffic assignment model based on logit route choice was first introduced by Dial (1971), in which demand for each origin-destination pair (OD) is assigned to multiple paths based on the logit model. Alternately, Fisk (1980) proposed an optimization method for the stochastic traffic assignment with logit route choice. In the model, an entropy term was added to the objective function, pushing the demand to the
paths other than just the optimal path according to a logit function. The entropy term contains a parameter called the dispersion parameter that determines how users behave compared with the deterministic case. The optimality conditions of Fisk’s model were shown to be the same as those of a stochastic logit path choice model. For modeling stochastic transit assignment in this study, Fisk’s approach is used to capture the assignment as an optimization problem.

The concept of boarding priority in transit networks was studied by Nguyen et al. (1998) and Nguyen et al. (2001). They introduced the available capacity for transit users based on their priority level. Given a set of transit paths and different groups of passengers approaching from different sources or at different times, if the available capacity on the transit vehicle is limited, the priority is given to a specific group of passengers. This concept can help to model transit systems and user behavior more realistically. Schmöcker et al. (2008 and 2011) also applied a similar priority concept in their transit assignment models. They defined the boarding priority among passengers approaching the transit vehicle from outside, and among passengers on the same vehicle (staying on-board) to define the failure-to-board probability. Hamdouch and Lawphongpanich (2008) and Hamdouch et al. (2011) provide other examples using capacity constraints and boarding priority, in which a predefined set of boarding alternatives are used, and passengers take the transit vehicle with higher utility unless it is full. In the latter case, they try to take the second alternative and the process continues until they eventually get on-board. Noh et al. (2012b) and Khani and Hickman (2013) are
more recent works that incorporate the boarding priority and capacity constraint in modeling the stochastic transit assignment problem.

Although many transit assignment models have been proposed in the literature for schedule-based transit assignment and strategy/hyperpath approaches, few models proposed a closed form mathematical model for the problem. Spiess and Florian (1989) proposed a linear programming model for strategy-based transit assignment with combined frequencies for measuring passenger waiting times. Nuzzolo et al. (2001) and Nuzzolo et al. (2012) proposed a dynamic passenger decision process that captures this process both within a day and from day to day. Although modeling frequency-based systems, Schmöcker et al. (2008) and Schmöcker et al. (2011) modeled the capacity-constrained assignment as a fixed point problem. In this approach, the failure-to-board probability is the endogenous variable affecting the passenger flow. Nguyen et al. (2001) used a variational inequality approach to model the schedule-based assignment problem considering boarding priority. The solution algorithm is based on the hyperpath assignment introduced in Nguyen et al. (1988). In more recent research, Noh et al. (2012b) proposed a mathematical formulation with a capacity-related penalty term in the objective function.

3.2. Capacitated Transit Assignment with Passenger Boarding Priority

To define the transit assignment problem mathematically, an appropriate network structure is needed to represent the service schedule. As opposed to most of the models in the literature that used time-expanded networks, we propose a trip-based representation in this study. The trip-based network has the advantageous property of keeping the stops (or
nodes in the graph) unchanged, and makes the connection between stops using transit vehicle trips: e.g., two consecutive stops can be connected by more than one link, each representing a transit vehicle trip. The trip-based network has two differences with typical static networks:

1- Each link is defined by a triplet, indicating the upstream node, the downstream node, and the transit vehicle trip ID. With this property, more than one link (each with a different trip ID) can connect two nodes.

2- There are departure and arrival times associated with each link as well as a travel time or cost. This property makes the links available at some times and unavailable at other times. More specifically, links are available in all time points before their departure time and unavailable after that. Vehicle capacity is one other important attribute of each link.

Because of these properties, the network is similar to a static network in terms of the number of nodes and has less complexity than typical time-expanded networks. Furthermore, the connection of any pair of stops on the same transit route can be made directly using a longer link representing the vehicle trip while skipping the intermediate stops. A sample network in the trip-based structure is shown in Figure 2, and more details of the trip-based transit network are provided in chapter 4.
The proposed transit assignment model is formulated in a similar way as the capacitated traffic assignment model (Larsson and Pattrikson 1995 and 1999) with consideration of boarding priority. The traditional capacitated traffic assignment problem (CTAP) with separable link travel time functions is formulated as:

$$\text{CTAP:} \quad \min Z = \sum_{a} \int_{0}^{x_{a}} t_{a}(\omega)d\omega$$  \hspace{1cm} (1)$$

$$s.t. \quad \sum_{k} f_{rs}^{rs} = q_{rs} ; \forall r, s$$  \hspace{1cm} (2)$$

$$f_{rs}^{rs} \geq 0 ; \forall k, r, s$$  \hspace{1cm} (3)$$

$$x_{a} \leq C_{a} ; \forall a$$  \hspace{1cm} (4)$$

where:

$f_{rs}^{rs}$: flow on path $k$ of OD pair $rs$

$q_{rs}$: demand on OD pair $rs$

$t_{a}$: travel time (cost) on link $a$ (a function of $x_{a}$)

$x_{a}$: flow on link $a$, defined by:

$$x_{a} = \sum_{k,r,s} f_{rs}^{rs} \delta_{rs}^{ak} ; \forall a$$  \hspace{1cm} (5)$$

where $\delta_{rs}^{ak}$ is 1 if link $a$ is on path $k$ of OD pair $rs$ and is 0 otherwise.
For modeling the passenger boarding priority, either additional constraints or a penalty term in the objective function can be used. The method proposed in this study is to add a penalty term in the objective function, allowing passengers with the highest priority to use the remaining capacity of the link, and imposing a high penalty on passengers with a lower priority if no capacity is available for them. A penalty term is introduced in the objective function so that flow is shifted to other paths if boarding on the chosen path is not possible. The penalty function for each link is defined in the following general form:

$$\beta x_{ak}^{rs} e^{-\alpha p_{ak}^{rs}}$$  \hspace{1cm} (6)

where:

$\beta$: A large penalty value to prevent passengers from boarding the vehicle in link $a$ if no capacity is available. This penalty must be greater than the additional cost imposed on passengers if they decide to take the second best path to the destination.

$x_{ak}^{rs}$: The flow on link $a$ associated with the flow in path $k$ of OD pair $rs$, defined by:

$$x_{ak}^{rs} = f_k^{rs} p_{ak}^{rs}$$  \hspace{1cm} (7)

$\alpha$: The scale parameter, defined in a way that the model has a minimum precision in modeling the capacity constraint.

$p_{ak}^{rs}$: The available capacity on link $a$ for passengers traveling on path $k$ of OD pair $rs$. This variable is defined at each flow state based on the flow in the paths with higher priority than path $k$. The definition of the available capacity is:
\[ p_{ak}^{rs} = C_a - \sum_{k' \in PS(krs)} x_{ak}^{r's'} \]  \hspace{1cm} (8)

where

\( PS(krs) \): The priority set of path \( k \) for OD pair \( rs \) at a given link, containing the paths with higher priority than path \( k \) in using link \( a \). This priority set is based on the network topology and is an input to the model.

**Lemma 1**: For a given time resolution of \( l \) unit and desired flow resolution of \( \varepsilon \) in the solution, the penalty value can be replaced by:

\[ e^{-\alpha (\varepsilon - p_{ak}^{rs})} \] \hspace{1cm} (9)

To ensure that no penalty is applied to the passengers when there is available capacity on the vehicle, i.e., \( p_{ak}^{rs} \geq \varepsilon \), the penalty must be smaller than the time unit to be neglected, or \( \beta e^{-\alpha p_{ak}^{rs}} \leq 1 \), resulting in \( \beta \leq e^{-\alpha \varepsilon} \). By choosing \( \beta = e^{-\alpha \varepsilon} \), the penalty value will be \( e^{-\alpha (\varepsilon - p_{ak}^{rs})} \). This replacement results in the penalty term with one parameter.

**Lemma 2**: For a desired minimum penalty value of \( \gamma \) for links with no available capacity, one reasonable penalty value is:

\[ e^{2ln(\gamma)(1 - \frac{p_{ak}^{rs}}{\varepsilon})} \] \hspace{1cm} (10)

To ensure that a significantly large penalty is applied to the passengers when no capacity is available for their boarding, meaning \( p_{ak}^{rs} \leq \frac{\varepsilon}{2} \), the penalty must be greater
than the minimum penalty value \( \gamma \), or \( e^{-\alpha (\epsilon - p_{ak}^r)} \geq \gamma \), resulting in \( \alpha \geq \frac{2\ln(\gamma)}{\epsilon} \), and by choosing \( \alpha = \frac{2\ln(\gamma)}{\epsilon} \), the penalty value will be \( e^{2\ln(\gamma)(1 - p_{ak}^r/\epsilon)} \).

The capacitated transit assignment problem with boarding priority (CTAP-P) is modeled using the mathematical formulation shown below. In this model, it is assumed that the link travel times are constant and are not a function of flow. Note that the side constraints for link capacity (4) in the CTAP are replaced by constraints ensuring both capacity and boarding priority (14).

\[
\text{CTAP-P} \quad \min Z = \sum_a t_a x_a + \sum_{a k r s} \beta x_{ak}^r e^{-\alpha p_{ak}^r} \\
\text{s.t.} \sum_k f_{k}^{rs} = q_{rs}; \forall r, s \tag{11}
\]

\[
f_{k}^{rs} \geq 0; \forall k, r, s \tag{12}
\]

\[
x_{ak}^r \leq p_{ak}^r; \forall a, k, r, s \tag{14}
\]

with constraints defined as:

\[
p_{ak}^r = c_a - \sum_{k' r' s' \in PS(krs)} x_{a k'}^{r'}; \forall a, k, r, s \tag{15}
\]

\[
p_{ak}^r \geq 0; \forall a, k, r, s \tag{16}
\]

and (5) and (7)

In the proposed model, the objective function has a linear term of the total travel time (as the sum of link travel times), and the nonlinear capacity penalty is defined by (6), given the current flow on all other paths. Constraints (12) and (13) are the flow conservation and non-negativity constraints, respectively. Constraints (15) define the
available capacity for each group of passengers according to their level of priority, while the vehicle capacity constraints are ensured by (4). At each given flow configuration, the optimality conditions, defined by the Karush-Kuhn-Tucker (KKT) conditions, result in the following relationships, in addition to the feasibility constraints (12)-(13) and the definitions (5), (7) and (15)-(16):

\[ f_k^{rs}(c_k^{rs} + \sum_a \beta e^{-\alpha p_{ak}^{rs}} - u_{rs} + \sum_a v_{ak}^{rs}) = 0; \forall k, r, s \]  \hspace{1cm} (17)

\[ c_k^{rs} + \sum_a \beta e^{-\alpha p_{ak}^{rs}} - u_{rs} + \sum_a v_{ak}^{rs} \geq 0; \forall k, r, s \]  \hspace{1cm} (18)

\[ v_{ak}^{rs}(x_{ak}^{rs} - p_{ak}^{rs}) = 0; \forall a, k, r, s \]  \hspace{1cm} (19)

\[ x_{ak}^{rs} \leq p_{ak}^{rs}; \forall a, k, r, s \]  \hspace{1cm} (20)

\[ u_{rs} \geq 0; \forall r, s \]  \hspace{1cm} (21)

\[ v_{ak}^{rs} \geq 0; \forall a, k, r, s \]  \hspace{1cm} (22)

where:

\( u_{rs} \): Lagrange multiplier of the flow conservation constraints (12)

\( v_{ak}^{rs} \): Lagrange multiplier of the priority and capacity constraints (14)

The optimality conditions define an equilibrium state we call a Prioritized Equilibrium condition. In this state, based on the decision of the passengers with higher boarding priority, the equilibrium for a group of passengers is defined. There are two possibilities in the optimality conditions:
1. In the first case, when there is available capacity for passengers in path \( k \) of OD pair \( rs \) to travel on link \( a \), \( p_{ak}^{rs} \) takes a positive value, resulting in \( e^{-\alpha p_{ak}^{rs}} = 0 \). Therefore, the traditional capacitated equilibrium holds by:

\[
f_k^{rs}(c_k^{rs} - u_{rs} + \sum_a v_{ak}^{rs}) = 0; \forall k, r, s
\]

\[
c_k^{rs} - u_{rs} + \sum_a v_{ak}^{rs} \geq 0; \forall k, r, s
\]

The above equations indicate that the value of multiplier \( u_{rs} \) is the minimum travel cost of users in OD pair \( rs \). This minimum cost is equal to the sum of the path travel time \( c_k^{rs} \) and the sum of the multipliers of the capacity constraints, \( v_{ak}^{rs} \), on all links of the path \( k \). In other words, if all the links in path \( k \) have available capacity, the minimum travel cost for this group of passengers is equal to the sum of the path travel time and shadow price of capacity (either zero or a positive value). More specifically, if \( x_{ak}^{rs} < p_{ak}^{rs} \) for all links in path \( k \) (there is no binding capacity constraint), the constraints (14) are not binding, and then \( v_{ak}^{rs} = 0 \), and \( u_{rs} = c_k^{rs} \).

2. In the second case, when no capacity is available for the group of passengers, \( p_{ak}^{rs} \) will be zero, resulting in \( \beta e^{-\alpha p_{ak}^{rs}} = \beta \). The optimality condition for this case is as follows:

\[
f_k^{rs}(c_k^{rs} + \sum_a \beta - u_{rs} + \sum_a v_{ak}^{rs}) = 0; \forall k, r, s
\]

\[
c_k^{rs} + \sum_a \beta - u_{rs} + \sum_a v_{ak}^{rs} \geq 0; \forall k, r, s
\]

The above formulation, with the assumption of a large value for \( \beta \), indicates that the value of multiplier \( u_{rs} \) is the minimum travel cost for the group of passengers in OD
pair rs. This minimum cost is equal to the sum of the path travel time $c^rs_k$, the sum of multipliers of the capacity constraints, $v^rs_{ak}$, in all links of the path $k$, and the sum of the capacity penalties of $\beta$ for the links with no available capacity in path $k$. Because $\beta$ is a large number, $u_{rs}$ will be significantly larger than the original path cost. This is intuitive considering the passengers failing to get on the transit vehicle because of limited capacity. Then, the flow on this path will be zero and the equilibrium conditions hold automatically.

3.3. Extension to Stochastic Transit Assignment with Logit Route Choice

The problem defined above can be extended to the case that users choose their paths stochastically based on the expected utilities of the paths assuming that they have different perceptions of the path cost. The assumption is that the path choice behavior is modeled by a sequential logit model. To model this behavior of the users, an entropy term is added to the objective function of the model so that the demand is distributed over a set of alternative paths according to their utilities. For this purpose, the entropy term introduced by Fisk (1980) is used to represent the logit path choice in the assignment model. Thus, the proposed stochastic transit assignment model with boarding priority and capacity constraints (SCTAP-P) is:

$$
\text{SCTAP-P} \quad \min Z = \sum_a t_a x_a + \sum_{akrs} \beta x^rs_{ak} e^{-\alpha p^rs_{ak}} + \frac{1}{\theta} \sum_{krs} f^rs_k \ln f^rs_k \quad (27)
$$

$$
s.t. (12)-(16), (5) \text{ and } (7)
$$

The first and second terms of the objective function are similar to the CTAP-P model and the third term is the entropy term to push users to attractive paths other than the
optimal path. All the constraints, including the flow conservation and capacity constraints, are similar to the CTAP-P problem defined earlier. The parameter $\theta$ is the dispersion parameter representing the level of information or the perception of users about the system. The higher value of $\theta$ indicates the higher level of information or higher accuracy in perception of travel cost by users. The optimality conditions of the model are as follows, in addition to the feasibility constraints (12)-(13) and definitions (5), (7) and (15)-(16):

$$f_{ks}^{rs} \left( \frac{1}{\theta} (\ln f_{ks}^{rs} + 1) + c_k^{rs} + \sum a \beta e^{-ap_{ak}^{rs}} - u_{rs} + \sum a v_{ak}^{rs} \right) = 0 \ ; \ \forall k, r, s \quad (28)$$

$$\frac{1}{\theta} (\ln f_{ks}^{rs} + 1) + c_k^{rs} + \sum a \beta e^{-ap_{ak}^{rs}} - u_{rs} + \sum a v_{ak}^{rs} \geq 0 \ ; \ \forall k, r, s \quad (29)$$

$$v_{ak}^{rs} (x_{ak}^{rs} - p_{ak}^{rs}) = 0 \ ; \ \forall a, k, r, s \quad (30)$$

$$x_{ak}^{rs} \leq p_{ak}^{rs} \ ; \ \forall a, k, r, s \quad (31)$$

$$u_{rs} \geq 0 \ ; \ \forall r, s \quad (32)$$

$$v_{ak}^{rs} \geq 0 \ ; \ \forall a, k, r, s \quad (33)$$

These conditions result in a straightforward logit route choice model if there is available capacity for a group of passengers on all links of a path, meaning positive values for all $p_{ak}^{rs}$ among all links $a$ in path $k$. 

$$f_{ks}^{rs} = q_{rs} \frac{e^{-\theta(c_k^{rs} + \sum a v_{ak}^{rs})}}{\sum_j e^{-\theta(c_j^{rs} + \sum a v_{aj}^{rs})}} \quad (34)$$
On the other hand, when a vehicle is full and no capacity is available for boarding, \( p_{ak}^{rs} = 0 \), and the logit route choice model will include the capacity penalty of \( \beta \) in the utility of path \( k \):

\[
 f_k^{rs} = \frac{q_{rs} e^{-\theta(c_k^{rs} + \sum_a \beta + \sum_a v_{ak}^{rs})}}{\sum_f e^{-\theta(c_f^{rs} + \sum_a \beta + \sum_a v_{af})}} \tag{35}
\]

The above formula implies that the utility of path \( k \) includes the capacity penalty of \( \beta \) if the vehicle is full, resulting in zero probability of choosing path \( k \) for OD pair \( rs \). In this case, people have to choose other alternatives according to the other path utilities.

### 3.4. Dynamic Multimodal Assignment Model

For the comprehensive modeling of transportation systems, the multimodal assignment problem is formulated using a mathematical programming model. This model, which is the general case of the assignment problem, has the following properties:

- Auto travel time on a link where public transit vehicles operate is a function of auto traffic flow and transit service characteristics such as transit vehicle speed and dwell time.

- Transit vehicles’ travel time where they operate in mixed traffic is a function of the traffic flow.

The above properties require a proper network representation so that the flow on auto links (streets) and on transit links (transit vehicle trips) are modeled separately. The proper network representation can have separate auto and transit links. A sample network in this format is shown in Figure 3. In this network, there is an auto link shown by a solid
line on which all types of vehicles including transit vehicles can travel. There are also other links shown by dashed lines that represent transit vehicle trips on which only transit passengers can travel. Similar to the transit network, transit links in the multimodal network have their departure time, travel cost and capacity (i.e. are in the trip-based format). The nodes of the multimodal network can be intersections in general, or transit stops if the stop is located in the middle of the link (a mid-block stop).

![Figure 3 A sample multimodal network](image)

The proposed multimodal assignment model, considering the interaction between auto and transit assuming that link travel time for both auto and transit is a function of multiple variables including auto and transit flow, is:

\[
MTAP \quad \min Z = \sum_a \int_0^X t_a(\Omega) d\Omega + \sum_{akrs} \int_0^X \beta e^{-\alpha p_{ak}} d\Omega \\
\text{s.t.} \sum_i f_{i}^{rs} = q_{rs}^A ; \forall r, s \tag{36} \\
\sum_k f_{k}^{rs} = q_{rs}^T ; \forall r, s \tag{37} \\
f_{h}^{rs} \geq 0 ; \forall h, r, s \tag{38}
\]

*Constraints (14)-(16), (5) and (7)*

where:

\[\Omega: \text{the vector of link flows},\]
\( q_{rs}^A \): auto demand in OD pair \( rs \)

\( q_{rs}^T \): transit demand in OD pair \( rs \)

\( f_i^{rs} \): \( l \)-th auto path in OD pair \( rs \)

\( f_k^{rs} \): \( k \)-th transit path in OD pair \( rs \)

\( f_h^{rs} \): \( h \)-th path (auto or transit) in OD pair \( rs \)

The main differences between the multimodal and transit models are:

1. The additional term for auto assignment in the objective function, the integral over auto links embedded in the first term, implies that the users select their path based on both modes’ performance. For example, transit passengers may not choose a transit route where a bus operates on a highly congested link, and auto users may decide not to drive on a street where multiple transit vehicles with high demand operate.

2. The variable link travel time for the transit vehicles results in variable link travel time for transit users. Therefore, the integral in the objective function is used to model the transit user equilibrium.

3. The additional constraints for flow conservation in the auto network (constraints (37)) imply that the problem is a non-linear multic commodity flow problem with capacity constraints on the transit side.

A stochastic transit assignment can be modeled using the above formulation by adding the entropy term to the objective function.
SMTAP:

\[
\min Z = \sum_a \int_0^x t_a(\Omega) d\Omega + \sum_{akrs} \int_0^x \beta e^{-\alpha r_k} d\Omega + \frac{1}{\theta} \sum_{krs} f_{krs} \ln f_{krs}
\]  \hspace{1cm} (40)

s.t. Constraints (14)-(16), (5), (7) and (37)-(39)

The proposed multimodal assignment model is an optimization problem with an asymmetric cost function. This is due to both the penalty term for the transit links and the interaction between auto and transit vehicles. In fact, the traffic congestion affects the transit vehicles’ travel time and it affects the passenger route choice. On the other hand, changes in the transit flow pattern will change the dwell time of transit vehicles, affecting roadway capacity for auto users. With some assumptions, the multimodal assignment model can be decomposed into an auto assignment and a separate transit assignment, similar to that proposed earlier. This is possible by diagonalizing the cost function and making the link travel times a function of auto users only in each step. The effect of the transit vehicles on traffic can be captured by iteratively solving the diagonalized model. That is, in each step, transit and auto models are solved separately, and their effect on each other is considered for the next iteration. However, the transit assignment model with capacity constraints and boarding priority is still an asymmetric optimization model because of the penalty term. To solve this problem, a simulation-based approach is proposed in chapter 5. In the simulation-based assignment, the KKT conditions are insured by either:

- the desired path is available and passenger can complete the trip, or
• the path is not available because of capacity constraint, and the high
experienced path cost is modeled by failure to board.

The simulation-based algorithm iterates by evaluating the above conditions and
assigning the passengers in the second category to a new path until all (or
significantly high proportion) of the passengers experience the first condition. At
convergence, the optimal solution is the equilibrium state. In the next chapter, a set of
efficient transit path algorithms are developed which are the requirements of the
simulation-based transit assignment model explained in chapter 5.
CHAPTER 4: PATH ALGORITHMS IN SCHEDULE-BASED TRANSIT NETWORKS

4.1. Schedule-based Transit Path Algorithms in the Literature

Because of complicated user behavior in transit networks compared with auto networks, different variations of transit path models have been described in the literature. Quickest path, optimal path (with weighted cost function), strategy or hyperpath, and A* path are the main models in the literature. A review of these models is provided before introducing the path algorithms in this chapter.

The shortest path algorithm is the most common type of path algorithms. Along with mathematical models, labeling algorithms have been used extensively to solve this problem, starting with Dijkstra (1959) in a label-setting and Moore (1957) in a label-correcting form. In public transit networks, Tong and Richardson (1984) adapted Dijkstra’s algorithm for schedule-based transit networks. They provided label-setting algorithms to find either the quickest path or the least-cost path using a predefined generalized cost function. Koncz et al. (1996) also proposed a GIS-based algorithm to find the best transit paths from a single origin to a single destination. Their method gives more than one path for a given origin and destination, and the best path based on different criteria may be chosen by a traveler. However, their method only generates the paths with two or fewer transfers. Liu et al. (2001) used adjacency and connectivity matrices in a transit network to find paths with a fixed number of transfers. They also considered only major hubs as the location of transfers in their algorithm.
Spiess and Florian (1989) introduced the concept of a strategy in frequency-based transit networks and exploited the strategy for transit assignment. They used a combined frequency to determine the set of attractive routes at each stop, assuming that users choose the first vehicle to arrive from this set. Nguyen and Pallottino (1988 and 1989) also defined a similar model in a hyperpath network representation, leaving the opportunity to use any cost function for the set of attractive routes, and Gallo et al. (1993) formulated the graph theoretic definition of the hyperpath. de Cea and Fernandez (1993) proposed a formulation for modeling transit networks with the common line property and used it for frequency-based transit assignment with capacity constraints. The common line property is used to distribute excess demand to alternative routes in a similar way as the “strategy” assignment.

Marcotte et al. (2004) later incorporated the strategy concept into a general transportation network and proposed a traffic assignment model based on this approach. Nguyen et al. (2001) developed a hyperpath for modeling schedule-based transit networks, an approach followed by Nuzzolo et al. (2001). The model used a so-called logsum as the label of the stops and calculated the probability of each elementary path using the logit function. Hamdouch and Lawphongpanich (2008) and Hamdouch et al. (2011) proposed a time-expanded network for modeling transit paths on schedule-based networks and developed a hyperpath search algorithm for the assignment. A more recent study in this area is the link-based time-expanded network representation for schedule-based transit networks, suggested by Noh et al. (2012a). In that study, a logit-based cost function is used, similar to what introduced by Nguyen et al. (2001).
The A* algorithm was first proposed by Hart et al. (1968), but Chabini et al. (2002) is the most relevant study in the literature for its application in a transportation network, especially with dynamic link travel times. In that study, they used the minimum static travel time for the estimated travel time to the destination in a dynamic network. Bander and White (1991) developed a new algorithm called IA* for application in public transit. Even though the IA* algorithm is shown to be faster than other labeling algorithms, it requires some intermediate nodes called *island stops* with a higher chance to be on the optimal path between an origin and a destination. In addition, a Euclidean-based travel time is used in their model to generate the estimated travel time to the destination. To the best of the author’s knowledge, there is no other A*-type algorithm for transit networks in the literature. Therefore, this gap in the literature provided the motivation to develop a robust A*-type algorithm to find one-to-one shortest paths more quickly in transit networks, in addition to proposing more advanced shortest path and hyperpath algorithms.

4.2. Hierarchical Trip-Based Transit Network

4.2.1. GTFS Data

Google’s General Transit Feed Specification, GTFS, is a standard format for transit network data provided by transit agencies and shared publicly by Google in many metropolitan areas (see GTFS in the references). Because the chosen trip-based network representation is based on GTFS data, a brief description of GTFS is provided first, including the files and their format.
GTFS is a set of text files each describing one component of a transit network, while being linked together using common attributes. There are six required files and six optional files in each GTFS data set, containing very detailed information about the network. The required files with their definitions are shown in Table 1. The content of the files and the connection between them are also shown in Figure 4.

Table 1 GTFS Required Files

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agency.txt</td>
<td>contains information about the transit agencies that provide the data</td>
</tr>
<tr>
<td>Stops.txt</td>
<td>contains information about transit stops (individual locations where vehicles pick up or drop off passengers)</td>
</tr>
<tr>
<td>Routes.txt</td>
<td>contains information about transit routes (a group of vehicle trips which are displayed to riders as a single service)</td>
</tr>
<tr>
<td>Trips.txt</td>
<td>lists all vehicle trips and their routes (a trip is a sequence of two or more stops that occur at specific time)</td>
</tr>
<tr>
<td>Stop-times.txt</td>
<td>lists the times that a vehicle arrives at and departs from each stop for each trip (transit schedule)</td>
</tr>
<tr>
<td>Calendar.txt</td>
<td>defines the available services on different dates (specifies when service starts and ends, and days of the week in which the service is available)</td>
</tr>
</tbody>
</table>

Figure 4 GTFS files. Attributes with common symbol and/or color are used for linking two files
Similarly, in each file, some of the attributes are required and some others provide optional information. In Figure 4, only the required attributes that are used in this study are shown. In general, each route has a set of vehicle trips, and for each vehicle trip, there is a list showing the stops as well as the scheduled arrival and departure times for each stop. For each stop, we have information about the location and the type of stop. The calendar shows the service provided on each day of the week.

There are optional files in GTFS for fares and for the shape of the routes, which are not required in this study for network preparation. However, information in the shape file can be used for visualizing the routes, and fare information can be used in the route choice model if an appropriate utility function is estimated. The other optional file shows transfer links, which are needed in this study to build the network. However, since this file is not provided in most cases, we utilize a method to generate transfer links using the other required GTFS data.

4.2.2. Transit Network Hierarchy

Transit networks have an important property regarding the types of nodes. Specifically, in transit networks, any change in a path (i.e. transfer to another route) can be done only at some particular stops. A passenger on board may not consider transferring to another route at every stop along the current route, but only at stops where an alternative route is available. Figure 5 shows a schematic transit network containing blue routes in the east-west direction and red routes in the north-south direction. The associated stops (at a disaggregate level) are shown by small circles with the same color as the routes. Among all the stops, those within the black rectangles are transfer stops,
and transfers between the other stops are unlikely. In other words, the mid-block stops on the blue routes cannot be transfer stops because they are located on one route only, and walking to another stop with the same route is not a good option. Therefore, only a subset of stops can be considered as transfer stops. This property helps us to generate a hierarchical transit network with transfer stops at the higher level and where the non-transfer stops may be disregarded in a path search algorithm. It means, instead of visiting every stop between two transfer stops, the transfer stops are connected directly in the network hierarchy. However, the non-transfer stops are maintained in the network and are used as access and egress points.

Figure 5 Transfer stops in a typical transit network

The criteria for defining transfer stops may depend on the network properties. Obviously, in an urban area with high transit coverage, routes and stops are very close to each other, allowing many transfer stops. In a suburban area, there might be longer distances between routes and stops. Thus, to generate transfer links, in addition to
distance, the routes serving the two stops are checked for the possibility of a transfer. That is, if stop \( i \) and \( j \) are located on the same route and no other route serves any of them, despite the fact that they are very close to each other, it is not likely that a transfer happens from \( i \) to \( j \) or vice versa. In general, transfer links are generated between a pair of stops only if:

1. the distance between the stops is less than a certain value (e.g. 0.25 mile), and
2. there is at least one route serving one of the stops but not the other.

After generating transfer links, transfer stops are defined in order to build the network hierarchy. A transfer stop is defined as a stop at which a passenger has the option to transfer to another route. Therefore, a stop is transfer stop if:

1. it is located on more than one route, or
2. it has at least one inbound or one outbound transfer link.

Using this simple logic, the hierarchical transit network is generated.

### 4.2.3. Trip-Based Network Structure

There are two major network structures which have been used extensively in transportation models: node-based and link-based networks. In this study, we propose an alternative network representation which is suitable for modeling schedule-based transit networks, called the *Trip-Based* network. It is defined by a graph \( G(N, P, T) \), where \( N \) is the set of nodes (or stops, as used in the rest of this chapter), \( P \) is the set of transit trips \( p \) where each trip belongs to a route \( r \) in \( R \) (the set of transit routes), and \( T \) is the set of
transfer links. For each trip there is a list $S(p)$ which contains the stops served by the trip as well as the associated vehicle arrival and departure times. Each stop has a set $A(n)$ containing all the vehicle trips that serve that stop. The main advantage of the trip-based network, compared with node- or link-based networks, is that the connection between two stops (whether or not they are transfer stops) can be established within a single trip $p$ if the stops are located on the same trip. In contrast, in node-based and link-based networks, the connection between any two stops is made using a series of nodes and links. The other advantage of the trip-based method is that it contains time-dependent information about the network, and the network does not need to be expanded in time to represent the variation of service over time. In other words, by keeping the same number of stops (i.e., the same network size), the set $A(n)$ shows different services at stop $n$ over time, and $S(p)$ contains arrival/departure times (the schedule) of the transit vehicles along with the travel times between any pair of stops on the route.

Modeling transfer links is straightforward. In general, a transfer link $t$ connects two stops if a transfer is possible between them, using the criteria explained before. Obviously, each stop has a set $T(n)$ containing its outbound transfer links. A main feature of the trip-based network representation is that transfer stops are kept in a separate set $N_t$. This means that for non-transfer stops, the set $T(n)$ is empty and those stops can be neglected in shortest path algorithms.

4.2.4. Notations Used in the Path Algorithms

The trip-based network has significant advantages over existing network structures in terms of network connectivity, dynamic representation of service, and hierarchical
structure of transfer stops. These properties, together with the data availability through GTFS, provide motivation to develop efficient path algorithms which can be used in a variety of transit network modeling problems. Based on the behavior of transit users, and the type of the models, the path algorithms are divided to 3 categories: shortest path, hyperpath, and A* algorithms. A description of each algorithm, including the possible variations, is provided in this chapter. In the last section of this chapter, a simple example is utilized to illustrate the algorithms, and a real test case is used to compare the algorithms with the existing alternatives. The variables and parameters used in the algorithms are shown in Table 2.

**Table 2 Notation of the Variables Used in the Algorithm**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PAT_i$</td>
<td>Preferred arrival time to stop $i$</td>
</tr>
<tr>
<td>$dt$</td>
<td>The time window in which the alternative trips are determined, meaning that the arrival time to the destination in this time window is acceptable</td>
</tr>
<tr>
<td>$seq_i^p$</td>
<td>Sequence number of stop $i$ on trip $p$</td>
</tr>
<tr>
<td>$d_{p,i}$</td>
<td>Departure time of trip $p$ at stop $i$ (usually the same as arrival in the schedule and/or in GTFS data)</td>
</tr>
<tr>
<td>$v_{ij}^p$</td>
<td>In-vehicle time from stop $i$ to stop $j$ using trip $p$</td>
</tr>
<tr>
<td>$t_{ij}$</td>
<td>Transfer time from stop $i$ to stop $j$ (typically equals to the walking time between two stops)</td>
</tr>
<tr>
<td>$a_i$</td>
<td>Arrival/departure time label of stop $i$</td>
</tr>
<tr>
<td>$w_i^p$</td>
<td>Waiting time at stop $i$ for trip $p$, equals to the difference between the departure time of trip $p$ at stop $i$ and the arrival time label of stop $i$</td>
</tr>
<tr>
<td>$p_i$</td>
<td>Predecessor of stop $i$</td>
</tr>
<tr>
<td>$m_i$</td>
<td>The mode (trip number or transfer link) used to reach stop $i$ in forward algorithms, or to leave stop $i$ in backward algorithms</td>
</tr>
</tbody>
</table>
### Lower bound of the minimum time (cost) from stop $i$ to the destination stop

$e_i$

### Minimum travel time (cost) of the path from the origin to the destination through stop $i$

$F_i$

### The utility (a function of travel time or cost) of trip $p$ at stop $i$ to reach the destination

$cp_i$

### The probability of taking trip $p$ at stop $i$

$pr_i(p)$

### The set of transfer links at stop $i$

$T(i)$

### The set of routes at stop $i$

$R(i)$

### The set of vehicle trips at stop $i$

$p(i)$

### The attractive set of trips at stop $i$

$p_a(i)$

### Scan Eligible list, containing the stops with temporary labels in the algorithm

$SE$

---

**4.3. Trip-based Shortest Path (TBSP) Algorithm**

The proposed shortest path algorithm, called the *Trip-Based Shortest Path* (TBSP), is a labeling algorithm based on Tong and Richardson (1984), but exploiting the hierarchical trip-based network format. Consider a passenger traveling in a public transit system, who gets on a transit vehicle and moves toward the destination, not necessarily in a topological direction based on Raveau et al. (2012). At any point where a transfer is needed, the passenger will transfer to another route to get closer to the destination. The same logic works along the next route(s), until the passenger arrives to the destination stop. During this trip, each vehicle may visit many stops including non-transfer and transfer stops, but the passenger may not consider transferring at every stop since transfers impose an inconvenience to the passenger. In other words, as long as the transit vehicle is getting closer (in time or cost) to the destination, the passenger’s first option is to stay on board. The TBSP algorithm finds a path in the same way a passenger may
search for his/her best path in real world, and this search is made possible by using a hierarchical trip-based network. The general form of the algorithm is shown in Figure 6. This algorithm can be implemented as either label setting or label correcting form.

**Figure 6 The TBSP algorithm**

The algorithm in Figure 6 is a forward-labeling algorithm starting from the origin with \( \tau \) as the planned departure time (PDT). With slight modifications, a backward
algorithm is developed with backward search from the destination. In the backward case, a preferred arrival time (PAT) is used at the destination. In the proposed forward algorithm, the label of each stop is the earliest arrival time to that stop, from the origin. In general, any cost function can be used to generate different variations of the shortest path algorithm. Some of these variations with their definitions are listed below.

- **Minimal Transfer Path**: This variation generates the path with the minimum number of transfers. In this variation, the label of each stop is the number of transfers required to travel from the origin to that stop. The modifications in the algorithm for finding the minimal transfer path occur in lines 13, 14, 19, and 20 as below:

  13- If \((l_i+1<l_j):\)
  14- \(l_j=l_i+1, a_j=a_i+t_{ij}, p_j=i, m_j=’T’\)
  19- If \(l_i<l_j:\)
  20- \(l_j=l_i, a_j=d_{jp}, p_j=i, m_j=p\)

- **Least-cost Path**: In this variation, different weights are applied to costs associated with different elements of the trip. For example, this approach may be used to model the inconvenience of waiting and walking times compared with in-vehicle time. Assuming the weight \(a_k\) is applied to the \(k\)-th element of the passenger trip, the least-cost path algorithm is developed based on the shortest path algorithm with changes as shown below:

  13- If \((l_i+a_1.t_{ij}<l_j):\)
  14- \(l_j=l_i+a_1.t_{ij}, a_j=a_i+t_{ij}, p_j=i, m_j=’T’\)
  19- If \(l_i+a_2.w_{jp}+a_v.v_{jp}<l_j:\)
  20- \(l_j=l_i+a_2.w_{jp}+a_v.v_{jp}, a_j=d_{jp}, p_j=i, m=p\)
4.4. Trip-based Hyperpath (TBHP)

Public transit systems have two properties that motivate the investigation of a hyperpath (a set of prioritized alternatives at each stop) instead of the shortest path for modeling user behavior. The first property is the capacity constraint associated with transit vehicles. In a transit trip, travel cost is almost constant as long as the vehicle load is lower than a certain number, but increases drastically when the load exceeds the capacity of the vehicle. After reaching the vehicle capacity, passengers experience a failure to board the vehicle they had planned, and they have to change their path. The other property that lends itself to the hyperpath is the stochastic arrival and departure times of transit vehicles at stops which may result in missing the transit vehicle in transfers. In these cases, users may consider more than one path alternative to reach their destination.

A hyperpath, as defined by Nguyen and Pallottino (1988), is an acyclic sub-network with at least one path connecting the origin to the destination, and where at each node, there are probabilities for choosing the alternative links. The trip-based hyperpath model proposed in this study (TBHP) is a sub-network with a set of attractive trips at each stop $i$, denoted by $p_a(i)$, where each trip has a probability to be chosen by users. The combination of the cost of the attractive trips results in the cost from stop $i$ to the destination, where the cost on a single path is defined by any general function.
Figure 7 Attractive set of transit paths in a hyperpath

To generate a hyperpath to the destination stop $D$, we assume that each user has a preferred arrival time at the destination, $PAT$, and that there is a time boundary $dt$ in which arrival to the destination is accepted. For example, for a passenger aiming to arrive to the destination at or before $PAT=8:00\ am$, the algorithm may search for the hyperpath which takes him/her to the destination at the latest possible time between 7:30 and 8:00, assuming a 30 minute time boundary for the alternative trips ($dt=30\ min$). By extending this assumption to intermediate stops (and therefore to the origin stop), the set of alternative trips from stop $i$ to the destination $D$ is defined as the trips which serve the stop within the time window $[PAT_i-dt,\ PAT]$. Figure 7 shows the schematic set of attractive transit vehicle trips to the destination stop.
A suitable combined cost function can represent the transit users’ behavior more accurately, showing their route choice process. The literature shows that there are many variations for calculating the cost of each set of trips. As explained in Noh et al. 2012a, the logit-based function (with logsum as stop labels) is the best representation to model passengers’ route choice, especially in a schedule-based transit system. This function uses the cost of trips in the alternative set and gives the combined cost given the choice set at the stop. The mathematical representation of the logsum cost function is:

\[ l_i = \frac{1}{\theta} \ln(\Sigma_{p \in P_a} e^{-\theta c_{pi}}) \]  \hspace{1cm} (41)

Parameter \( \theta \) is the dispersion parameter to determine how diverse the alternatives in a hyperpath are. I. e., it determines the sensitivity of the users to cost differences across multiple paths. For example, with a large value of \( \theta \), few (or one) alternative(s) may be included in a hyperpath, while with a smaller (although positive) value of \( \theta \), more alternatives are included in the hyperpath. The utility of each trip can be defined by a weighted sum of different components of the trip (the path utility function), using the factor \( \alpha_k \). The proposed TBHP algorithm, in label-setting form, is shown in Figure 8. After finding the hyperpath, the probability of selecting alternatives at each stop can be calculated using the logit model:

\[ Pr_i(p) = \frac{e^{-\theta c_{pi}}}{\Sigma_{q \in P_a} e^{-\theta c_{qi}}} \]  \hspace{1cm} (42)
The TBHP algorithm

4.5. Trip-based A* (TBA*)

Although the proposed TBSP algorithm is efficient enough for planning purposes, sometimes, especially in very large networks, a path is needed in a very short amount of time. For this goal, an A*-based algorithm is proposed for transit networks. The main difference between A* and the typical labeling algorithms is that, in A*, an estimated value of the travel time, $e_{iD}$, from node $i$ to the destination node $D$ is used to estimate the
total travel time $F_i$ for the best path from the origin to the destination through node $i$. This estimate helps the algorithm to find the shortest path to the destination in fewer computational steps by limiting the search space (pruning off some nodes in the shortest path tree). However, using early termination of the algorithm for efficiency purposes, the algorithm may result in a path which is not necessarily optimal. This efficiency-optimality trade-off can be controlled either by using different values for the lower bound of the travel time to the destination or by using a less strict termination criterion.

Travel time estimation based on the Euclidean distance is not the most appropriate assumption in transit networks, since it is possible that in a passenger’s journey, a route does not move toward the destination in distance, but gets closer to it in time. In other words, one may use a route to go to a transfer stop which is a greater distance from the destination, but this transfer provides the opportunity of boarding another route which takes the passenger to the destination faster. For this reason, a new lower bound is defined on travel times based on the travel time components in transit. Assuming constant in-vehicle time for each vehicle trip during a day, and because transfer time is equal to the walking time and is constant, the only source of variation in the passengers’ experienced travel times is the waiting time. For example, two paths from the same origin to the same destination, at different times of day, may have equal in-vehicle time and walking time, but the waiting time (due to different schedules) makes the total travel time different. To find the lower bound of the travel time, we assume that there is an arbitrary path in which every transfer is made with zero waiting time. If we use the trip with minimum in-vehicle time, the resulting travel time is a lower bound of the travel time to
the destination and can be used in the proposed A* algorithm. The procedure of finding such a path is similar to the TBSP algorithm with some modifications in lines 16-20:

16-
17- $\forall p \in P(i)$
18- $\forall j \in S(p)$ with $seq_j > seq_i$:
19- If $l_i + v_{ijp} < l_j$:
20- $l_j = l_i + v_{ijp}$, $a_j = d_{jp}$, $p_j = i$, $m_j = p$

The transit trip-based A* algorithm (TBA*) is a general form of the TBSP in terms of using the estimated travel times $F_i$ as the substitute labels. The algorithm also has an advantage over the existing A* algorithms in that it uses the hierarchical trip-based network and can be more efficient in application. The termination condition is the selection of the destination stop for labeling. The difference between TBSP and TBA* is that TBA* has to be in label-setting form. The pseudocode representation of the TBA* algorithm is shown in Figure 9. Similar to the TBSP algorithm, TBA* can also be implemented with any generalized cost function and/or in a backward direction.
Regarding the effectiveness of the TBA* algorithm in finding the optimal path, it can be proved that the proposed lower bound helps the algorithm to find the optimal path. In a typical transit path, travel time is a combination of walking times (access and egress), waiting time, and in-vehicle time as shown below:

$$t = \sum_{i\in W} t_i^w + \sum_{i\in D} t_i^d + \sum_{i\in V} t_i^v \quad (43)$$
where:

- \( t_i^W \): the time of waiting at link \( i \), when \( W \) is the set of waiting links in the path
- \( t_i^D \): the time of walking at link \( i \), when \( D \) is the set of walking links in the path
- \( t_i^V \): the time of in-vehicle at link \( i \), when \( V \) is the set of in-vehicle links in the path

The estimated lower bound in TBA*, shown by \( \hat{t} \), is the combination of the in-vehicle time and walking time:

\[
\hat{t} = \sum_{i \in D} t_i^d + \sum_{i \in V} t_i^v
\]

which is less than or equal to the total travel time, \( t \). This guarantees the optimality of the solution in the TBA* algorithm (see Hart et al. 1968). In other words, since \( \hat{t} < t \) for every path in the same OD pair, the shortest path is not excluded from the search, and the algorithm will find that path as the optimal solution. Compared with the distance-based lower bound estimate used in the literature, the TBA* algorithm can be even more efficient in estimating a lower bound of travel time. In the distance-based method, the estimated lower bound on travel time is:

\[
\tilde{t} = \frac{d}{\bar{v}}
\]

which is not as accurate as the proposed lower bound:

\[
\hat{t} = \sum_{i \in D} t_i^d + \sum_{i \in V} t_i^v = \sum_{i \in D} \frac{d_i^w}{\bar{v}_w} + \sum_{i \in V} \frac{d_i^v}{\bar{v}_v}
\]

By comparing equations (45) and (46), since the walking speed is lower than the average travel speed, and also the Euclidian distance is the minimum distance between
point A and B, the resulting distance-based lower bound of travel time is expected to be lower than the proposed lower bound. In other words, the relationship

\[ \bar{t} \leq \hat{t} \leq t \]  

holds, meaning that the TBA* outperforms the existing model both in efficiency and effectiveness of the results.

4.6. Numerical Tests of the Path Algorithms

4.6.1. An Illustrative Example

To show how the proposed algorithms work, they are tested in a simple transit network (Tong and Richardson 1984) containing 15 stops, 5 routes and 20 vehicle trips (see Figure 10). Based on the transfer stop definition in section 4.1, stops 1, 6, and 14 are transfer stops with more than one route, and stops 3, 4, 5, 12, and 13 are transfer stops with transfer links. The remaining stops are non-transfer. Table 3 shows the list of vehicle trips for each route and the scheduled times for the trips. The transfer time is 1 minute for all transfer links, shown by the dashed lines in Figure 10.

Figure 10 The sample network
Table 3 The Schedule of the Network Shown in Figure 10

<table>
<thead>
<tr>
<th>Route ID</th>
<th>Trip ID</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>1001</td>
<td>10:00, 10:02, 10:04</td>
</tr>
<tr>
<td></td>
<td>1002</td>
<td>10:10, 10:12, 10:14</td>
</tr>
<tr>
<td></td>
<td>1003</td>
<td>10:20, 10:22, 10:24</td>
</tr>
<tr>
<td></td>
<td>1004</td>
<td>10:30, 10:32, 10:34</td>
</tr>
<tr>
<td>Blue</td>
<td>2001</td>
<td>10:06, 10:08, 10:10</td>
</tr>
<tr>
<td></td>
<td>2002</td>
<td>10:16, 10:18, 10:20</td>
</tr>
<tr>
<td></td>
<td>2003</td>
<td>10:26, 10:28, 10:30</td>
</tr>
<tr>
<td></td>
<td>2004</td>
<td>10:36, 10:38, 10:40</td>
</tr>
<tr>
<td>Purple</td>
<td>3001</td>
<td>10:00, 10:02, 10:04, 10:06, 10:08, 10:10, 10:12</td>
</tr>
<tr>
<td></td>
<td>3002</td>
<td>10:07, 10:09, 10:11, 10:13, 10:15, 10:17, 10:19</td>
</tr>
<tr>
<td></td>
<td>3003</td>
<td>10:14, 10:16, 10:18, 10:20, 10:22, 10:24, 10:26</td>
</tr>
<tr>
<td></td>
<td>3004</td>
<td>10:21, 10:23, 10:25, 10:27, 10:29, 10:31, 10:33</td>
</tr>
<tr>
<td>Orange</td>
<td>4001</td>
<td>10:08, 10:10, 10:12, 10:14</td>
</tr>
<tr>
<td></td>
<td>4002</td>
<td>10:18, 10:20, 10:22, 10:24</td>
</tr>
<tr>
<td></td>
<td>4003</td>
<td>10:28, 10:30, 10:32, 10:34</td>
</tr>
<tr>
<td></td>
<td>4004</td>
<td>10:38, 10:40, 10:42, 10:44</td>
</tr>
<tr>
<td>Green</td>
<td>5001</td>
<td>9:55, 9:57, 10:04, 10:06</td>
</tr>
<tr>
<td></td>
<td>5002</td>
<td>10:05, 10:07, 10:14, 10:16</td>
</tr>
<tr>
<td></td>
<td>5003</td>
<td>10:15, 10:17, 10:24, 10:26</td>
</tr>
<tr>
<td></td>
<td>5004</td>
<td>10:25, 10:27, 10:34, 10:36</td>
</tr>
</tbody>
</table>

In this example, the shortest path is sought from stop 1 starting at time 10:00 to all other stops in the network. The label-setting form of the TBSP algorithm is tested first and the resulting path to stop 6 (as an example) is to take the red route (trip 1001) first and then to transfer to the blue route (trip 2001) using the transfer link connecting stop 3.
to stop 4. During this algorithm, 8 stops were added to the scan-eligible (SE) list and then processed to update labels; this implies 8 iterations of the algorithm. Using an existing labeling algorithm (i.e. without using the hierarchical trip-based network), all the stops are added to the SE list and processed to complete the algorithm (i.e. 15 iterations). If a label-correcting algorithm is used, the number of iterations in the TBSP algorithm is 9 as opposed to 17 in the non-trip-based algorithm.

In the next test, TBA* is used to find the shortest path from stop 1 to stop 6, leaving the origin at time 10:00. The lower bound of travel time to stop 6 is calculated in the pre-processing stage and used in the TBA* algorithm. These lower bounds for stops 1 to 15 are, respectively \( \{8, 7, 5, 4, 2, 0, 10, 8, 6, 4, 2, 4, 2, 6, 2\} \). The output of the TBA* algorithm is the same as the result from the TBSP (taking trips 1001 and 1002), but the TBA* produced this path after only 3 iterations. In fact, after adding stops 1, 6, 3, 14, 4, and 12 to the SE list, and only processing stops 1, 3, and 4, the algorithm terminates and the shortest path is found. In this combination of origin, destination, and departure time, the result of the TBSP and the TBA* are the same, indicating that the TBA* has given the optimal solution with fewer computations.

Finally, we have tested the TBHP algorithm to find a sample hyperpath. In this test the optimal hyperpath with a 10-minute time boundary is searched, to arrive at stop 6 with PAT=10:25. A weight of 2 is considered for walking and waiting times, when compared to in-vehicle time. The output of the algorithm in the form of the attractive set of trips at each stop is shown in Table 4. Additionally, the probability of choosing each alternative is calculated using a logit model and is shown in the table. The reason that the
The selection probability is close to one for the lowest-cost trip, and almost zero for the other trips, is that a dispersion factor of $\theta = 1$ is used in the test. Lower values of the parameter will result in more chance of selecting a path other than the optimal path.

### Table 4 The Result of the TBHP Algorithm in the Sample Network

<table>
<thead>
<tr>
<th>Stop</th>
<th>Label</th>
<th>Latest Departure Time to Ensure Destination PAT</th>
<th>Attractive Trips</th>
<th>Departure Time</th>
<th>Successor Stop</th>
<th>Probability of Taking each Trip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>10:15</td>
<td>3002 1002 5003</td>
<td>10:07 10:10 10:15</td>
<td>6 3 14</td>
<td>0.000 0.007 0.993</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>10:12</td>
<td>1002</td>
<td>10:12</td>
<td>3</td>
<td>1.000</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>10:19</td>
<td>Transfer</td>
<td>10:19</td>
<td>12</td>
<td>1.000</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>10:19</td>
<td>Transfer 2002</td>
<td>10:19 10:16</td>
<td>12 5</td>
<td>0.953 0.047</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>10:21</td>
<td>2002 Transfer</td>
<td>10:18 10:21</td>
<td>6 13</td>
<td>0.047 0.953</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>10:25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>10:09</td>
<td>3002</td>
<td>10:09</td>
<td>6</td>
<td>1.000</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>10:11</td>
<td>3002</td>
<td>10:11</td>
<td>6</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>10:13</td>
<td>3002</td>
<td>10:13</td>
<td>6</td>
<td>1.000</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>10:15</td>
<td>3002</td>
<td>10:15</td>
<td>6</td>
<td>1.000</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>10:17</td>
<td>3002</td>
<td>10:17</td>
<td>6</td>
<td>1.000</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>10:20</td>
<td>4001 4002</td>
<td>10:10 10:20</td>
<td>13 13</td>
<td>0.000 1.000</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>10:22</td>
<td>4001</td>
<td>10:22</td>
<td>13</td>
<td>1.000</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>10:18</td>
<td>5002 4001 4002</td>
<td>10:07 10:08 10:18</td>
<td>6 12 12</td>
<td>0.000 0.000 1.000</td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>10:14</td>
<td>5002</td>
<td>10:14</td>
<td>6</td>
<td>1.000</td>
</tr>
</tbody>
</table>
4.6.2. A Real Case Test

As mentioned before, the proposed network model and path algorithms were designed mainly for planning purposes to solve the assignment models proposed in chapter 3. These path algorithms are the components of a transit and intermodal assignment model in an integrated travel demand model (an activity-based travel demand model, a dynamic traffic assignment, and a schedule-based transit assignment model). The integrated model and its application are described in the following chapters. However, computational test of the transit path algorithms is provided here for the large-scale test network.

The case study of the integrated travel model is the regional transportation network in Sacramento, CA. GTFS data were used to build the schedule-based transit network (Sacramento Regional Transit), which contains 2,880 stops, of which 1,028 are transfer stops, and 75 routes providing 2,294 vehicle trips in a typical weekday. On average, each vehicle trip serves about 38 stops.

The trip-based path model has been implemented in C++ and has 3 main functions associated with the proposed path algorithms. The algorithms were coded in both label-correcting and label-setting format, except for the TBA*. In the label-correcting format, a double-ended queue was used for the SE list, while in the label-setting algorithm, the scan-eligible nodes are kept in a priority queue with a heap structure. The conventional labeling algorithms (without using the hierarchical trip-based network) were also coded in the same way to evaluate the trip-based path models. These algorithms are called baseline in the tests. During the tests, a set of origins, destinations, and times of travel (either PAT or PDT) were generated randomly, the algorithms were run, and the average
computational times were recorded. A summary of the results is shown both numerically and graphically in Figure 11. As shown in the figure, the algorithm in the trip-based format reported better computational efficiency compared with the baseline cases. Moreover, the figure shows very small computational time in the proposed A*-type algorithm both in the trip-based and the non-trip-based format. In both the shortest path and hyperpath algorithms, the label-setting format resulted in a slightly lower computational time, which may be the result of using the priority queue in the SE list.

![Computational Test Graph](image)

**Figure 11 Computational test of the trip-based path algorithms with the Sacramento network**

In addition to the average values of the computational times, the distributions of the computational time over a random set of <O,D,t> triplets are shown in the following graphs. In Figure 12, the computational time distributions are shown for the shortest path algorithms in label correcting (a) and label setting formats (b). By comparing the trip-based algorithms with the conventional algorithms, lower computational time, even with lower variation, is observed. In cases where the computational time is very small and far
from the average, it is likely that the origin or destination nodes are connected to the network with limited service (e.g. in off-peak or night times), resulting in fewer steps to find the shortest path tree. The comparison of the label correcting and label setting algorithms shows that label correcting increases the computational time significantly in the conventional algorithms, and makes the distribution less spread out from the average, while in the trip-based algorithms, the improvement in the label setting is less significant.

Figure 13 shows the computational time distribution in the hyperpath algorithms, and almost similar observations as the shortest path case can be made. In fact, the time is less spread out from the average in TBHP, and the label-setting form improves the conventional hyperpath more than the trip-based hyperpath.

The last observation is on the A*-type algorithms for which the results are shown in Figure 14. Note that because the A* algorithms are single-origin and single-destination algorithms, they have to be in label-setting format, to ensure the optimality condition during the algorithm run, and not only in the final solution as opposed to the other algorithms. Therefore, only the label-setting form of the A* algorithms are implemented and tested. In the figure, while the average computational time is decreased significantly compared with the shortest path algorithms, the TBA* algorithm resulted in a very narrow range (and at the same time smaller values) of computational times compared with the A* algorithm. This may be because the trip-based algorithm makes the connection between stops by the trips, and therefore with very few steps, any destination stop can be connected to the origin stop. On the other hand, the non-trip-based algorithm requires that every intermediate stop be processed until the destination stop is reached,
meaning that closer destination stops will be found in significantly different amount of computational steps and time than farther destination stops.
Figure 12 Distribution of the computational time for shortest path algorithms over random set of origin-destination-times
Figure 13 Distribution of the computational time for hyperpath algorithms over random set of origin-destination-times
Figure 14 Distribution of the computational time for A* algorithms over random set of origin-destination-times
CHAPTER 5: SIMULATION-BASED ASSIGNMENT MODEL FOR CONGESTED TRANSIT NETWORKS

5.1. Transit Assignment Algorithms in the Literature

There have been many transit path choice and assignment models proposed in the literature. In many of these models, a frequency-based network has been studied. However, schedule-based network modeling has been of growing interest to researchers and modelers in public transit. Here a review of the most related work in transit assignment is provided.

Tong and Richardson (1984) were the first to adapt Dijkstra’s shortest path algorithm to find the quickest and least cost path in public transit networks with a predefined schedule. Their model, although was not the first schedule-based model in the literature, was the starting point of extensive modeling in schedule-based networks. Spiess and Florian (1989) introduced the concept of a “strategy” in frequency-based transit networks and used the strategy model for transit assignment. The main idea in a strategy is the assumption that users define a set of attractive transit routes at each stop, and they board the first vehicle arriving within this attractive set. They used the combined frequency of the attractive routes to model the waiting time at each stop. Spiess and Florian (1989) tried to improve the model to incorporate crowding by penalizing passengers on crowded routes. Nguyen and Pallattino (1988 and 1989) used the term “hyperpath” and defined it in a frequency-based transit network, which is conceptually similar to a strategy. The difference is that the hyperpath is a general path model that can be used with any type of
generalized cost function for the set of attractive routes. Nguyen et al. (1998) proposed a logit-based cost function for determining a hyperpath in a frequency-based transit network.

In the area of transit assignment considering system dynamics and the capacity of transit vehicles, Nguyen et al. (2001) and Nuzzolo et al. (2001) applied a multi-path assignment model to schedule-based transit networks. The former model was based on the hyperpath model with a logit route choice proposed by the same authors (Nguyen et al. 1998), and the latter was a different path assignment approach to model the day-to-day as well as within-day dynamic of passenger path choices. A network expansion technique for modeling the schedule in transit networks was proposed by Hamdouch and Lawphongpanich (2004). They developed a hyperpath search algorithm and a transit assignment model using the time-expanded network. In their model, transit users choose their path among a set of paths which are ranked based on their utilities. Whenever a higher utility path is not available due to a capacity constraint, the next path is used by the passenger. Hamdouch et al. (2008) extended the model to take into account the probability of finding a seat on the transit vehicle, in addition to the overall capacity constraint. Schmöcker et al. (2008) developed a hyperpath-based assignment with consideration of a failure-to-board probability to solve the capacity-constrained transit assignment. Their model takes into account priority of on-board passengers, but assumes that passengers mingle on the platform and uses a Markov-based loading process. Schmöcker et al. (2011) extended the model to the case that the seating probability is also considered in the hyperpath. Nuzzolo et al. (2012) developed a dynamic assignment
model with explicit capacity constraints. A sequential discrete choice model is proposed in their study for choosing departure time, boarding stop, and transit run based on the utility of the choices. Noh et al. (2012a) used a different network representation and defined a link-based time-expanded (LBTE) transit network. They proposed a hyperpath model in the LBTE network with a logsum cost function for stochastic transit assignment. Noh et al. (2012b) also proposed a logit-based stochastic assignment model in the LBTE network, which takes into account the capacity of the transit vehicles by defining a penalty cost for overloaded vehicles. Khani et al. (2012b) also used the logit-based path choice model and defined a set of path algorithms in a hierarchical trip-based transit network that improve the computational time for a schedule-based transit assignment. Finally, Khani et al. (2013) extends the models in Khani et al. (2012b) to create a schedule-based transit assignment model that incorporates both a logit-based assignment and strict capacity constraints. This model is called FAST-TrIPs (Flexible Assignment and Simulation Tool for Transit and Intermodal Passengers) and is explained in this chapter.

5.2. The Simulation-based Transit Assignment Model

A numerical method is developed in this chapter to solve the transit assignment models proposed in chapter 3. Starting with the first problem, the schedule-based transit assignment with capacity constraints and boarding priority, the assumption that people choose the path with the minimum travel cost results in a shortest path. In fact, in an instance of the network, when a new passenger is added to the system, regardless of the capacity constraints, the passenger tries to find the shortest path between their origin and
destination. This shortest path algorithm is implemented based on the TBSP algorithm. The algorithm can take into account the parameters of the transit path cost such as waiting time, in-vehicle time, walking distances, transfer penalties, etc. By using the trip-based network structure, the path cost components can be considered in the algorithm without the need to expand the network in time. In other words, temporal variation of the network properties is considered in the trip-based network by defining different links for different transit vehicle trips. Additionally, by taking advantage of the transit network hierarchy (Khani et al. 2012b), the algorithm runs with greater computational efficiency.

In addition to the shortest path algorithm, to model a stochastic assignment, a hyperpath algorithm is developed to generate the set of paths for each OD pair. In the hyperpath algorithm, based on the TBHP introduced in the previous chapter, instead of choosing the optimal path, a set of attractive paths is generated, and based on the utility of the available paths, passengers are proportionally assigned to the alternatives. A logit-based function is used in the algorithm, which takes into account the utility values corresponding to the alternatives, and the logsum function defines the label of the stops based on their attractiveness.

In the implementation of the hyperpath algorithm, the attractiveness defined by the logsum formula is used as the cost function and the label of the stops. I.e., when \( u_i \) is the utility of path \( i \) from stop \( s \) to destination \( d \), and \( A_s \) is the set of prospective transit vehicle trips at stop \( s \), the label of stop \( s \) is defined as:

\[
-\frac{1}{\theta} \ln(\sum_{i \in A_s} e^{-\theta u_i})
\]  

(48)
The set of attractive paths can be defined in a time window, implying that the passengers prefer not to spend more than a certain amount of time waiting for a transit vehicle, regardless of how high its utility may be. In addition, the paths with significantly higher cost (lower utility value) compared to other paths may be removed from the set of prospective paths. The detailed implementation steps of the path algorithms are explained in chapter 4.

After defining the path choice set, the demand is assigned to the alternatives using the logit probability function. More specifically, each passenger at each decision point (e.g. origin, or transfer stop), decides about the next boarding vehicle by comparing the path utilities calculated by:

$$P_s(j) = \frac{e^{-\theta u_j}}{\sum_{i \in \Lambda_s} e^{-\theta u_i}}$$  \hspace{1cm} (49)

The dispersion parameter $\theta$ is used both in the hyperpath cost function and the path probability calculation, mainly to adjust the number of paths in the path set, and to prevent the dramatic increase in the attractiveness of a stop by adding more alternatives. In essence, this parameter shows the level of stochasticity in users’ path choice behavior. $\theta$ is positive, and higher values represents users’ higher level of information about the system and their attitude in choosing the optimal path. In contrast, a lower value of $\theta$ will result in more stochastic user behavior and a greater variety of paths for an OD pair.

After generating the path for every passenger, no matter which path algorithm is used, the decision of the users are determined. The next step is to load the demand to the network according to these decisions, and evaluate the outcome of their decisions. To
perform this loading, the other module of the assignment model is the mesoscopic simulation of passenger movements in the transit system, given the scheduled time for vehicle arrivals and departures. The main idea in using simulation is to load the demand to the network, to ensure the boarding priority, and to determine the capacity violations in the system. In the simulation model, transit vehicles and stops are given (system) elements, and passengers are the moving agents in the system. The interaction between the system and the passengers is modeled with predefined logic in a detailed approach. Since the service schedule is given for the vehicle trips, the arrivals and departures of the vehicles are defined as simulation events and are processed to determine the network state.

During the simulation, passenger objects are generated and are loaded to the boarding stops according to their predefined paths. The passengers form a queue at the stops regardless of their selected vehicle trip, and exit the queue for boarding in a first-in-first-out (FIFO) order. At each transit event, i.e. when a vehicle arrives at a stop, passengers first alight from the vehicle as needed and then passengers from the stop board the vehicle if they intend to take the current vehicle trip. In the case that the available capacity is less than the number of passengers wishing to board, using FIFO behavior for boarding, a limited number of passengers get on the vehicle and the remainder are labeled as passengers with unsuccessful paths. Along with this event simulation, the system models the passengers walking between stops, and loads them to the boarding stops or sends them to their final destinations. After running the simulation, passengers’
experiences are evaluated and a subset of passengers, generally those with unsuccessful paths, is selected for assignment to alternative paths in the next iteration.

The transit assignment model is an iterative process of the above steps until a termination criterion is met. Because the link costs are constant, the only parameter changing over the iterations is the available capacity for each specific group of passengers; this is shown mathematically in the optimization problem in chapter 3. Therefore, the network is updated according to the vehicle loads, and the assignment of the unsuccessful passengers is then performed in the updated network. In essence, after each simulation, the available capacity for each passenger in each path is determined, and the exponential term in the objective function will be either 0 or 1. Whenever there is no additional capacity on a link for a passenger, the penalty of $\beta$ is added to the link cost for the next iteration’s path assignment. The explicit definition of the available capacity, $p_{ak}^{rs}$ in the simulation results, transforms the assignment model to a Minimum Cost Flow (MCF) problem with modified link costs and capacities, as follows.

\[
\text{MCF: } \min Z = \sum_a \bar{t}_a x_a \tag{50}
\]

\[
\text{s.t. } \sum_k f_{k}^{rs} = q_{rs}; \forall r, s \tag{51}
\]

\[
f_{k}^{rs} \geq 0; \forall k, r, s \tag{52}
\]

\[
x_a \leq \bar{C}_a; \forall a \tag{53}
\]

where:

$\bar{t}_a$: updated link travel time/cost taking into account the capacity penalty,
\( \bar{C}_a \): updated vehicle capacity taking into account the number of on-board passengers.

This transformation implies that the iterative solution method is equivalent to the successive shortest path (SSP) method (Ahuja et al. 1993) for solving the MCF problem. However, the difference is in how the network is updated in each iteration. While the residual network is used as the updated network in the standard MCF problem, in the proposed assignment model, available capacity is defined considering the passenger priority, and the network is updated accordingly. The proposed method is significantly more efficient than other assignment methods, such as the Method of Successive Averages (MSA), and results obtained from the case study suggest its efficiency.

The algorithm runs in such a way that in each iteration, the assigned passengers who can complete their trip to the destination are traveling on their optimal path. By looking at the optimality conditions in chapter 3, especially the complementary slackness conditions in (17) and (18), the passengers who can complete their trip successfully are on their optimal path, assuming the deterministic assignment. In this case, the penalty term related to the capacity constraint is zero (\( \sum_a \beta e^{-\alpha p_{ak}} = 0 \)). This is because the path assignment is performed by the TBSP algorithm, taking into account the available capacity of the vehicles. However, the Lagrange multiplier of the capacity constraints (\( \sum_a \nu_{ak}^{rs} \)) may take a positive value in the path cost, which is unavoidable because of the priority level of the passengers (i.e., the shorter paths are taken by passengers with higher priority). Therefore, path optimality in the path assignment is ensured. Moreover, the link travel costs are constant, and with the capacity constraint as the only source of increase in path
costs, the conclusion is that, in each iteration of the algorithm, the completed passenger trips are always optimal, meaning that the equilibrium condition holds:

\[ f_k^{rs} > 0 \rightarrow u_{rs} = c_k^{rs} + \sum_a v_{ak}^{rs} \]  \hspace{1cm} (54)

Now considering the passengers who failed to board because of a capacity limit, their experienced travel cost is very large, modeled by the penalty term \( \sum_a \beta e^{-\alpha p_{ak}^{rs}} \). Note that the penalty term is significantly larger than the other cost components. Also, for the passengers who failed to complete their trips in the last iteration, \( f_k^{rs} = 0 \). The complementary slackness in the KKT conditions implies that the path with no available capacity, and therefore a very large cost, has zero flow. This condition can be shown by:

\[ u_{rs} = c_k^{rs} + \sum_a v_{ak}^{rs} + \sum_a \beta e^{-\alpha p_{ak}^{rs}} \rightarrow f_k^{rs} = 0 \]  \hspace{1cm} (55)

The two conditions (54) and (55) suggest that passengers with successful experience stay on their path unless their experience is affected by other users, and passengers who failed to board try to take the best available remaining path. Through more iterations of the algorithm, more passengers can complete their trips, and finally all the passengers take their optimal available path. This condition is exactly the equilibrium condition, and is the desired solution to the problem. However, in each iteration, the number of unsuccessful passengers represents how far the solution is from the “optimal and feasible” solution.

According to the logic explained above, the percentage of passengers with unsuccessful paths due to capacity violation is used as the measure to test the convergence of the algorithm. In essence, with more completed passenger trips, not only
does the number of passengers assigned to their optimal path increases, but also the amount of the penalty in the objective function decreases and the objective function moves toward its minimum value. As a result, the gap at iteration \( I \) of the algorithm is defined as the fraction of passengers with unsuccessful paths:

\[
g_I = \frac{\sum_{rs} q_{rs}^I}{\sum_{rs} q_{rs}}
\]

(56)

where \( q_{rs}^I \) is the number of unsuccessful passengers in OD pair \( rs \) in iteration \( i \).

It is noteworthy here that the passenger failure-to-board phenomenon in the simulation is modeled as an algorithmic technique to facilitate the solution method. That is, instead of simulating passengers adaptive behavior in a single iteration, those passengers who failed to board are removed from the system in the current iteration, and are added to the system with an updated path in the next iteration. In other words, the path adjustment process is iterative as opposed to being en route. This procedure may overestimate the capacity of the vehicles downstream to where the failure happened, and may affect other passengers’ decisions. But, returning the failed passengers with adjusted path costs in the subsequent iterations will further adjust the passenger flow. There may be other options to treat the path adjustment process, as potential alternatives to the proposed method, but these were not tested in this study. Some of these alternative options are:

- Modifying passengers’ PAT or PDT time windows, so that they can find a service with capacity. This option is suitable if there are passengers who could not complete their trips after several iterations. Departure time choice
or OD matrix calibration are the general forms of adjusting travel decisions with the purpose of finding a better path.

- Assigning the failed-to-board passengers en route to make them complete their trips. Note that the new path assignment has to be done in the middle of the simulation, and should take into account the corrected behavior of passengers when missing a vehicle. Additionally, this method does not necessarily result in the equilibrium solution (or stochastic equilibrium solution) since other paths with lower cost may be available starting from the origin.

Therefore, the proposed method was used as a simple but rational way of treating the failure-to-board experiences. Moreover, the proposed method is computationally efficient and effective in terms of producing behaviorally rational results (explained in the next section). The alternative methods, however, can be considered as potential future work.

The solution method and convergence criterion are applicable in both deterministic and stochastic assignment models by using the TBSP and TBHP, respectively.

5.3. **Numerical Test in San Francisco MUNI Transit Network**

The proposed solution algorithm has been applied to the MUNI network case study in San Francisco, CA, where there are significant capacity problems. In this application, GTFS data were used to build the transit network. A complete description of the GTFS, including the transit stops, routes, and vehicle trips is provided in chapter 4. The other component of the transit network to be generated in the model is the set of walking
(access/egress and transfer) links. The access and egress walking links were generated using a 10 minute threshold (about 0.5 mile) and the transfer walking links are generated using a 5 minute threshold (about 0.25 mile), both in a disaggregate (all-streets) GIS network. It is important to mention that in the San Francisco case, because the transit network is very dense with a small distance between routes and multiple routes serving each stop, the transit network hierarchy (using transfer stops) is not significantly smaller than the original network. Finally, by combining the GTFS data, and the walking links, the trip-based network was built, and the capacity of each vehicle was assigned from other data sources.

The transit demand data (passenger trips) were generated by the San Francisco activity-based travel demand model SF-CHAMP in the form of disaggregate person trips with tour information. The demand was distributed by time-of-day using the observed automated passenger counting (APC) data to capture the dynamic variation of demand within a day. This variation is not only over major time periods (AM, Mid-day, PM, etc.), but also within each time periods. The tour information (e.g. inbound or outbound trip) and the high resolution of the trips made possible a more accurate assignment of passengers to the network. Table 5 shows the network configuration and demand data for modeling the MUNI transit network in the 5-hour PM peak period.

Table 5 San Francisco MUNI Transit Network and Demand in the 5-hr PM peak

<table>
<thead>
<tr>
<th>Network Feature</th>
<th>TAZs</th>
<th>Stops</th>
<th>Routes</th>
<th>Vehicle Trips</th>
<th>Stop-Times</th>
<th>Demand (Person Trips)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>981</td>
<td>3,598</td>
<td>79</td>
<td>2,987</td>
<td>120,143</td>
<td>120,919</td>
</tr>
</tbody>
</table>
The model has been implemented for the 5-hour PM peak period of a typical weekday. A transit utility function estimated in previous studies was used for the route choice model. In the utility function, walking times in different situations such as access, egress, or transfer have different weights, as do the waiting times, relative to in-vehicle time. The dispersion parameter of $\theta$ was also estimated within a given range and different values of $\theta$ were tested in the model.

The transit assignment was performed both deterministically and stochastically on a computer with an Intel Core-i5 2500k CPU and 16GB of memory. The results including the computational times are shown in Table 6. Results show that the algorithm converges efficiently and the computational time is reasonable for the large-scale test network in San Francisco. Moreover, the model did not use more than 300MB of memory during the tests. Figure 15 shows the algorithm convergence in which the gap value (capacity violation) decreases by iterations. In the tests with lower values of $\theta$, the algorithm converges with fewer iterations. Figure 16 also shows the gap value by computation time. In this figure, while the deterministic assignment has better performance in the early iterations (because of the faster path algorithm), it converges to the optimal solution in a relatively similar amount of time. Figure 17 depicts the average travel cost of the passengers (those who arrive to the destination) as it changes by iteration number. This figure shows that by assigning more passengers to the network, the average travel cost per passenger increases. I.e., the solution moves toward a feasible solution while the total travel cost increases, which is compatible with what has been explained before about the
convergence of the assignment algorithm. It is also observed that with a lower value of $\theta$, travelers experience higher travel cost on average.

Table 6 Transit Assignment Results for San Francisco MUNI Network

<table>
<thead>
<tr>
<th>Measure</th>
<th>Deterministic Assignment</th>
<th>Stochastic Assignment $\theta = 1.0$</th>
<th>Stochastic Assignment $\theta = 0.5$</th>
<th>Stochastic Assignment $\theta = 0.2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Iterations</td>
<td>37</td>
<td>24</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td>CPU Time (hr)</td>
<td>15.11</td>
<td>17.8</td>
<td>17.0</td>
<td>13.5</td>
</tr>
<tr>
<td>Average Travel Time (min)</td>
<td>37.07</td>
<td>36.44</td>
<td>39.16</td>
<td>41.84</td>
</tr>
<tr>
<td>Average Travel Cost (min)</td>
<td>59.64</td>
<td>65.04</td>
<td>70.55</td>
<td>78.33</td>
</tr>
<tr>
<td>Average Number of Transfers</td>
<td>0.82</td>
<td>0.78</td>
<td>0.87</td>
<td>1.04</td>
</tr>
</tbody>
</table>

Figure 15 Transit assignment gap by iteration number in San Francisco MUNI network
Figure 16 Transit assignment gap by time in San Francisco MUNI network

Figure 17 Average passenger travel cost by iteration in San Francisco MUNI network
To establish the connection between the mathematical model in chapter 3 and the results obtained in this chapter, we can compare the results with the optimality condition proposed earlier. For the deterministic assignment model, two additional terms \( \sum a \beta e^{-\alpha p_{ak}^r} \) and \( \sum a v_{ak}^r \) are added to the traditional equilibrium model. In fact, the capacity penalty of \( \sum a \beta \) (when \( p_{ak}^r = 0 \)) applies to the passengers in paths with no available capacity and makes their travel cost very large. This is the same as failing to board and not being able to complete the trip using the intended path. This situation happens in the early stages of the algorithm, where there are more passengers who fail to board. The second term \( (\sum a v_{ak}^r) \) applies to the passengers influenced by the capacity constraint who switched their paths in the subsequent iteration. That is, to avoid the very large penalty of not completing the trip, they decide to take a longer path while the additional cost to their trip is equal to the shadow price of capacity \( (\sum a v_{ak}^r) \). Therefore, by performing more iterations of the algorithm, people change their decisions to find a satisfactory path, and more passengers are able to complete their trips with minimal additional cost to their initial path. It is noteworthy that the convergence of the algorithm is based on the same concept, in which passengers with very high travel cost (failed to board) are assigned to new paths (including capacity cost) so that the travel cost in the objective function is reduced (the failing penalty \( \sum a \beta \) is replaced by \( \sum a v_{ak}^r \), when \( \sum a \beta \gg \sum a v_{ak}^r \)). It is intuitive that after most of the passengers are assigned to a path and can complete their trip, the algorithm is close to the optimal solution, and the solution is known as the converged solution. Therefore, it justifies the use of the introduced gap
measure (based on the proportion of completed trips) in section 5.2 and well represents the convergence of the algorithm.

This optimality condition is represented in Figure 17, in which the average travel cost for the arrived passengers increases by iterations. Note that in the graphs, the cost of passengers failing to complete their trips is not included; otherwise, the average cost would be a very large number depending on the value of $\beta$, and decreasing across iterations. Therefore, we can conclude that the difference between the initial average travel cost and the final average travel cost in each test is equal to the average additional cost imposed to passengers because of capacity constraints.

In Figure 17, in the case of deterministic assignment, the final average travel cost is 59.64 min while the initial cost is 54.13 min, meaning that people take paths with 5.51 min (10.18%) more cost only because there is a capacity constraint in the system. This additional cost differs in various OD pairs and will be discussed in the following paragraphs. Looking at the same graph, for the stochastic assignment, the additional cost due to capacity constraint is 7.31%, 6.12%, and 3.57%, for $\theta$ equal to 1.0, 0.5, and 0.2, respectively.

From the perspective of the stochastic assignment optimality condition, similar analyses can be provided. In this case, there is an additional term $\left(\frac{1}{\theta} (\ln f^T_k + 1)\right)$ for the path cost which represents the effect of travel cost perception by transit users. This additional term can be applied in both the capacitated case and the situation without capacity constraints. Looking at Figure 17, the average travel cost in the first iteration of
each test shows a difference in average travel cost which is due to the stochasticity in the model. In this case, an additional average travel cost of 6.48, 12.35, and 21.50 is added to the travelers in the stochastic assignment with $\theta$ equal to 1.0, 0.5, and 0.2, compared with the travel cost in the deterministic assignment (a 12%, 23%, and 40% increase, respectively). Finally, if the effect of both the capacity constraint and stochasticity is taken into account, the average travel cost is increased in the system by 10.91, 16.42, and 24.20 minutes (20%, 27%, and 36% increase) respectively. Each of these values is the sum of $\sum_a v_{ak}^{rs}$ and $\frac{1}{\theta} (\ln f_k^{rs} + 1)$ for different values of $\theta$.

To evaluate the results for various OD pairs, the passengers’ experienced costs were compared in different situations when $\theta = 0.5$. For this purpose, all OD pairs with at least 5 passengers were selected, and all the following evaluations are based on this sample of OD pairs. Figure 18 shows the comparison between the minimum travel cost at iteration zero vs. the minimum travel cost at the converged solution. In other words, the comparison is in the optimal path with and without capacity constraints, showing the effect of the capacity constraint on the travel costs. Compared with the y=x line, the observation is that in some of the OD pairs, the initial minimum-cost path is not available because of the capacity constraint, and people have to use other attractive paths (obviously with additional cost equal to $\sum_a v_{ak}^{rs}$). In most other OD pairs, the minimum-cost path is still available for use.

Another comparison is shown in Figure 19, in which the average travel cost is compared with the minimum travel cost for the same OD pairs. This comparison is made
in a one-shot stochastic assignment, meaning that the capacity constraint is not considered. Therefore, the differences between the average and minimum experienced costs are solely the result of the stochastic assignment, capturing the perception of users in the path costs. The dispersion seen in this figure is because passengers may choose paths that are not the minimum cost, but rather choose paths with the minimum perceived cost. Obviously, the difference in the travel cost is equal to \( \frac{1}{\theta} (ln f_k^{rs} + 1) \).

Finally, the last test is the evaluation of the average travel cost in the final solution (using the full iterative assignment) and comparing it with the minimum-cost paths in the solution with no capacity constraint. It shows how passengers distribute over different paths and, on average, experience higher travel cost, compared with the case where they use the minimum cost path with no capacity constraint. The higher travel cost in this figure is the result of both stochastic assignment and capacity constraints with boarding priority. This comparison is shown in Figure 20, showing more increase in travel cost compared with the previous tests, and that is equal to \( \sum_a v_{ak}^{rs} + \frac{1}{\theta} (ln f_k^{rs} + 1) \).

Figure 21 is also provided for the set of OD pairs in which passengers are affected by both the capacity constraint and stochastic behavior. In this figure, the average travel cost for passengers in each OD pair is shown and is divided to 3 parts: minimum travel time in the OD pair, the additional cost due to various travel cost perceptions (stochasticity), and additional cost due to capacity constraint.
Figure 18 Minimum OD travel cost in the final solution of the proposed model vs. minimum OD travel cost in the one-shot stochastic assignment with no capacity constraints. The differences show the effect of capacity constraints with boarding priority.

Figure 19 Average vs. minimum OD travel cost in the one-shot stochastic assignment without capacity constraint. The differences show the effect of stochastic assignment.
Figure 20 Average OD travel cost in the final solution of the proposed model vs. minimum OD travel cost in the stochastic assignment without capacity constraint. The differences show the effect of stochastic assignment and capacity constraints with boarding priority.

Figure 21 Average travel cost in OD pairs significantly affected by stochasticity and capacity constraints, showing different parts of the travel cost.
5.4. Application in San Francisco Integrated Dynamic Travel Model

The proposed transit assignment model for San Francisco provided the motivation to integrate this model with the current travel demand model, including the activity-based travel demand model and the DTA model. In this section, the proof of concept, implementation steps, and some results are provided for the integrated model of the proposed transit assignment with the DTA model in San Francisco County. Further research with the activity-based travel demand model is ongoing.

5.4.1. Input Data

In application of the transit assignment model in San Francisco, several local data sources provide critical inputs to generate, calibrate, and validate the model. In this section, a brief description of these data sources is provided.

**Automatic Passenger Counts (APC):** Approximately one-third of the San Francisco Municipal Transportation Agency (SFMTA, or simply Muni) bus fleet is equipped with Automatic Passenger Counters (APC) which track boardings and alightings through both front and rear doors. The APC data from the Muni bus network during spring 2012 was used to simulate passengers’ preferred arrival and departure times in five-minute windows (Khani et al. 2013). The APC data were also used to validate the transit assignment at the level of the transit vehicle trip and of the more general time period, for both passenger activity and arrival and departure times. Additionally, Zorn et al. (2012)
used September 2008 Muni APC data to estimate the dwell time as a function of boardings, alightings, and transit vehicle type.

The 2012 APC data set used in this study is based on a 30 percent rotating sample of all transit vehicle trips between March 3, 2012 and June 6, 2012. The APC-equipped buses are assigned to different trips each day. Therefore, each individual transit vehicle trip has approximately one month of APC data recorded in the three-month dataset.

In Zorn et al. (2012), Muni bus dwell times were estimated as a function of the number of boardings and alightings as well as whether the transit vehicle was a standard two-door vehicle, or an articulated three-door vehicle. The best-fit regression model from their study is shown below. The R-squared value is 0.446 for this model and n=182,424 (APC entries), 8% of which represented vehicle loads over the seated capacity.

\[
DWT = 4.90 + 2.44R + 3.72B + 2.11A - 0.71RB - 0.88RA
\]  \hspace{1cm} (57)

where:

- \(DWT\): dwell time,
- \(R\): binary variable indicating whether the bus is articulated,
- \(B\): number of boardings,
- \(A\): number of alightings.

In addition to estimating the dwell time function, APC data was used for assigning preferred arrival and departure times, and to validate the assignment model, which are explained later in this chapter.
**General Transit Feed Specification (GTFS):** The transit network in this study is based on GTFS provided by MUNI. In addition to the details of the GTFS explained in chapter 4, another feature in GTFS is the representation of routes and stops with high geographic resolution, and the capability of matching them with any other geo-coded network. Because transit vehicle trips on each route are coded separately and their information is independent from other transit vehicle trips, each trip can have its own shape file which shows the actual path followed by the vehicle. This property was useful in integrating the roadway and transit networks.

**Individual Travel Patterns:** In order to capture the individual travel characteristics of simulated passengers, disaggregate transit person trips from SF-CHAMP were fed into the transit assignment model. Relevant characteristics include the traveler’s daily activity pattern (activity destinations, times and durations) and the traveler’s value of time, which is higher for mandatory (work, college and work-based purpose) person trips and lower for non-mandatory (grade school, high school and other purpose) person trips.

The day pattern is relevant because it determines the origins and destinations of the transit person trips to be simulated, and indicates whether the traveler has a preferred arrival time (typical for outbound trips) or a preferred departure time (typical for homebound trips). While SF-CHAMP simulates travel demand for the entire Bay Area, the San Francisco DTA model and the transit assignment models only cover the city of San Francisco. Thus the disaggregate trip list was filtered to only those with an origin and destination within San Francisco, which is represented by 981 Traffic Analysis Zones (TAZs). San Francisco is approximately 49 square miles, so this corresponds to a dense
TAZ structure in downtown (usually one block), with TAZs covering a few blocks in less densely developed areas.

The SF-CHAMP model currently has five time periods. In order to more accurately assign transit trips, each trip in the disaggregate demand was assigned a preferred arrival and departure time based on the observed APC distribution of boardings and alightings, respectively. An offset of 5 minutes was included, assuming that alights do not correspond exactly to preferred arrival times at the destination and must account for a short egress walk. Figure 22 shows the typical distribution of average weekday boardings and alightings, using a sample extracted for the first weekdays of April 2012. As expected, the boardings and alightings show typical peaking in the morning and evening, with an obvious lag between passenger boardings and passenger alightings that corresponds to in-vehicle travel times.

![Figure 22 Distribution of passenger boardings and alightings based on APC data](image-url)
5.4.2. Integration with the DTA and Activity-based Model

To capture the impacts of different models on each other, and to model the evolution of travelers’ choices in the dynamic transportation system, an integrated modeling framework with feedback was developed based on the models proposed in chapter 3. The general integrated model framework includes SF-CHAMP as the activity-based travel demand model, a transit and automobile network preparation process, and a multimodal assignment model performing a dynamic traffic assignment (DTA) using Dynameq software and FAST-TrIPs (the model proposed in this study) as the transit assignment model, shown in Figure 23.
Transit Vehicle Simulation in DTA: To improve the transit model’s effectiveness in representing the real world, more realistic departure times of transit vehicles (instead of the schedule) were used for the transit assignment and simulation. For this purpose, the transit vehicles’ trajectories are produced in the DTA model. In order to generate transit vehicle trajectories with the temporal resolution and accuracy of a DTA model, each transit trip in GTFS had to be individually simulated in the DTA model. DTA Anyway (see reference) is an open source tool to facilitate the combination of static networks with
other GIS- and text-based input data sources to generate a network with the richer level of detail required for DTA. Using DTA Anyway, an algorithm (Khani et al. 2012) was developed to translate the GTFS encoding of the Muni schedule into a set of transit vehicle trips for Dynameq. Each transit vehicle trip is represented as a line that only runs once; this is so that when the transit assignment model assigns transit person trips to vehicles and recalculates the dwell time at each stop (based on the boarding and alighting activities at that stop), the next iteration of DTA can use the same transit encoding with updates to the dwell times. This allows a vehicle-trip-specific determination of stop dwell times, since each dwell time represents a single stop for a single transit vehicle trip. This differs from the more conventional representation of a single dwell time for all transit vehicle trips on a single route.

**Dwell Time Feedback to DTA:** While the goal is to model travelers’ behavior and their choices in the most reasonable way, the level of information available to travelers plays a key role and may affect the model outputs. Passengers in San Francisco often refer to either their recent experiences or available real-time data, as opposed to published schedules, when considering path options. Therefore, the realized rather than scheduled transit vehicles’ travel time was used for the transit assignment, which takes into account congestion and delays. Using the realized travel time in the model shows how travelers change their paths after testing different alternatives and may stabilize their path choice over time.

To model the network performance, realized transit vehicle travel times are used from the equilibrated DTA model to update the arrival and departure times for each transit
vehicle trip at each stop. Transit assignment and simulation then determines the passenger flow and level of service, and re-calculates the dwell time at each stop for each transit vehicle. The updated dwell times are fed back to the DTA model to capture the interaction between auto and transit vehicles, modeling possible changes in the equilibrium automobile travel times. The new vehicle travel times are used again in the transit assignment, and this procedure is repeated (Figure 23). After iterating this procedure several times, the dwell times will converge to a stable value and do not change significantly. This procedure is called the multimodal assignment, and the equilibrium state is the multimodal equilibrium solution.

In the multimodal assignment model, the Hessian matrix of the model is an asymmetric matrix, i.e.:

\[ \exists i \neq j, \ a_{ij} \neq 0 \]  \hspace{1cm} (58)

Where \( a_{ij} \) is the element in row \( i \) and column \( j \) of the matrix.

There are two parts for the non-zero off-diagonal elements in the matrix. The first part is the effect of capacity constraints in the transit network, and the second part is the interaction between auto and transit links. The latter is considered for analyzing the convergence of the multimodal model. In the simulation-based assignment, the interaction between auto and transit networks is captured by the transit vehicle dwell times, assuming the auto assignment (with transit vehicle simulation) in the upper level and the transit assignment in the lower level. It implies that the off-diagonal elements of the Hessian matrix are a function of the transit dwell times. While the problem is
diagonalized by fixing the dwell time values in each multimodal assignment iteration, the amount of changes in the dwell times represent the instantaneous effect of the transit system on the auto network. In other words, bigger changes in the dwell time mean larger values for the off-diagonal elements of the Hessian matrix, and therefore, more fluctuation in the results. With more iterations of the multimodal assignment model, the dwell time changes are supposed to decrease toward zero. When the changes become very small, the transit system will have marginal effect on the auto network (link travel times). Consequently, the auto assignment and simulation result in similar transit vehicle trajectories in the next transit assignment. This situation, with negligible values for the off-diagonal elements of the Hessian matrix,

\[ a_{ij} = 0; \quad \forall i \neq j \] (59)

and therefore symmetric link travel costs both in auto and transit, is considered a multimodal equilibrium state. This is curious, considering the fact that auto and transit equilibrium states are reached in the separate models (with symmetric link travel costs). This solution is the desired (near-optimal) solution to the multimodal assignment model, and the changes in the dwell times represents how far the current solution is from the optimal solution.

According to the discussion, the convergence of the multimodal assignment model is evaluated by the gap value defined by the changes in the transit dwell times. The dwell time gap function measured at each iteration of the multimodal assignment is defined as follows:
\[ g_d^i = \frac{\sum_{rs} (dt_{rs}^i dt_{rs}^{i-1})}{\sum_{rs} dt_{rs}^{i-1}} \]  

(60)

where:

\[ dt_{rs}^i : \text{average dwell time of transit route } r \text{ at stop } s, \text{ in the big iteration } i.\]

The dwell time gap in this application is aggregated, as the dwell time at each stop is calculated for each route and not for each vehicle trip. That is, the dwell time at each stop, for each route, is averaged over transit vehicle trips and used in the gap function. The reason for this aggregation is that some of the routes in San Francisco have short headway and people may not consider them as different services, and instead they will take any of the trips departing in a certain time window. However, they make their path decision based on the route and the boarding stop. Therefore, passengers may change their decision regarding the specific vehicle trip across the iterations of the model, especially because of travel time variation across iterations that occurs from the DTA model. It results in significant change in the dwell time, and a higher value for the dwell time gap if it is measured at the trip level. On the other hand, since passengers still take the same route at the same stop, the route-level dwell time gap is still a good measure representing the equilibrium in passengers’ decision-making.

In the iterative process, as soon as the dwell time gap \( g_d^i \) takes a value below a threshold \( \delta \), the algorithm stops. Alternately, the iterative process may reach a maximum number of iterations while the gap value is still not below the threshold. In either case, the algorithm terminates with multimodal equilibrium solution (as explained earlier in this section). The result of the transit assignment in the last iteration is the transit
equilibrium in the multimodal equilibrium state, and the results such as vehicle loads and travel costs can be used for planning purposes or for feedback to the demand model.

**Feedback to the Activity-Based Model:** The final component of the integrated model framework is to provide a realistic transit assignment for use in planning. In order to do this, the transit network model must provide a realistic level of service for the demand model based on an accurate representation of user experience and reasonable physical capacity constraints. When run iteratively with FAST-TrIPs, travelers modeled within SF-CHAMP will be able to compare their available travel options and adjust their decisions (e.g. destination, time of day, mode, etc.). This study provided the mechanism for providing the transit level of service information to the demand model through the skim tables (the skim tables provide such measures as average wait time, in-vehicle time, transfer time, access/egress time, fare, etc for each time period and even for each transit sub-mode).

### 5.4.3. Results and Validation

**Integrated Model Results:** The Dynameq-FAST-TrIPs integrated model is able to run for the present-day San Francisco County network, including the road network and Muni transit network, for the PM period (2:30pm to 9:30pm which contains warm up and cool down times). The network contains 981 zones, 14,654 nodes, and 37,059 links in the road network, and 3,598 stops, 79 routes, and 3,527 transit-vehicle trips in this PM period. On the demand side, SF-CHAMP simulates more than 100,000 travelers who make about 104,000 tours with at least one walk-to-local bus or walk-to-light rail trip within San Francisco in the PM period. This results in about 121,000 transit person
journeys. Among these transit journeys, according to the mode choice sub-model in SF-CHAMP, 85,665 journeys are completed by Muni buses and streetcars.

Based on empirical estimates from a previous study (Zorn et al. 2012), the following utility function for transit paths has been determined for the assignment:

\[ U_i = T_i + 2.23W_i + 36.58D_{1i}^o + 107.80D_{1i}^d + 147.96D_{1i}^x + 7.76X_i \]  \hspace{1cm} (61)

where:

- \( T_i \): total in-vehicle time,
- \( W_i \): total waiting time,
- \( D_{1i}^o \): walking distance at origin (access),
- \( D_{1i}^d \): walking distance at destination (egress),
- \( D_{1i}^x \): walking distance at transfer points,
- \( X_i \): number of transfers,

Note that the walking distances (from origin, to destination, at transfer points) are measured in miles. The logit dispersion parameter is set to be 0.5 as a default value, according to the level of knowledge and information of the users from the transit service in San Francisco.

The stochastic transit assignment model has been prepared (for the Muni network only) for integration with the DTA model. The integrated model was run in 3 big iterations, including 3 rounds of the DTA and 3 rounds of the transit model. The DTA itself uses 70 iterations to reach the equilibrium, and it takes about 60 hours to complete.
The stochastic transit assignment takes 8 iterations to reach the SUE for transit users, and it takes about 3.5 hour to converge with 1% gap. That is, the algorithm is assumed to be converged when 99 percent of the transit users are assigned to their best available paths.

As explained before, the dwell time in the disaggregate level is calculated and is fed back to the DTA model in each big iteration. The initial dwell time is set to be 30 sec. At the end of the last big iteration, the transit skim is generated and is prepared for feedback to the SF-CHAMP travel demand model. Table 7 shows the aggregate results of the last transit assignment in the integrated model. It shows that, during the third big iteration, the transit capacity violations have decreased from 13.5 to 0.9 percent of all passenger journeys. Figure 24 shows the convergence of the transit assignment model by iterations.

After the last big iteration, the dwell time gap is 19.3 percent. Note that the dwell time gap is measured at the route level, meaning that equilibrium in passengers’ decisions is evaluated regarding the passengers’ boarding/alighting location and the direction they go along the route. Some other measures were also tested and evaluated (e.g. aggregation over the stops, time periods, etc), but the logic implied that proposed measure is the most realistic measure. The effect of the input to the model from the DTA is also important, since vehicle travel times change significantly by iteration and these influence passengers’ decisions severely. The effect of changes in transit vehicles’ travel time is significant, as Figure 27 indicates that there are 20% and 8% travel time change in the DTA model output, during the 2\textsuperscript{nd} and 3\textsuperscript{rd} big iterations respectively. I.e., the 19.3 percent dwell time gap is not solely because of the variation in the passengers’ path choice decision in the transit assignment, but is the effect of variable inputs to the transit
assignment model. However, the stochastic transit assignment may inflate this effect, resulting in higher dwell time gap.

Another analysis shows that 37% of the change in dwell time in the last big iteration is due to the fact that passengers change their boarding or alighting stops in the same transit trip. Also, 38% of the change is the result of passengers changing their transit vehicle trip in the same route, and the last 35% is the result of changing the transit route. With this evidence, one may conclude that the dwell time gap calculated in this application is reasonable, implying that the transit assignment model is robust and produces stable results.

The average dwell time at each stop after the last iteration is 7.7 sec, while the average dwell time for the vehicle-stops with non-zero dwell times is 17.8 sec. This is for the “hail-stop” operation of the transit vehicles. In the converged solution, the average travel time of transit users is 30.1 minutes including access and egress time, and the average number of transfers is 0.75 transfers per passenger. Figure 26 shows the distribution of the trips with the number of transfers and travel time.
Table 7 Transit Assignment Results with Simulated Travel Times for Transit Vehicles

<table>
<thead>
<tr>
<th>Measure</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Big Iterations</td>
<td>3</td>
</tr>
<tr>
<td>Number of FAST-TrIPs Iterations per Big Iteration</td>
<td>8</td>
</tr>
<tr>
<td>Dwell Time Gap (%)</td>
<td>19.3</td>
</tr>
<tr>
<td>CPU Time (min)</td>
<td>222.7</td>
</tr>
<tr>
<td>Transit Demand Size (passenger trips)</td>
<td>85,665</td>
</tr>
<tr>
<td>Initial Capacity Violation (%)</td>
<td>13.5</td>
</tr>
<tr>
<td>Capacity Violation at Convergence (%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Average Travel Time (min)</td>
<td>30.1</td>
</tr>
<tr>
<td>Average Number of Transfers</td>
<td>0.75</td>
</tr>
<tr>
<td>Average Dwell Time (sec)</td>
<td>7.7</td>
</tr>
<tr>
<td>Average Non-Zero Dwell Time (sec)</td>
<td>17.8</td>
</tr>
</tbody>
</table>

Figure 24 Capacity violation reductions in the iterative transit assignment
Figure 25 Dwell time gap convergence in the multimodal assignment
Figure 26 Distribution of the transit trips with a) number of transfers b) travel time
Validation of the Results using APC Data: Although the model calibration is not fully completed, and some of the model’s parameters have been set based on previous studies or estimations, the results of the transit assignment have been compared with the available data to test the validity of the model. Among the available data, the passenger counts with vehicle location (APC/VL) give us the best information for this validation procedure. It is noteworthy that the travel times of the transit vehicles from the DTA simulation have to be validated first as an input to the transit assignment model. For this reason, the transit vehicle travel times from these three sources of data, including the GTFS schedule, the recorded vehicle location data, and the DTA model are compared in Figure 27. It is clear that, for the refined sample data, observed travel times from the APC data are compatible with the GTFS schedule since the $R^2$ is more than 0.86 (Figure 27a). In Figure 27b though, the travel times from DTA are significantly lower than the GTFS data, and therefore lower than the APC data. However, the correlation between the two data sets is relatively high with 0.79 for the $R^2$. This may be an indication that more effort should be spent on the calibration of the DTA model as it underestimates the transit vehicles’ travel time. Other attributes may be affecting the shorter travel times, including pedestrian interaction with vehicle turning movements, bus-auto interaction, and additional dwell time factors.
(a) APC Travel Time vs. GTFS Travel Time

\[ y = 1.055x \]

\[ R^2 = 0.861 \]

(b) DTA Travel Time vs. GTFS Travel Time

\[ y = 0.708x \]

\[ R^2 = 0.789 \]
For validation of the transit assignment, the number of boarding and alighting passengers and transit vehicle loads from the assignment model can be compared with the observed APC data. This comparison can be done either at a disaggregate level for every vehicle trip or at an aggregate level for each route. Only sample trip-level comparisons are shown here. Figure 28 shows the load profile of two sample vehicles on route 38-Geary Boulevard, which is one of the busiest routes in the San Francisco transit network. The model results match well at some locations, while at some points in the route, a difference is seen between the model results and the APC data. In general, the overall predicted travel pattern is good, as it shows the peak loads consistent with the observations.
Figure 28 Sample load profile in inbound vehicle trips of route 38 – Geary) Westbound vehicle departing at 4:37PM b) Westbound vehicle departing at 5:42PM
Another type of validation is based on aggregate measures in each route to be compared with the observation data. These measures are usually in the form of ridership (how many passengers use each route), and is done usually in regional travel models. Therefore, we compared, for each route, three measures calculated in the model with the values calculated from the APC data in one sample weekday. Figure 29 shows the average number of on-board passengers (load) on the vehicles in each route. In other words, this measure shows on average how many passengers are on the vehicles in each route during the operation. The comparison shows that the model reported significantly lower values. However, many of the points in the figure follow high correlation with the observed data, and the low slope of the regression line can be the result of the outlier points located in the lower side of the graph. It is likely that the outlier points are associated with the routes that are not coded properly, and passengers do not consider them as attractive routes.

In another validation test, the average ridership on vehicle trips on each route is calculated and compared with the APC data (Figure 30). The graph shows almost the same pattern observed in Figure 29, meaning that the ridership is underestimated in the model. By looking at the two figures, one can interpret that the model works better in estimating the ridership than the vehicle load, which is affected by both boarding and trip length.

The last test is to compare the average of the total dwell time over vehicles in each route with the observed values in AVL data. This comparison is depicted in Figure 31, showing almost similar trend as observed in the last two graphs. The value of dwell time
is highly correlated with the ridership, and one can interpret that the dwell time function underestimated the dwell time value, while well-estimating the pattern.

In general, the model seems to underestimate the route-level measures, but predicts the travel pattern close to the observations. These results are promising for a new model, as the calibration process is still ongoing and more importantly, the inputs to the model may not be completely realistic. In other words, the travel demand sub-models in the activity-based model (including mode choice and departure-time choice) may require more calibration to represent the real situation. Moreover, the DTA model is not fully developed, and further steps in model validation and application can improve the results.

Figure 29 Average number of on-board passengers in different routes
Figure 30 Average ridership (number of boardings) in different routes

Figure 31 Average of transit vehicle’s total dwell times in different routes
CHAPTER 6: MODELING INTERMODAL TOURS IN A DYNAMIC MULTIMODAL NETWORK

6.1. Intermodal Assignment Models in the Literature

As one of the first studies in the area, Abdulaal and LeBlanc (1979) introduced a discussion on intermodal travel modeling. They introduced two ways to combine mode choice and route choice models. In one approach, mode choice and route choice are done sequentially, and in the other, they are done simultaneously. Fernandez et al. (1994) provided a discussion on trip planning with combined modes. In their study, they mentioned two major issues in multi-modal transportation modeling: (a) how users choose their mode of the trip, and then, depending on the answer to the first question, how the best route is chosen; and (b) how the transfer point from the private to the public mode is selected. By this discussion, they proposed three approaches to model intermodal trips:

1. Using the generalized cost of the combined mode, people choose their path based on Wardrop’s principle of optimality to minimize the cost of their trip.

2. The mode of the trip is determined using a mode choice model in which the combined mode is considered as a pure mode, and then the shortest path is found in the selected mode.

3. An extension of the second model by including the choice of transfer point as the sub-model.
While they proposed the third model as a new approach in the integration of demand and network models in an analytical framework, the network is assumed to be static.

Modesti and Sciomachen (1998) proposed an algorithm for finding a multi-objective shortest path in a multimodal transportation network. They introduced a utility function for weighting the links based on their cost and time and used the classical Dijkstra’s shortest path algorithm to find the path with maximum utility. Ziliaskopoulos and Wardell (2000) developed an algorithm for finding the intermodal least time path in multimodal networks with time-dependent link travel times and turning delays. Their label-correcting algorithm is designed for all time intervals, and its complexity is $O(T^3V^5)$, independent of the number of modes, where $T$ is number of time intervals and $V$ is the number of nodes. They claimed that, based on computational experiments, the algorithm has a practical computational time that is linear with the number of nodes and time intervals. Abdelghany (2001) proposed a dynamic assignment and simulation framework for different modes of transportation. Their model is the incorporation of a multi-objective time-dependent shortest path and DYNASMART as the simulation model.

Lozano and Storchi (2002) also applied a label-correcting algorithm to find the shortest viable hyperpath with a predefined maximum number of modal transfers. The approach is useful when there is no exact schedule for the transit system (i.e., the transit network is frequency-based). Because it considers more than one criterion, the result of the algorithm is not necessarily optimal, and the user can choose the best hyperpath(s) among the output, according to their preferences.
A multi-modal assignment formulation was proposed by Garcia and Marin (2005) in the form of variational inequalities considering the combined modes. They used a nested logit model as the equilibrium model, capturing the choice of mode and transfer point between modes as well as routes. Compared with Fernandez et al. (1994), they formulated the problem in a hyperpath space and performed stochastic assignment with elastic demand. Zhou et al. (2008) developed an integrated framework to model choices of departure time, mode and path in a multimodal transportation system. As a part the model, a time-dependent least-cost path algorithm based on Ziliaskopoulos and Wardell (2000) was used to generate intermodal paths. For this algorithm, a set of constraints for possible mode transfer was applied. Khani et al. (2012a) and Nassir et al. (2012) provide recent studies on intermodal path and tour problems taking into account scheduled service for the transit network.

By reviewing the most important studies in this area, we believe that there may be a more efficient and flexible algorithm for intermodal path generation which can be used in the assignment model. Moreover, the dynamics of the multimodal system (interaction between auto and transit systems), the transit schedule, and transit capacity constraints, are the motivations to pursue a more advanced intermodal assignment model based on simulation. Therefore, algorithms that are more efficient computationally and that are appropriate for modeling intermodal trips and tours are developed. The combination of the path models with a dynamic multimodal simulation model and a DTA model are also proposed to facilitate a comprehensive multimodal transportation network model.
6.2. Intermodal Shortest Path Algorithm

6.2.1. Modeling the Intermodal Transportation Network at Park-and-Rides

To model a multimodal transportation network, the complete transport chain (e.g., a set of intermodal paths) should be represented accordingly in the model, especially for the transfer between modes (Figure 32). The method proposed here moves toward a more realistic behavioral representation of the traveler’s path in the vicinity of each park-and-ride, considering the accessibility and movement of travelers and vehicles. Although the output simply shows the travel times from a set of intermodal transfer paths, this method explicitly captures the interaction between the auto and the transit network; i.e., explicitly capturing access to park-and-ride by auto, parking the car, and walking to the transit stop.

Figure 32 Framework of the intermodal network in park-and-ride facilities
To facilitate this interaction, aerial-photo-based access points (AP) are introduced for each park-and-ride. These access points represent driveways from (to) the road network to (from) the park-and-ride lot. Each AP can be placed in the auto network as a node along the road network. The distance from/to a street junction (SJ) (e.g., intersection or nearest node) can be easily applied.

To model the transfers explicitly, all the access points (AP) in the auto network which are on the perimeter of the park-and-ride lot should be connected to all the transit stops within the park-and-ride. This is achieved by connecting the AP to the centroid of the park-and-ride lot, as an auto mode link, to complete the auto part of the mode transfer. Then, by adding walking links from the centroid of the park-and-ride to the transit stops or stations in the transit network, an intermodal transportation network can be established.

In the same way, in order to estimate travel times at each stage, we can enumerate the set of intermodal transfer paths. These paths enhance the representation of travel times (costs) and transfer behaviors in the vicinity of a park-and-ride and can be more practical in modeling intermodal travel. Figure 33 illustrates an intermodal transportation network, consisting of different modal networks. Within these networks, a traveler can move from an AP in the auto network to an explicit transit stop location in the transit network.

This disaggregate approach of developing intermodal paths can be useful to incorporate any additional information (e.g., congestion) into the associated stage. For example, the generalized cost of the mode transfer links may include a penalty representing the delay in finding a parking spot. Transfer cost (from auto to transit) may
include in-vehicle time from a surrounding node, to an access point to an available parking space, walking to the transit stop, and waiting time for the transit vehicle. In our study, driving speeds in the park-and-ride lot are 10 mph, and walking speeds are 3 mph.

In addition to the typical values for each part of the transfer in the disaggregate intermodal network, additional delay may be considered. The main sources of this additional delay are vehicle deceleration before accessing an AP, time required for entering the lot (e.g., taking a ticket or paying a fee if applicable), finding a parking space (in congested parking lots), and parking the car.

Figure 33 Intermodal network, including different layers of transportation modes, connection between nodes and transit stops, at Sunrise Park-and-Ride, Rancho Cordova, CA
6.2.2. Intermodal Optimal Path Algorithm

In a multimodal transportation network with $P$ as the number of park-and-ride locations, choosing the optimal point to transfer from auto to transit is similar to a multi-destination choice problem, and its complexity increases linearly with the number of choices. The problem is defined as finding the optimal path between an origin and a destination with a preferred arrival time (PAT) at the destination, considering all transportation modes (see Figure 34). If the park-and-ride points are defined as the alternatives for an auxiliary destination, and calling the true destination as the “main destination,” the model determines the best choice among the alternatives for the auxiliary destinations considering the whole trip from origin to the main destination.

![Diagram of intermodal trips](image)

Figure 34 A set of intermodal trips to a destination through park-and-ride facilities

The structure of the model is shown in Figure 35. The main input data is GTFS data for the transit network, an auto network including node and link geometry and properties, time-dependent link travel times, and aerial photos of the park-and-rides as well as the detailed roadway network in GIS format. In addition, using the model introduced in the previous section, access and egress links between nodes in the auto network and the
transit stops, and also mode change links at park-and-ride locations, are generated in advance for the algorithm.

Figure 35: The structure of the intermodal optimal path algorithm

To meet the PAT at the destination, and more importantly, to ensure the use of transit, a backward transit shortest path (TBSP) is run from the destination at the PAT, and all the stops are labeled accordingly if there is a service available for them to reach the destination. The main input files used in this module is the transit schedule in GTFS format. In detail, first the main destination node, which is in the auto network, is selected and the transit stops within walking distance from this destination are labeled, using travel times on the access links. Then, the transit path is found from these labeled stops to
all other stops in the transit network. After this step, the access to transit stops from the auto network is established. At this point, the label of each node is the travel cost from that node to the destination, using transit only.

The connection between the auto and transit network is then established using the mode change (P&R) links that were described in the previous section. I.e., from the transit stops in the park-and-rides (i.e. the auxiliary destinations), the adjacent nodes in the auto network are labeled using the mode transfer cost.

On the auto side, instead of finding the shortest path from each park-and-ride to the origin node separately, we utilize a multi-source (Klein 2005) time-dependent shortest path (MTDSP) algorithm, which is a label-correcting algorithm that takes more than one node as the starting point of the algorithm. The nodes in the auto network at the park-and-ride facilities keep their labels and are added to the scan-eligible (SE) list in the MTDSP. By running the MTDSP, the best transfer location (i.e. park-and-ride) from the origin is found, considering the cost for the transit part of the trip. In other words, the initialization step of the MTDSP algorithm is as below:

1. Add the destination with label zero to the SE list. This can be an optional step for the case that the optimal path is sought, while for a bi-modal path, this step can be skipped.

2. Set the label of all the nodes to infinity and the predecessor nodes to null, except those nodes which have a label and predecessor from the execution of the TBSP in the transit network.
3. Add nodes updated from mode change links (AP nodes from the park-and-ride) to the scan eligible list.

This technique, i.e. adding multiple nodes to the scan-eligible list at the beginning of the algorithm, introduced by Klein (2005), has the advantage of finding the best source (park-and-ride) node to the target (the origin) by automatically comparing the travel cost from the destination to each source node. Because the initial label of the source nodes are set by the TBSP algorithm, the MTDSP also takes into account the transit travel cost to choose the best park-and-ride location. In the conventional approach, a time-dependent shortest path must be found for each source node to the origin, which can be a time-consuming procedure when there are many park-and-ride facilities in the network.

For the implementation of the algorithm, the auto network is defined by a graph $G_A(N, L)$ where $N$ is the set of nodes, indexed by $n$, and $L$ is the set of links, indexed by $l$. The transit network is defined by a graph $G_T(S, R)$ where $S$ is the set of stops, indexed by $s$, and $R$ is the set of routes, indexed by $r$. Each route contains a set $TR$ which is the set of trips, $tr$. Each trip also contains a subset of stops to serve as well as the schedule time for each stop. There is also a set of transfer links $tf(s_1, s_2, w_{tf}) \in TF$ which include the walking time between a pair of transfer stops, indicated by $w_{tf}$. The access links $a(n, s, t_a) \in A$ connect a node, $n$, and a stop, $s$, with the walking time, $t_a$. The mode change links $m(n, s, t_m) \in M$ also have the same format as the access links, while the stops are the transit stops at park-and-rides, and the travel time is the intermodal transfer time. The input of the algorithm is the passenger’s origin, $O \in N$, destination, $D \in N$, and preferred
arrival time to the destination, \( \text{PAT} \). The intermodal shortest path algorithm is shown in Figure 36:

**Step 0- Initialization:**
- Set \( \text{Label}(n) = \infty \), \( \text{Pre}(n) = \text{Null} \), and \( \text{Mode}(n) = \text{Null} \), for \( n \in N \),
- Set \( \text{Label}(s) = \infty \), \( \text{Pre}(s) = \text{Null} \), \( \text{Trip}(S) = \text{Null} \), and \( \text{Mode}(n) = \text{Null} \), for \( s \in S \),
- Get \( O, D \), and \( \text{PAT} \), and set \( \text{Label}(D) = \text{PAT} \).

**Step 1- Egress to the Destination from Transit by Walking:**
- For all \( a \in A \) with \( n_a = D \):
  - \( \text{Label}(s_a) = \text{PAT} - t_a \), \( \text{Pre}(s_a) = D \), \( \text{Trip}(s_a) = W \), \( \text{Mode}(s_a) = T \), and add \( s_a \) to SE list.

**Step 2- Transit Shortest Path:**
- Run the TBSP backward from nodes in the SE list to the source nodes. In this step, for any \( s \in S \), if \( \text{Label}(s) \) is updated, set \( \text{Mode}(s) = T \).

**Step 3- Access to Transit by Walking:**
- For all \( a \in A \), if \( \text{Label}(s_a) \neq \infty \):
  - \( \text{Label}(n_a) = \text{Label}(s_a) - t_a \), \( \text{Pre}(n_a) = s_a \), and \( \text{Mode}(n_a) = T \).

**Step 4- Access to the Destination by Auto:**
- Add the destination node \( D \) with \( \text{Label}(D) = 0 \) and \( \text{Pre}(D) = \text{Null} \) to SE list.

**Step 5- Access to Transit by Auto:**
- For all \( m \in M \), if \( \text{Label}(s_m) \neq \infty \):
  - \( \text{Label}(n_m) = \text{Label}(s_m) - t_m \), \( \text{Pre}(n_m) = s_m \), and \( \text{Mode}(n_m) = T \), and add \( n_m \) to SE list.

**Step 6- Auto Shortest Path:**
- Run the MTDSP backward from the nodes in the SE list to the origin. In this step, for any \( n \in N \), if \( \text{Label}(n) \) is updated, set \( \text{Mode}(n) = A \).

**Step 7- Extract the Path:**
- Trace the path starting from the origin, \( O \), using \( \text{Pre}(n) \) and \( \text{Mode}(n) \).

---

**Figure 36 Intermodal shortest path algorithm**

In the case that an optimal intermodal path (rather than a shortest path) is required, a label for the generalized cost is used and a weighting system \((f_W, f_A, f_T, f_M, p_m)\) is applied to the times in each part of the path. The parameters are defined as:

- \( f_A \): the weight of travel time by auto, used in MTDSP.
- \( f_T \): the weight of travel time by transit, used in TBSP.
- \( f_W \): the weight of walking time, applied to access links and transfer links.
- \( f_M \): the weight of mode transfer times, used in step 4.
- \( p_m \): the additional penalty cost for the mode transfer.

The complexity of the proposed algorithm is \( O(S^2 + P + N^2) \) where \( S \) is the number of stops, \( P \) is the number of park-and-ride locations (source nodes), and \( N \) is the number of nodes in the auto network. In general, if \( S \) and \( N \) are comparable, the complexity of the algorithm is dominated by the transit side. In the typical schedule-base transit shortest path algorithm, the complexity is \( O(S^2) \), but in the TBSP, as explained earlier, the number of iterations are decreased by taking the advantage of the transfer stop hierarchy. Therefore, the average run time lower depending on the number of transfer stops in the network. For finding all-to-all shortest intermodal paths in all time intervals \( T \), the whole computational effort will be \( O(TNS^2) \) which is better than existing intermodal path methods in the literature, e.g. Ziliaskopoulos and Wardell (2000).

Since the complexity of the algorithm is not dependent on the number of park-and-ride facilities, the procedure can be used for modeling kiss-and-ride trips. In kiss-and-ride trips, the passenger is dropped off at a transit stop using a shared ride and does not have a car to be parked. In other words, in kiss-and-ride trips, access to transit can be made at any stop. These trips can be properly modeled by the proposed algorithm by relaxing the mode change point to every transit stop. The required modification in the proposed algorithm to model kiss-and-ride trips is in step 3, which connects the transit and auto network using the access links as below:

- For all \( a \in A \), if \( Label(s_a) \neq \infty \):
  
  For all \( a \in A \), if \( Label(s_a) \neq \infty \):
- \( \text{Label}(n_a) = \text{Label}(s_a) - t_a, \) \( \text{Pre}(n_a) = s_a, \) and \( \text{Mode}(n_a) = T \) and add \( n_a \) to SE list.

It should be mentioned that the proposed approach compares all the paths in all combination of modes for the optimal path and does not guarantee that both auto and transit modes are used. This outcome seems intuitively rational, as it results in a single-mode path in the extreme cases where either auto-only or transit-only paths dominate the bimodal paths. The examples are the cases that the origin is directly connected to the destination by a transit route (e.g. they are relatively close) and using auto is not required to access to transit stops; or, in the other extreme, there is no attractive transit service or park-and-ride location on the way from the origin to the destination and commuting by auto is faster/more cost effective than changing the travel mode to transit. Finally, the cases in which a bimodal path is required can be modeled using a slight modification of the algorithm, but again, this may not necessarily result in the optimal intermodal path.

### 6.3. Modeling Park-n-Ride Choice in the Intermodal Tours

The intermodal shortest path introduced in section 6.2 is appropriate for the one-way trips and results in the optimal path in the main destination-bound trips only. However, in the planning applications, especially in the context of tour-based and activity-based demand forecasting models, the daily travel is modeled in the form of complete tours from home, to the main destination (e.g. work, school), to potential secondary destinations (e.g. grocery shopping, restaurant, recreation, etc) and finally to home. It implies that, while the demand (the set of destinations in the given timeframe) is modeled at the tour level, the network model has to be compatible with the demand models. More
particularly, when intermodal travel is modeled, it is very important to take into account the park-and-ride location and the constraint in returning to the same park-and-ride as what has been used in the first half of the tour. That is, people have to pick up their cars from where they have parked it and neglecting this constraint may result in inaccurate modeling results. To deal with this problem, the intermodal shortest path algorithm is extended to an intermodal shortest tour algorithm and the dynamic assignment of intermodal tours. In this section, the assignment of the optimal park-and-ride location to the tours is introduced, and the dynamic simulation of the intermodal tours is explained in section 6.4.

The path model for the intermodal tour is based in the same model elements introduced so far: the TBSP, park-and-ride model, and the MTDSP. The difference is that instead of using the origin and destination with PAT as the input to the algorithm, a complete tour is processed as the input, and the optimal park-and-ride is found with the minimum travel cost for the tour. In general, if a park-and-ride is optimal in the first trip, there is no guarantee that it has a reasonable cost in the later trip and that it results in the minimum tour travel time, especially in a dynamic network. Therefore, an algorithm is developed to enumerate all the possible park-and-rides and to find the one with minimum tour cost. Obviously, the enumeration procedure has linearly higher complexity as the result of higher number of park-and-rides. The complexity of the problem will be considerably higher when there are multiple destinations in the tour, so that it becomes a very challenging problem. There have been some efforts in the past to model this problem (Nassir et al. 2012), but the efficiency of the solution algorithms is still an issue.
The proposed algorithm in this study uses some simplifying but logical assumptions and makes the problem much easier to solve while keeping the results as good as the original problem. The assumptions are listed below:

1. There is one primary destination in each tour, which is considered as the “main” activity. This main activity with the given start time and duration, along with the home activity, define the time window for the inbound and outbound trips.

2. The intermodal trips are used for the main destination, and the secondary destinations are visited by either auto or transit. More particularly, such activities as grocery shopping on the way home, pick-up/drop-off, or recreational activities are done by auto mode, and secondary tours such as dining (while at work) can be done by walk or transit.

Therefore, the input tours are reviewed in the preprocessing step and the potential intermodal trips of the tour are selected to make a new chain of trips for park-and-ride assignment. The resulting (simplified) tour contains the PDT from the origin and PAT to the destination in the first half-tour, and PDT from the destination and PAT to the origin in the second half-tour. A schematic simplified tour as the input to the algorithm is shown in Figure 37.
The park-and-ride assignment model is a combination of two intermodal shortest path algorithms from the trips level, with some modifications to account for the constraints introduced in the tours. The algorithm aims to find the cost associated with each park-and-ride in the first half-tour while removing the infeasible park-and-rides from the choice set according to the available time-window. I.e., if a path through a specific park-and-ride is so long that its travel time exceeds the available time window (e.g. PAT-PDT), the park-and-ride is removed from the choice set. By doing the similar procedure for the second half-tour, and combining the results, the feasible park-and-rides will be sorted in terms of the cost and the most appropriate park-and-ride can be assigned to the passenger tour. The assignment can be done using any choice model such as the logit model, since the choice set and the cost (i.e. utility) of each choice is determined.
However, the optimal park-and-ride (a deterministic choice based on minimum tour cost) is selected and assigned to the passenger in this study. The steps of the algorithm are:

1. The forward auto shortest path tree is found at PAT from the origin $T_1$, and the label $l_1$ is set for each park-and-ride showing the auto travel cost.

2. The backward transit shortest path tree is found at PAT to the destination $T_2$, and the label $l_2$ is set for each park-and-ride showing the transit travel cost.

3. $C_o = l_1 + l_2$ is calculated as the travel cost from the origin to the destination (initial trip) through each park-and-ride, if the intermodal trip is feasible according to the time window $T_2 - T_1$.

4. The forward transit shortest path tree is found at PDT from the destination $T_3$, and the label $l_3$ is set for each park-and-ride showing the transit travel cost.

5. The backward auto shortest path tree is found at PAT to the origin $T_4$, and the label $l_4$ is set for each park-and-ride showing the auto travel cost.

6. $C_i = l_3 + l_4$ is the travel cost from the destination to the origin (inbound trip) through each park-and-ride, if the intermodal travel is feasible according to the time window $T_4 - T_3$.

7. The sum of the intermodal travel costs, $C_t = C_o + C_i$, is the total tour travel cost through each park-and-ride. Therefore, the optimal park-and-ride with lowest value of $C_t$ is found and assigned to the tour.
The result of the intermodal tour assignment algorithm is the optimal park-and-ride, so that the auto and transit parts of the tour are determined. In other words, the intermodal tour is broken into two auto trips and two transit trips, each with either new origin or new destination, and associated PDT or PAT. The proposed model is a combination of four shortest path algorithms, and its complexity is \( 2O(N^2) + 2O(S^2) + O(P) \) where \( P \) is the number of park-and-rides. In fact, because of the greater complexity of the transit networks, the overall complexity of the algorithm is \( O(S^2) \). Considering that the intermodal tour problem is a challenging problem, the proposed model is very efficient without many simplifying assumptions.

Although the auto path is found in a network with time-dependent link travel times, it can be fed into a DTA model for more appropriated assignment. A similar assignment may occur for the transit trips, resulting in a more accurate representation of the transit user behavior. Nonetheless, the proposed algorithm is an appropriate model for the whole tour assignment and the results are used in a simulation model that is explained in the next section.

6.4. Dynamic Simulation of Transit and Intermodal Tours

The results of the intermodal tour assignment are used in a simulation model to evaluate the experience of the passengers in a high-resolution dynamic multimodal network. The simulation model consists of a DTA model (DynusT in this study) and the passenger simulation model introduced in chapter 5. In fact, the auto trips of the intermodal tours are given to the DTA model, and the transit trips are given to the
passenger simulation model (the FAST-TrIPs model), and an interface was developed to pass the information between the models.

Starting with the park-and-ride assignment, the next step of the model is to simulate the auto trips in the DTA model, so that all auto trips (SOV, HOV, trucks, etc) and the auto parts of the intermodal tours are assigned iteratively in a congested network. In each iteration of the assignment, the vehicles are loaded into the highway network and the path adjustments happen iteratively. The goal is to assign the vehicles to the most appropriate (dynamic user equilibrium) path according to the traffic congestion. This process is in the scope of the DTA models and more information can be found in the related references.

The part that is important for the intermodal model is the simulation of the drive-to-transit trips, so that the actual arrival time to the park-and-ride is estimated in the congested network. This information is critical in the passenger assignment model since the transit path assignment is dependent on the arrival time of the passenger at the park-and-ride location.

The third step of the model is to reassign a proper transit path to the intermodal passengers according to their actual arrival time to the park-and-ride location. A path algorithm similar to the algorithms proposed in chapter 4 is used for this purpose, with a slight modification in terms of the departure time (i.e. the TBSP algorithm is used starting with the actual arrival time to the park-and-ride). After this path reassignment step for the intermodal passengers, the whole transit network is simulated including the simulation of transit only and intermodal passengers. In this simulation model, many factors are considered, such as passenger arrival time to the transit stop/station, transit vehicle arrival
time to the stop/stations, capacity constraints on the transit vehicles, etc. Therefore, any inconvenience during the trip, or any reason for missing the best transit alternative, is captured in the simulation, so that the path can be adjusted in the following iterations.

One important property of the transit model is in the simulation of the transit system using the travel time of the transit vehicles from the DTA as they are simulated along with the other vehicles in a congested roadway. Even the rail systems (e.g., light rail or streetcar) are modeled for simulation in the network according to their operational properties. To make this interaction between DTA and transit simulation possible, a new set of transit vehicle arrival/departures to/from each stop is estimated in the DTA model and is provided to the transit simulation, for more accurate assignment and simulation of transit passengers. This adjustment takes into account the fact that the transit schedule does not necessarily represent the actual transit vehicle arrival to the stops, and informed users behave based on their experience with the transit operations.

The simulation model also produces high resolution outputs such as the trajectory of passenger movements, ridership and boarding/alighting activities for every transit vehicle, and other MOE’s by market segments such as transit and intermodal demand, or bus and rail transit systems.

6.5. Application in the Integrated Travel Model in Sacramento Regional Network

The intermodal tour assignment and simulation have been implemented in C++ in three modules and are applied in a real project in the Sacramento regional transportation network as a part of the SHRP2 C10-B project. The project’s goal was to develop an
integrated dynamic travel model in a high-resolution multimodal transportation network. As a part of this model, an activity-based travel demand model (DaySim) was used to model daily travel activities and produce demand data at the tour level. The result of the activity-based model is the input to the network model. As mentioned before, the multimodal network model consists of DynusT as the DTA model and FAST-TrIPs as the transit and intermodal model. While there is a mechanism to make the activity-based travel demand and network models an integrated while, the focus in this study is on the network model only. However, auto, transit, and intermodal skim tables are produced to provide feedback to the demand model.

The dynamic multimodal network model has been developed in the form of an iterative process of running DynusT and FAST-TrIPs until the multimodal model converges. The structure of the model is shown in Figure 38, in which the intermodal demand is given to the intermodal assignment model, and after splitting the tours into auto and transit trips, each part is given to the appropriate model for further processing. After running one iteration of the DTA and FAST-TrIPs model, information is shared for the next iteration. For example, after running the DTA model, the travel times of transit vehicles are passed into FAST-TrIPs for passenger assignment and simulation. On the other hand, FAST-TrIPs produces the updated dwell times for every transit vehicle at each stop, and feeds it back to DynusT for the next iteration to simulate transit vehicles in the highway network. Additionally, new time-dependent link travel times, and transit vehicle trajectories, are used in the intermodal assignment model for reassigning the tours to park-and-ride locations. I.e., if the chosen park-and-ride does not result in a
satisfactory experience, the passenger may decide to choose a new park-and-ride according to the network state.

Figure 38 High level structure of the dynamic multimodal network model for Sacramento, CA
Figure 39 shows the modules and files passing between the modules to provide information to different models. The process starts with a Python script (ft_BST.py) to convert the output of the activity-based model to the proper text file (veh_sout.dat). This file contains the trip and tour information of the auto, transit, and intermodal travel. Then, FAST-TrIPs is run by a Python script (FAST-TrIPs.py) and executes the ft_intermodal.exe, ft_assignment.exe, and ft_simulation.exe sequentially. During this process, DynusT produces the two important files containing transit vehicle trajectories, and intermodal passengers’ arrival times (AltTime_transit.dat and AltTime_intermodal.dat). Accordingly, FAST-TrIPs produces the new set of dwell times in TransitDwellTime.dat for feedback to DynusT. This process continues until either convergence or the maximum number of iterations.
The convergence of the multimodal system is tested by measuring the stability of the transit vehicle dwell times. When the transit solution does not change by iterations, meaning that transit passengers find their best paths and continue using them, the dwell times will not change significantly anymore. Therefore, the effect of the transit vehicles on the auto network is almost fixed and the next set of transit vehicle trajectories will be similar to the current ones. This is how the system reaches a converged solution, where the effects of auto and transit on each other becomes negligible. Therefore, a similar dwell time gap defined in chapter 4, although at a lower level of aggregation (the transit vehicle trip), is used to measure the convergence of the system.

The multimodal network model was applied to the Sacramento regional network as a case study. Again, the DTA model development was not a part of this study, and the description can be found in the references. However, for the transit and intermodal network model, a brief description is provided. Similar to previous applications, GTFS files were used to build the schedule-based transit network in the hierarchical trip-based format. In fact, for this application, more than one set of GTFS files were used, and the transit routes in five different agencies were combined together. Table 8 shows the network properties in the Sacramento regional application. In addition to the transit network, the intermodal network at park-and-rides was modeled as described in section 6.2, and mode transfer delays were estimated. There are 23 park-and-rides modeled in the network. Figure 40 shows the Sacramento regional network, including auto and transit network as well as park-and-rides. The walking links are also generated by the estimated distance to a stop, considering the size of each traffic analysis zone (TAZ). That is, transit stops
that are either in the TAZ or within the 0.5 mile of the TAZ centroid are selected as accessible stops and a walking time is estimated for those stops. The transfer links are generated using a 0.25-mile distance criterion between pairs of stops.

Table 8 Sacramento transit network properties

<table>
<thead>
<tr>
<th>Agency</th>
<th>Full vs. Partial Network</th>
<th># of Stops</th>
<th># of Routes</th>
<th># of Trips</th>
<th># of Walking Links</th>
<th># of Park-n-Rides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sacramento Regional Transit</td>
<td>Full</td>
<td>3,063</td>
<td>67</td>
<td>2,323</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roseville Transit</td>
<td>Partial</td>
<td>45</td>
<td>18</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unitrans (Davis)</td>
<td>Partial</td>
<td>245</td>
<td>11</td>
<td>458</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yuba-Sutter Transit</td>
<td>Partial</td>
<td>63</td>
<td>3</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yolobus</td>
<td>Partial</td>
<td>380</td>
<td>11</td>
<td>131</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>-</td>
<td>3,796</td>
<td>110</td>
<td>2,954</td>
<td>22,256</td>
<td>23</td>
</tr>
</tbody>
</table>

Figure 40 Sacramento regional multimodal transportation network
The model was tested in 2 iterations of the multimodal assignment, in which 30 iterations of DynusT were performed to achieve equilibrium in the auto network. The tests were performed on a computer with Intel Core i5 CPU and 16GB of RAM. The transit and intermodal assignment of the model takes about 4 hours in each iteration, considering a full 24-hour day is simulated. After 2 iterations of the model, it was terminated (by reaching the maximum number of iterations) and the dwell time gap in the disaggregate level has reached the value of 0.16.

The main transit MOE’s produced by the model are shown in Tables 9 and 10. These results were obtained using a not-calibrated model and one should not expect very realistic numbers. However, using the default values (e.g. for the parameters of the transit route choice model), the existing results are satisfactory and the numbers make sense in most of the cases. For example, as shown in Table 9, the transit vehicle simulation indicates that the average operational speed of 15.25 miles per hour, and the average ridership of 30 passengers per vehicle trip. These numbers are intuitive considering the fact that all transit vehicle trips including those in peak and off-peak are simulated. On the other hand, as shown in Table 10, the average travel time in the transit network is 38.79 minutes with 22.29 minutes in-vehicle time. Table 11 shows some intermodal trips' characteristics, including the travel time in different components of the intermodal trips. As an example, the average auto travel time to access park-and-rides is 37.4 minutes in the network, and average transit travel time is 25.3 minutes for trips from park-and-rides to the destination. The results also show that people spend about 11 minutes on average to park their car, walk to transit stops, and board a transit vehicle at park-and-rides.
Table 9 Transit Vehicle Simulation Results in the Sacramento Case Study

<table>
<thead>
<tr>
<th>MOE</th>
<th>Total Value</th>
<th>Average Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># of transit vehicles trips</td>
<td>2,955</td>
<td>-</td>
</tr>
<tr>
<td>Traveled Distance (veh-miles, per veh-trip)</td>
<td>30,893</td>
<td>10.46</td>
</tr>
<tr>
<td>Operated Time (veh-hours, per veh-trip)</td>
<td>115,797</td>
<td>0.65</td>
</tr>
<tr>
<td>Speed (mph)</td>
<td>-</td>
<td>15.25</td>
</tr>
<tr>
<td>Ridership (unlinked passenger trips)</td>
<td>88,667</td>
<td>30.02</td>
</tr>
<tr>
<td>Average Load along the Route (persons)</td>
<td>-</td>
<td>7.35</td>
</tr>
<tr>
<td>Average Dwell Time at Each Stop (Sec)</td>
<td>-</td>
<td>6.81</td>
</tr>
</tbody>
</table>

Table 10 Transit Passenger Simulation Results in the Sacramento Case Study

<table>
<thead>
<tr>
<th>MOE</th>
<th>Total Value</th>
<th>Average Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># of passengers</td>
<td>42,803</td>
<td>-</td>
</tr>
<tr>
<td>In-Vehicle Distance (passenger-miles)</td>
<td>257,795</td>
<td>6.02</td>
</tr>
<tr>
<td>In-Vehicle Time (passenger-minutes)</td>
<td>954,119</td>
<td>22.29</td>
</tr>
<tr>
<td>Waiting Time (passenger-minutes)</td>
<td>371,823</td>
<td>8.69</td>
</tr>
<tr>
<td>Walking Time (passenger-minutes)</td>
<td>355,330</td>
<td>8.30</td>
</tr>
<tr>
<td># of Transfers</td>
<td>58,872</td>
<td>1.38</td>
</tr>
<tr>
<td>Transfer Time (passenger-minutes)</td>
<td>218,593</td>
<td>5.11</td>
</tr>
<tr>
<td>Travel Time (passenger-minutes)</td>
<td>$1.66 \times 10^6$</td>
<td>38.79</td>
</tr>
</tbody>
</table>
### Table 11 Intermodal Passenger MOEs in the Sacramento Case Study

<table>
<thead>
<tr>
<th>MOE</th>
<th>Average Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive Access Time (min)</td>
<td>37.4</td>
</tr>
<tr>
<td>Transit In-Vehicle Distance</td>
<td>4.6</td>
</tr>
<tr>
<td>Number of Transfers In the Transit Trip</td>
<td>0.5</td>
</tr>
<tr>
<td>Mode Change Delay at Park-and-Ride (min)</td>
<td>11.1</td>
</tr>
<tr>
<td>Transit In-Vehicle Time (min)</td>
<td>11.3</td>
</tr>
<tr>
<td>Transit Transfer Time (min)</td>
<td>1.5</td>
</tr>
<tr>
<td>Walk Egress Time (min)</td>
<td>1.4</td>
</tr>
<tr>
<td>Transit Travel Time (min)</td>
<td>25.3</td>
</tr>
<tr>
<td>Intermodal Travel Time (min)</td>
<td>62.7</td>
</tr>
</tbody>
</table>

#### Figure 41 Transit vehicle travel times

The equation for the line of best fit is $y = 1.010x + 0.235$ with $R^2 = 0.965$.
While the aggregate level transit results are good for a model without any calibration, some disaggregate transit outputs are provided and investigated to show how the proposed model can be used as a tool for planning and scenario analyses. Figure 41 compares the transit vehicle travel times in the simulation model and GTFS schedule. The simulated travel time is in fact the result of the DTA model, and includes the effect of traffic delay. However, transit dwell time as a part of the total travel time is estimated in the transit assignment based on the passenger boarding and alighting activities. The graph shows that travel times are consistent with the GTFS schedule.

Figure 42 shows the average number of on-board passengers along the route for the inbound and outbound trips of the two LRT lines. In the Gold line, the graphs are very consistent with the expectations and higher morning peak can be seen in the inbound trips while a moderate afternoon peak is seen in the outbound trips. In the Blue line, because it passes through the downtown area (or CBD), there are two peaks in both inbound and outbound trips each representing either AM or PM peak. Figure 43 also shows the number of on-board passengers along the route for the LRT Blue line. In the inbound trips, by going toward the downtown area, the vehicle load increases, and then decreases quickly. This pattern is consistent with the fact that people use the LRT system to access the downtown area for work trips. In the outbound trips, the vehicle load is high at the locations closer to the downtown area, and starts to decrease, but there is another peak along the route. The first peak is still consistent with the travel pattern for the work trips, and the second peak can be justified as other potential transit trips in another part of the
city. Note that the two peaks are still lower than the peak in the inbound trips, representing dispersion of the transit use in the afternoon.

The concluding remark is that the proposed model is an appropriate tool for modeling transit and intermodal tours. Some of the evidence showed that the model, although without calibration, produced results that are close to the expectations, and predicts the transit travel pattern consistently. However, the model outputs were not validated by comparison with the observation data, because such data were not available for this study. Therefore, the current results cannot be used for planning or decision making and calibration of the model and validation of the results are something to be done as future work.
Average Number of on-board Passengers

Vehicle Trip Departure Time

(a)

Vehicle Trip Departure Time

(b)
Figure 42 Average number of on-board passengers in the LRT lines a) 507-Gold, inbound b) 507-Gold, outbound c) 533-Blue, inbound d) 533-Blue, outbound
Figure 43 Average number of on-board passengers in the LRT Blue line a) inbound b) outbound
CHAPTER 7: CONCLUDING REMARKS

This study has been conducted with the motivation of enhancing transit and intermodal transportation network models and to make them compatible with the advances in Dynamic Traffic Assignment (DTA) and Activity-Based Models (ABM). The goal was to model schedule-based transit networks with high resolution, and to model user behavior appropriately in this type of network. Another goal was to build a set of models to fill the lack of transit modeling in the current DTA models and to complete the cycle in modeling multimodal transportation networks. Therefore, with an extensive effort, a comprehensive set of models were developed with a high level of flexibility in terms of modeling and integration with other transportation models. The models were integrated into a framework called FAST-TrIPs, which is now a powerful tool in the market for advanced travel modeling in the regional or local level. It has had several applications so far, mainly introduced in this report, and some other potential applications to be pursued in the future.

To give a brief conclusion about what has been achieved in this study, a review of the models is provided. At the beginning, a set of optimization models were proposed for different forms of transit-related problem, in the context of schedule-based transit systems. These models include the transit assignment with capacity constraints and boarding priority, extension to the stochastic assignment, and the extension to the multimodal assignment. While the original problem is mathematically challenging because of the asymmetric properties of the objective function, by using a penalty term
for the capacity constraints and boarding priority, the model may be solved using simulation. The solution method was developed based on two important sub-models:

1. Transit path algorithms in the hierarchical trip-based network
2. Mesoscopic transit passenger simulation

The path models were developed for both better user behavior representation and computational efficiency. For the first purpose, three path algorithm in schedule-based networks were proposed including a shortest path algorithm, a hyperpath algorithm to generate the choice set of paths using sequential logit model, and an A*–type algorithm for very fast path generation applications. These algorithms were implemented in the hierarchical trip-based network format to facilitate computational efficiency. The proposed network representation is an alternative to the conventional time-expanded networks, and is suitable for schedule-based transit network modeling.

The simulation model was developed to load the individual passengers to the network considering the operational characteristics of transit vehicles such as the schedule, arrival delays, capacity. The simulation model is capable of modeling every transit vehicle and individual passenger and is developed to model high-resolution interaction between supply and demand. In other words, a complete traveler trajectory including arrivals to the boarding stop, joining the queue, boarding and alighting activities, transfers between routes, and egress to the destination is modeled in the simulation model. By capturing the inconvenient situations such as missing a transit vehicle for being late, or failure to board because of the capacity limit on the transit vehicles, the simulation model is capable of evaluating the experience of the passengers as a result of their choices.
These new modules were used to shape the transit assignment and also the intermodal assignment algorithms. In the first application, an iterative process of the path assignment and loading by simulation were combined to solve the transit assignment model with capacity constraints and boarding priority. The application of the model was in San Francisco MUNI network with a major capacity problem, and the model was found to be useful for this case. Moreover, the application of the model in the San Francisco case study showed good computational efficiency, reasonable convergence properties, and effectiveness of the model in representing user behavior in spite of limited model calibration effort.

The other model was the intermodal tour assignment in an integrated travel model in Sacramento, CA. Based on the transit path models, and a multisource time-dependent auto shortest path algorithm (MTDSP), a model was developed to assign intermodal tours to the optimal park-and-ride locations and to determine the auto and transit trips of the intermodal tours. Then, the produced trips were simulated in the DynusT and FAST-TrIPs to ensure the trip chain is complete in terms of the locations (to park and pick up the car) and time (to transfer from and to auto to and from transit). Unsuccessful experiences were reported and the assignment was adjusted in an iterative process. The application of this model was in the Sacramento regional network and using the default values for the model parameters, a reasonable set of results were obtained.

This research study was defined with a large scope, meaning that a comprehensive set of models were developed. Yet, it is obvious that calibration and validation of the models could not be done with the limited time, data, and other resources. In the San Francisco
case study, some model estimation results were available for transit route choice from the previous studies. However, multiple model parameters have yet to be fully calibrated, and a complete validation effort is required to make a model ready for planning and scenario analysis purposes. This issue is more obvious in the Sacramento case study, as less information was available for model estimation, and default values were used to test the model. Therefore, no validation can be done about the model outputs in this case study. In addition to the transit models, the DTA models have to be fully calibrated because they affect the transit model drastically. I.e., FAST-TrIPs uses the transit vehicle trajectories to model transit system and unless realistic transit vehicle travel times are produced by the DTA models, one cannot expect good results on the transit side. These facts indicate that model calibration and validation is an important step for the future applications of the developed models.

Another future work is improvement of computational efficiency, although they are relatively efficient in the current form. Future application of the models, in either larger networks or problems with more complexity, justifies the exploitation of advanced computational techniques (multi-thread, multi-core, parallel processing) to make the models run in even faster speed. Additionally, a more in-depth investigation of user behavior such as passenger arrival time to the boarding stops, and the evaluation of arrival time to the destination (being early vs. late to the destination), and also the inclusion of individual value of time (VOT) in path choice, will be pursued in future applications. The former requires a comprehensive set of observation data, while the latter is simple because the models are person-based with the capability of modeling
individual travelers. As long as the VOT information is provided with the route choice model, these details can be incorporated in the assignment model.

Another important concept in transit modeling is how to deal with transit sub-modes such as bus (regular vs. express vs. rapid, etc), rail (light rail, commuter, etc) and other sub-modes. The author’s experience suggests that the transit sub-modes are usually modeled separately in activity-based models, meaning that the mode choice model may determine which transit sub-mode will be used by a traveler. However, the assumption in this study was to assign the travelers to the best transit option, which may include either (or a combination) of the sub-modes. In fact, the two approaches have their pros and cons in different applications. Nevertheless, for being consistent with the previous steps of travel forecasting models (i.e. ABM models), FAST-TrIPs can be enhanced for assigning individual passengers to the determined transit sub-modes. This enhancement, which is not very complicated, has not been done in the current applications because it was not critical to the model development objectives, and the required information was not provided by the demand models.

The path-overlapping problem is another important problem in transportation network modeling, especially in transit modeling. Although this problem has been reviewed by the authors and simplified methods are conceptualized for implementation, an advanced modeling consideration can be a potential future work in the proposed models. Finally, an in-depth study on the integration of demand and supply models, particularly with consideration of a schedule-based transit system, and investigation of the overall model
characteristics such as system convergence, can be a challenging research relevant to this study.
REFERENCES


32. Khani A., Sall E., Zorn L. and Hickman M., (2013). “Integration of the FAST-TrIPs Person-Based Dynamic Transit Assignment Model, the SF-CHAMP Regional, Activity-Based Travel Demand Model, and San Francisco’s Citywide Dynamic Traffic Assignment Model”. Proceedings of the 92nd Annual Meeting of Transportation Research Board, Washington DC.


