

ABSTRACT

Title of Dissertation: INTEGRATING ACTIVITY-BASED TRAVEL
DEMAND AND DYNAMIC TRAFFIC
ASSIGNMENT MODEL: A BEHAVIORAL USER
EQUILIBRIUM APPROACH

Di Yang, Doctor of Philosophy, 2018

Dissertation directed by: Professor Lei Zhang
Department of Civil and Environmental Engineering

Recently, the focus of transportation planning has evolved from accommodating long-term mobility needs to providing near-term and more efficient transportation systems management and operations (TSMO) solutions, the result of limited transportation funding and road capacity build-out. This planning-for-operation concept calls for modeling tools that are sensitive to dynamic interactions between travel behavior and network supply so that the impacts of emerging TSMO strategies (e.g., variable road pricing, ramp metering, etc.) can be accurately estimated. The integration of activity-based travel demand models (ABM) and dynamic traffic assignment (DTA) models offer a perfect solution. However, existing operational integrated ABM-DTA models suffer from several limitations, including excessively long runtime and poor convergence quality, which severely hinders large-scale implementations.

This dissertation proposes to integrate operational ABM and DTA models based on an innovative behavioral foundation: behavior user equilibrium (BUE). Different from the normative behavior theory (i.e., user equilibrium, or UE), BUE is based on a positive theory of travel behavior that avoids impractical assumptions, such as complete information and perfect rationality. BUE describes what travelers actually do in the system and thus emphasizes the role of information acquisition, knowledge updating, and learning in travel decision-making. The BUE-based model saves runtime because DTA models no longer need to run iteratively to reach UE internally and fewer agents undergo behavioral adjustments through iterations. In addition to runtime savings, the BUE principle proposes an alternative way to explain the behavior adjustment process and provides improved behavioral realism. This BUE-based integration framework is applied to the Washington-Baltimore Metropolitan Area as a case study. The integrated model includes InSITE, an ABM developed for the Baltimore Metropolitan Council (BMC), and DTALite, a mesoscopic DTA model. The BUE-based integrated model is then compared with a traditional, sequentially integrated benchmark regarding model convergence and performance. Lastly, to enhance the transferability of the BUE-based integration approach, this dissertation develops a calibration method that estimates parameters associated with the BUE principle using readily available local data so that this integration framework can be easily applied to operational ABM and DTA models elsewhere.

INTEGRATING AND ACCELERATING ACTIVITY-BASED TRAVEL
DEMAND AND DYNAMIC TRAFFIC ASSIGNMENT MODEL:
A BEHAVIORAL USER EQUILIBRIUM APPROACH

by

Di Yang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:

Dr. Lei Zhang, Chair

Dr. Francis B. Alt, Dean's Representative

Dr. Frederick W. Ducca

Dr. Cinzia Cirillo

Dr. Chenfeng Xiong

© Copyright by
Di Yang
2018

Dedication

This dissertation is dedicated to my fiancée Jing Du, and my parents Ailing Wang and Zeyan Yang.

Acknowledgements

I am grateful to a number of people who have helped, supported, and inspired me throughout my PhD Study. First of all, I would like to express my immense gratitude to my advisor and mentor, Dr. Lei Zhang, for the instruction, encouragement, and protection during my study at UMD. You have always been a role model for me in almost every aspect of life and academics.

I would like to extend my appreciation to the rest of the dissertation committee: Dr. Fred Ducca, Dr. Francis Alt, Dr. Chenfeng Xiong, and Dr. Cinzia Cirillo for the support, detailed comments and constructive suggestions to this dissertation. I would also like to thank my colleagues Liang Tang, Zheng Zhu, Carlos Carrion, Bo Peng, Jun Zhao, and Yixuan Pan for helping me and always having my back.

Lastly, my deepest gratitude goes to my family for the love and care. Especially, my beloved fiancée Jing Du. You are my soulmate and I am the luckiest guy in the world to have you.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	viii
List of Figures	ix
Chapter 1 Introduction	1
1.1 Background	1
1.2 Research Scope and Objectives	5
1.3 Research Contributions	9
1.4 Dissertation Organization	11
Chapter 2 Literature Review	14
2.1 Travel Demand Models and Earlier Modeling Approaches	16
2.2 Activity-Based Models	19
2.2.1 Constraints-based models	20
2.2.2 Econometrics-based models	22
2.2.3 Rule-based models	24
2.2.4 Agent-based models	25
2.3 Network Supply Models	26
2.4 Integrated ABM and DTA Models	28
2.5 Behavioral Foundation	31
2.5.1 Perfect rationality and bounded rationality	31
2.5.2 Behavioral user equilibrium	34

2.6 Summary	36
Chapter 3 ABM and DTA Integration Framework.....	39
3.1 A Sequential Integration Framework.....	40
3.2 A BUE-Based Integration Framework	41
3.2.1 Framework overview	41
3.2.2 Knowledge and learning	45
3.2.3 Subjective search gain.....	48
3.2.4 Perceived search cost	50
3.2.5 Model estimation	52
3.3 Model Calibration.....	54
3.3.1 Problem statement.....	55
3.3.2 Simultaneous perturbation stochastic approximation (SPSA).....	57
3.4 Discussion.....	60
Chapter 4 Data	62
4.1 Data from Behavioral Surveys.....	63
4.2 Model Calibration Data	65
4.3 Passively Collected Data	68
4.3.1 Introduction.....	69
4.3.2 Methodology	71
4.3.3 Data collection	72
4.3.4 Data processing	74
4.3.5 Empirical results	77
4.4 Summary.....	79

Chapter 5 Integrated Model Application in the Washington-Baltimore Area.....	80
5.1 Major Model Components	81
5.1.1 BMC InSITE ABM.....	81
5.1.2 DTALite.....	85
5.2 Model Integration Implementation	88
5.2.1 Overall process.....	88
5.2.2 Information exchange between models.....	90
5.2.3 Towards a tour-based BUE model.....	91
5.3 Accelerating InSITE ABM and DTALite.....	92
5.3.1 Multiprocessing in InSITE ABM.....	92
5.3.2 Time-dependent skim generation in DTALite.....	93
5.4 Calibration and Validation Results	94
5.4.1 SPSA calibration.....	95
5.4.2 Validation results	98
5.5 Application Results.....	103
5.6 Practical Implementation of Fast Convergence and Runtime	107
5.7 Summary.....	109
Chapter 6 Conclusions	112
6.1 Research Summary	112
6.2 Discussions and Future Research	115
Appendices.....	118
Appendix A: Integration Implementation Code	118
Appendix B: SPSA Code.....	129

Bibliography 133

List of Tables

Table 4-1 HTS Descriptive Statistics.....	67
Table 4-2 Mode Detection Data Description of Trajectory Features	76
Table 4-3 Overall Model Statistics	77
Table 4-4 Confusion Matrix.....	78
Table 5-1 SPSA Coefficients	95
Table 5-2 Calibration Results	97
Table 5-3 Calibrated Search Cost Model Parameters.....	98
Table 5-4 Validation by Function Class	101
Table 5-5 A 15-Iteration Model Runtime Breakdown.....	108
Table 5-6 Runtime Comparison.....	108

List of Figures

Figure 1-1 InSITE Model Traffic Analysis Zone (TAZ) Map	8
Figure 2-1 Decision-Making Process in BUE	35
Figure 3-1 Sequential Integration Framework.....	41
Figure 3-2 BUE-Based Integration Framework Illustration	43
Figure 4-1 Traffic Count Sensor Location.....	66
Figure 4-2 The User Interface of Smartphone GPS Data Collection App.....	73
Figure 5-1 InSITE Model Structure.....	83
Figure 5-2 DTALite System Architecture with Key Modeling Components (Source: (Zhou & Taylor, 2014)).....	87
Figure 5-3 Integration Framework Structure	89
Figure 5-4 SPSA Convergence Over Iterations	97
Figure 5-5 Observed and Simulated Trip Mode Shares.....	99
Figure 5-6 Observed and Simulated Trip Distance Distributions.....	100
Figure 5-7 Observed and Simulated TOD Distributions	101
Figure 5-8 Assignment Results Compared with Observations	102
Figure 5-9 Locations of Over- and Underestimated Links	103
Figure 5-10 Changes of Gain/Cost Ratio over Iterations.....	105
Figure 5-11 Convergence of the Integrated Model.....	106
Figure 5-12 Mode Share over Iteration.....	107

Chapter 1 Introduction

1.1 Background

Transportation, by definition, is the movement of humans, animals, and goods from one location to another. A transportation system is an essential prerequisite for people to trade and conduct activities; the first transportation systems were integral to the development of civilization and urbanization. People often take transportation systems for granted until something goes wrong; traffic congestion is a constant reminder of their importance. When recurring congestion occurs, the blame is typically placed on inaccurate growth anticipation and poor transportation planning efforts. The rapid economic development over the past half-century has signified the critical role transportation planning plays in supporting investment allocation, policy-making and economic growth resulting in significant progress in transportation planning techniques.

In practice, a comprehensive transportation planning process is usually dependent on the development of a complex, yet rigorous mathematical model to represent the demand and supply for travel in an urban region, which is referred to as a travel demand model. The primary objective of a travel demand model is to simulate planning scenarios of proposed plans and policies accurately. Driven by emerging transportation policies and modern traffic management strategies, the focus of travel demand analysis

has shifted accordingly. While earlier travel demand models emphasize accommodating long-term mobility goals at the aggregate level, recent models are concerned more with understanding the individual behavioral process in response to mid-term or even short-term transportation management strategies. For instance, activity-based travel demand models (ABMs) have improved trip-based approaches by: 1) considering activity patterns instead of individual trips; 2) modeling the individual traveler's decision-making process; 3) incorporating intra-household interactions; and 4) treating time as a continuum (Lin, Eluru, Waller, & Bhat, 2008).

In addition to methodological improvements, travel demand analysis has also experienced theoretical advances. The normative theory of travel behavior (e.g., perfect rationality, utility maximization, etc.) previously dominated travel demand analysis because of its behavioral simplification. However, simplifications imply limitations, especially in modeling travelers' responses to emerging control strategies and sophisticated policies. Therefore, rule-based and agent-based ABMs, based on either bounded rationality theory or advanced behavioral foundations, are proposed to relax the perfect rationality assumption in utility maximization-based ABMs. This makes it possible to model the impacts of emerging transportation management systems such as real-time information systems (RTIS) and advanced travel information systems (ATIS).

Similar methodological improvement could also be observed in the development of network supply models. Traditionally, a static traffic assignment model is employed as

the last step of the classical four-step travel demand model. Recently, dynamic traffic assignment (DTA) models have gradually replaced static models by: 1) considering time-dependent interactions of travel demand and supply; 2) modeling the movement of each individual traveler; 3) capturing congestion build-up and dissipation; and 4) incorporating demand management strategies and intelligent transportation system technologies.

As recently underscored by the U.S. Department of Transportation, it is critical to integrate transportation systems management and operations (TSMO) strategies into the planning process to advance transportation system efficiency, reliability, and operations (Grant, Bauer, Plakson, & Mason, 2010). This planning for operations concept calls for modeling tools that are sensitive to dynamic interactions between travel behavior and network supply so that the impacts of emerging TSMO strategies (e.g., variable road pricing, ramp metering, etc.) can be accurately estimated. The integration of ABM and DTA models is naturally a perfect choice because ABM and DTA were initially designed for planning and operation purposes, respectively. The combination of both models captures the interplay of demand and supply, which addresses the shortcomings of each independent model. Furthermore, both models employ micro-simulation approaches and operate at finer temporal resolutions, which provides consistent results and utilizes their full potential.

The behavioral foundation determines not only how travel demand models are formulated but also how the ABM and DTA models are integrated. Even though the

behavioral foundation in ABM has evolved from perfect rationality to a bounded rationality paradigm, the behavioral foundation in DTA has focused on perfect rationality-based user equilibrium (UE) principles. In this context, DTA models must load traffic to the network iteratively to reach UE conditions, which is already time-consuming. Additionally, most existing ABM-DTA integrated models rely on a gap-function-based convergence criterion. In other words, ABM and DTA must operate sequentially and iteratively to ensure that differences in travel time or origin-destination (OD) tables between two iterations fall into specific predefined gaps. Consequently, the model runtime remains a significant issue hindering agencies from applying the integrated model for policy analysis, which is especially true for large-scale implementation.

For example, the Second Strategic Highway Research Program (SHRP2) Project C10B developed a dynamic integrated model that combined the activity-based demand model DaySim and a DTA model DynusT for Sacramento, California (T. Rossi & Systematics, 2012). The runtime of this model used by the Sacramento Area Council of Governments (SACOG) was around 70 hours; it included three feedback loops (between DaySim and DynusT) with only 10 iterations of DynusT runs. It should be noted that 10 DTA iterations are challenging to reach UE, especially for such a large-scale implementation, and it is still not clear how many feedback loops would guarantee model convergence. Therefore, the DTA iteration and feedback loop numbers in this project can be viewed as a compromise that sacrifices model convergence for model runtime. In addition to model runtime, the existing sequential integration method does

not model the decision-making process the way travelers behave in real-world situations. The behavioral simplification and impractical assumption in the integrated models would yield poor behavior realism, which ultimately leads to deficient model sensitivity to emerging transportation policies. These integration challenges are faced by all researchers in the ABM and DTA integration field.

1.2 Research Scope and Objectives

Given the aforementioned integration challenges, this dissertation seeks to integrate ABM and DTA models based on an innovative behavioral foundation that improves the model efficiency while attaining advanced behavioral realism. Behavioral user equilibrium (BUE) is based on positive behavior theory that is concerned with how travel-related decisions are actually made instead of how they should be made as in normative theory. This positive theory assumes limited information in decision-making procedures and emphasizes the searching/learning process of each traveler, where past experiences influence future travel behavior. Two behavioral concepts are theorized when travelers search for travel alternatives (e.g., travel destination, departure time, travel mode, etc.): perceived search cost and expected search gain. The BUE in a transportation system is reached when all travelers stop searching for alternative travel options because the perceived cost of an additional search exceeds the expected gain. The integrated ABM and DTA model based on the BUE principle employs the agent-based modeling approach and defines the convergence status based on the behavior of

an agent instead of the whole network performance. This principle saves model runtime because the DTA model does not need to run iteratively to reach UE condition internally and BUE convergence is guaranteed regardless of the network size (Xiong, 2015; Zhang, 2011). In addition to runtime savings, the BUE principle explicitly models the behavior adjustment process at the individual level, which provides more insights on the behavior dynamics.

To implement the BUE principle in integrating ABM and DTA models, this dissertation first proposes a BUE-based theoretical integration framework. The framework describes how to employ the agent-based approach in the integration and how each agent learns and searches through iterations. Key BUE concepts, perceived search cost, and expected search gain are specified in the context of an econometric-based ABM. Specifically, the expected search gain is affected by travelers' network information and expectations on travel cost saving, which is calculated based on the variation of the traveler utility. The perceived search cost measures the efforts involved in the search process; three ways to estimate search cost are proposed in this dissertation. Previously, the perceived search cost was estimated from behavioral survey data. With previous estimated model parameters, a calibration approach is developed to calibrate search cost parameters based on observed patterns. This approach provides excellent transferability, where existing ABM and DTA integrated models can easily be converted to a BUE-based paradigm without additional data collection cost.

While behavioral survey data could provide stated behavior adjustment information, facts revealed regarding people's actual behavior changes are preferred. Passively collected data could be a perfect data source for estimating BUE-based models. This dissertation showcases how to derive travel behavior information from passively collected data with the help of data mining and machine learning methods to supplement or replace the traditional travel surveys. Specifically, a random forests model is developed to impute the mode information from smartphone-collected GPS data. The detailed information regarding data collection, data processing, and results are presented in Chapter 4.

The proposed theoretical integration framework is then applied to the Washington-Baltimore metropolitan area. The integrated model includes InSITE, an ABM originally developed for the Baltimore Metropolitan Council (BMC), and DTALite, a mesoscopic DTA model. The modeling area includes Anne Arundel County, Baltimore County, Carroll County, Harford County, Howard County, Montgomery County, Prince Georges County, Frederick County, Baltimore City and the District of Columbia (Figure 1-1).

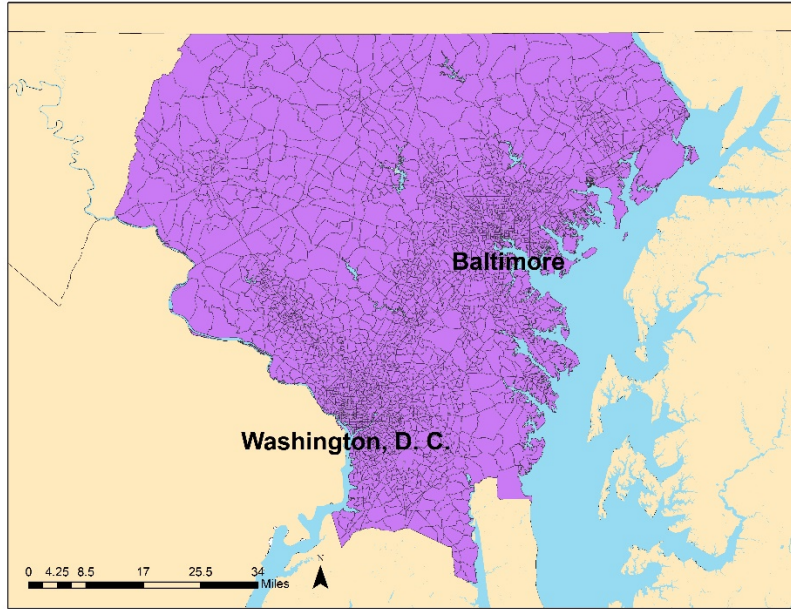


Figure 1-1 InSITE Model Traffic Analysis Zone (TAZ) Map

To further accelerate the integrated model, this dissertation improves the runtime of both the InSITE and DTALite models by maximizing computational resource utilization. Specifically, the InSITE model has been modified to take advantage of multi-processing in Python, which improves the runtime fourfold. A fast time-dependent skim generation module has been developed to significantly reduce DTALite runtime. With speed improvements from both model components, the BUE-based integrated model is compared with a typical sequential model integrating approach. Comparison details, including model performance, convergence status, and runtime, are carried out in Chapter 5.

1.3 Research Contributions

The aforementioned modeling challenges in integrating ABM and DTA models motivate this study to develop a BUE-based ABM and DTA model integration approach. Unlike the existing integrated ABM and DTA approach that merely runs both models sequentially, the proposed integration approach is based on a theoretically sound behavioral foundation. This approach employs an agent-based modeling technique and focuses on modeling the learning and searching process of each agent through iterations. At every iteration, each agent would first decide whether to search for new travel alternatives based on his/her limited information and prior experience. The BUE of the transportation system is achieved when no traveler in the system is willing to search for new travel alternatives.

Methodologically, the new ABM-DTA integration method proposed in this study provides improved behavioral realism by capturing what a traveler does in the decision-making process. The model convergence in the context of the BUE principle is measured by the behavior of each traveler instead of network-wide performance as in the traditional sequential integration method. Therefore, the convergence status is guaranteed, since no traveler would search for alternative travel options unendingly; this has been proved by previous studies (Xiong, 2015; Zhang, 2011). Additionally, to enhance the transferability of the BUE-based integration approach, this study develops a calibration process to estimate the perceived search cost in the model so that this

integration approach can be easily applied to ABM and DTA integration needs elsewhere.

To implement the BUE theory in the context of ABM and DTA integration, the theory is extended and enhanced in the following four aspects: 1) provide the initial traffic condition without relying on outside travel demand models; 2) consider trip-chaining constraints to fit into the tour-based setting in ABMs; 3) incorporate a new behavior dimension—destination choice—into the framework; 4) propose a model calibration method to streamline the search-cost parameter estimation process.

Practically, the BUE-based integration method improves the model runtime by relaxing the UE assumption in the assignment model. Therefore, DTA models do not need to run iteratively to reach the UE condition. Additionally, the overall model convergence is measured by the behavioral adaptation process of each traveler instead of calculating the average network travel time between iterations. Consequently, the convergence of computation time is irrelative to network size and the number of travelers. To further accelerate the integrated model, parallel computing techniques and tree-based shortest path storage are developed for the ABM and DTA models, respectively.

In addition to the BUE-based integration approach, this study also explores how travel behavior information can be derived from passively collected data. Passively collected data (e.g., smartphone GPS data) could reveal important behavior information with the help of proper data mining and machine learning techniques. The random forests model

used in this study could accurately identify travel mode information from smartphone GPS data. Similar methods could reveal other behavior information, such as trip purposes and their user socio-demographic information. This approach could supplement or replace existing travel surveys, which are typically costly and time-consuming.

1.4 Dissertation Organization

The rest of this dissertation is organized as follows:

Chapter 2 provides a comprehensive literature review on the evolution of travel demand modeling with a focus on various types of ABMs. This chapter then revisits the advances in network supply models and the development of DTA models. The motivation behind ABM and DTA integration is explained and existing integration methods are introduced. Lastly, various behavioral foundations in the literature are summarized and the concept of BUE is presented.

Chapter 3 develops the theoretical framework on integrating ABM and DTA models based on the BUE principle. The first section in this chapter introduces the sequential integration approach, which is commonly employed in the literature, as the benchmark method. The BUE-based integration framework is then illustrated where integration-technical details—including the convergence criteria and specifications of expected

search gain and perceived search cost—are carried out. Lastly, a model calibration approach is proposed to improve the transferability of the integration framework.

Chapter 4 describes three data sources: data previously used, data currently being used and data to be used in the future. This chapter starts by describing the longitudinal data that was previously used to estimate the BUE model. A section then introduces the local data that can be used to calibrate the model parameters. Lastly, a case study is presented to demonstrate how passively collected data could provide valuable travel behavior information.

Chapter 5 summarizes the real-world application of the proposed integration framework in the Washington-Baltimore Metropolitan Area. Model components BMC InSITE ABM and DTALite are introduced, as well as intermediate steps for data communication between the two models. The overall software implementation is described with a focus on the parallel computing technique employed in InSITE to speed up the integrated model. Following the implementation details, empirical results are presented. Calibration results are first given, including the performance of the SPSA algorithm and calibrated parameter values. The actual model results at various travel dimensions, as well as the model behavior dynamics, are then presented. The discussion section that follows analyzes the model results and compares the model performance with the sequential integrated model.

Chapter 6 concludes the dissertation. This chapter summarizes research efforts, results and major findings. It ends with a discussion on the future research direction to further enhance the BUE-based integration framework, as well as application prospects in travel demand modeling.

Chapter 2 Literature Review

Transportation planning involves the process of designing policies/regulations, managing transportation infrastructures, and managing future travel growth. Accurate transportation forecasting, the key to a successful transportation planning process, usually relies on the development of rigorous travel demand models. Over the past few decades, tens of thousands of papers have dedicated to improving travel demand models, especially on the urban passenger side. From aggregate, four-step, trip-based models to microsimulation, activity-based, tour-based models, advances in both statistical methods and computational technologies have allowed the development of travel demand models with two objectives: individual travel behavior realism and sensitivity to policy scenarios. Section 2.1 briefly introduces the history of urban passenger travel demand models, as well as earlier travel demand modeling approaches. The development of activity-based models (ABM) and typical ABM categories are reviewed in Section 2.2.

Network supply models are critical components of transportation forecasting. As the last step of the traditional four-step travel demand modeling approach, it has long been recognized that travel demand is affected by transportation network supply. From static traffic assignment to microsimulation dynamic traffic assignment (DTA), network supply models have evolved to address vast modeling challenges arising from emerging policy goals and various application needs. Section 2.3 introduces the history of

network supply models and summarizes the development of DTA models over the past several decades.

ABM and DTA models represent significant advancements that have occurred on the demand and supply sides; both operate at the individual level and consider time as a continuum (or at small time intervals). The integration of these two models would provide improved behavioral realism and yield dynamic and consistent results. The integration of ABM and DTA models is a topic that has been discussed for a long time but still lacks good practical applications. Section 2.4 reviews both the state-of-the-art and state-of-the-practice in ABM and DTA integration and presents major integration challenges encountered in integrating a fully operational ABM and DTA model at the metropolitan level.

Given the significant integration challenges (including model runtime, convergence criteria, etc.) that are summarized in Section 2.4, this dissertation proposes to integrate ABM and DTA models based on a new behavioral principle: behavioral user equilibrium (BUE). Built on the agent-based modeling approach, the BUE principle defines the equilibrium status from an individual's perspective rather than a network-wide point of view, which allows for better behavioral realism and more realistic convergence criteria. Section 2.5 examines various user equilibrium principles under different behavioral assumptions and the positive theory of travel behavior SILK (e.g., Search, Information, Learning and Knowledge), from where BUE is derived.

Finally, Section 2.6 presents the summaries from the literature review, including the research gap in the existing literature and how this dissertation contributes to the ABM and DTA integration field.

2.1 Travel Demand Models and Earlier Modeling Approaches

Travel demand models have evolved rapidly over the past few decades, particularly with increased computer performance in recent years. From the simplest sketch-planning models and strategic-planning models to the dominant trip-based models and more advanced ABM, the development of travel demand models has undergone a long and complex process. Therefore, it is not in the scope of this section to offer a comprehensive review of the history of travel demand model development; rather, this section will discuss the development of activity-based travel demand models and recent trends of the agent-based modeling approach to forecasting travel demand.

In general, travel demand models may take two forms. One form focuses on addressing specific questions in transportation planning, such as traffic impact studies for new development or travel analysis for a particular corridor or city district. Both sketch-planning and strategic-planning models are tools designed to tackle problems that are often narrow in scope. They are usually simple to implement, require fewer data and only produce rough estimates of travel demand. The other form of models focuses on seeking comprehensive answers to multiple, often interrelated aspects of transportation

planning. They are usually complex and large-scale models; four-step models and ABM are typical examples (Castiglione, Bradley, & Gliebe, 2014). The rest of this section focuses on introducing travel demand models in this category.

The four-step travel demand model is considered the most popular and widely-implemented comprehensive travel demand model in the literature. It was first implemented on the mainframe in the 1950s through the Detroit Metropolitan Area Traffic Study and Chicago Area Transportation Study (Chicago Area Transportation Study, 1959) to plan major highway facilities. The four-step modeling practice was standardized in the 1960s; since then, the use of travel demand models to support Metropolitan Planning Organization (MPO) planning activities has been enforced by federal requirements. As of today, many MPOs still actively maintain four-step models for transportation planning analysis.

As the name suggests, a four-step model consists of four primary components, which execute in sequence. The first trip generation step estimates the number of trips produced and attracted by each traffic analysis zone (TAZ). The second trip distribution component determines the destinations of trips generated at each TAZ, which is why this component is also referred to as destination choice. The third mode choice step predicts the travel mode (e.g., drive, transit, walk, bike, etc.) for each trip. The fourth trip assignment component, also known as route choice, calculates which routes in the network facilities each trip chooses. Other travel information in this process is either regarded as exogenous inputs (e.g., land use) or neglected (e.g., time-of-day choices).

With the main structure of four-step models being standardized, subsequent studies target more on improving model components with new methodologies. For instance, FHWA (1967) introduced a regression analysis to replace traditional cross-classification methods in trip generation. Of all these methodological advances, the most significant one was the introduction of discrete choice models (Ben-Akiva, 1973; Domencich & McFadden, 1975; McFadden, 1978). The proposed utility-based econometric formation was later widely employed in any choice-related travel dimensions such as destination choice and mode choice.

Despite the popularity, four-step models have been criticized for the following aspects: 1) four-step models are typically trip-based, which overlook the interdependence among different trips (e.g., the mode choice for two successive trips should be the same, which cannot be recognized by trip-based models); 2) four-step models are aggregate in nature, which fails to capture travel behavior at the individual level; 3) the static nature of the model over time; 4) long analysis periods that makes it difficult to model time-of-day related decisions; and 5) failure to consider intra-household interactions (e.g., escorting kids to and from schools; (Castiglione et al., 2014; T. F. Rossi & Shiftan, 1997)). Recognizing these shortcomings, researchers have been studying the next generation of travel demand models, which are introduced in the following sections.

2.2 Activity-Based Models

Activity-based models (ABMs) are considered the second generation of travel demand models. A fundamental premise of ABMs is that travel demand stems from the need for activity participation. Modeling the decision process of each traveler, ABMs capture improved behavioral realism and better represent how policies, developments, and travel growth impact people's travel behavior, which leads to more accurate predictions. Key features of ABMs include: 1) ABMs capture each individual's decision-making process, which provides detailed information and enables the analysis of personalized policies; 2) ABMs incorporate the continuous temporal dimension, which allows for modeling time-related studies (e.g., departure time choice and dynamic pricing strategies); 3) ABMs are typically tour-based, which takes into account the interrelation among several consecutive trips and distinguishes between mandatory and non-mandatory tours; and 4) ABMs usually consider the intra-household interaction.

The theoretical exploration of the activity approach began in the 1970s-80s. Hägerstraand's work (Hägerstraand, 1970) first addressed the relationship between activity participation and time-space concepts. Chapin (1974) later explored the activity patterns of the urban population. Fried et al. (1977) attempted to analyze the impact of social structures and rules on household travel behavior in urban areas. Jones (1977) nicely summarized the work and proposed the well-known theory that travel is derived from the need to participate in activities at different points in space and time.

Parallely, the random utility maximization (RUM) theory proposed by McFadden laid the foundation for empirical ABMs and remains the most popular travel analysis approach to date.

The early research discussed above have established the conceptual foundation of the activity-based approach. In recent decades, with the improvements of modeling methodologies, computation capacity, and data collection methods, the development of conceptual and empirical ABMs began to explode. Up to now, ABMs have received significant attention and have made substantial progress regarding standardization and implementation. The rest of this section focuses on introducing several categories of ABMs based on their modeling approach: 1) constraints-based models; 2) econometrics-based models; and 3) rule-based models. It is also critical to note that the above categories are not exclusive. Two or more of the approaches, or other approaches such as agent-based approaches, can be combined to develop ABMs. These hybrid approach-based ABMs, however, are not the focus of this section.

2.2.1 Constraints-based models

Constraints-based models are the first generation of ABMs, which try to explore whether an activity schedule is feasible under particular space-time constraints. PESASP (Program Evaluating the Set of Alternative Sample Paths) model, developed by Lenntorp (1977, 1978), is the first attempt to implement Hägerstrand's theoretical

framework in a way that allows for significant policy analysis. PESASP was designed to analyze possible combinations of activities under time and space restrictions. Jones et al. (1983) later proposed CARLA (Combinatorial Algorithm for Rescheduling Lists of Activities) model, which generated all feasible activity patterns caused by policy changes. Model inputs include a list of activities to be scheduled, the durations, and time-of-day constraints. The model then outputs a list of all possible arrangements of these activities. CARLA was implemented in the Burford School study where the schedules of a few pupils were generated following a policy change.

Besides the two earlier models mentioned above, examples of constraints-based ABMs also include BSP (Huigen, 1986), MAGIC (Dijst, 1995; Dijst & Vidakovic, 1997) and GISICAS (Kwan, 1997). Compared to later modeling approaches that predict individual and household activity patterns, constraints-based ABMs focuses on checking whether an activity agenda is feasible in particular space-time constraints. The space-time constraints are often locations, available transportation modes, and travel times between locations by specific modes. A combinatorial algorithm is usually employed to generate all feasible activity patterns and then the feasibility of each pattern is checked based on certain rules. However, constraints-based ABMs also suffer from some limitations: 1) most models only acknowledge individual-level rather than household-level accessibility; 2) space-time constraints assume isotropic conditions where travel is equally smooth in all directions; and 3) most models lack an explicit mechanism related to choice behavior under uncertainty (Rasouli & Timmermans,

2014). Consequently, new activity-based modeling approaches have been proposed to resolve these limitations.

2.2.2 Econometrics-based models

Econometrics-based ABMs primarily find their theoretical foundation in random utility maximization (RUM) theory from choice modeling, where individuals make each travel-related decision that maximizes their utility. These ABMs are typically composed of a series of utility-maximization based discrete choice models (e.g., multinomial logit, nested logit, and mixed logit models) that predict different dimensions in travel-related decisions (e.g., mode choice, destination choice, and departure time choice). Additionally, several other econometric structures, including hazard-based duration models and ordered response models, are also used to model various travel-related decisions.

Earlier discrete choice models in transportation only focused on one dimension of the travel-related decisions, of which mode choice is the most popular topic. Adler and Ben-Akiva (1979) first extended the single-dimension model to multi-dimension activity-travel patterns including trip chaining, trip characteristics, travel modes, destination choice, etc. Discrete choice models in different travel dimensions are executed sequentially, which works similarly to four-step models but with many more decision-making processes taken into consideration and improved behavior realism. In

general, econometrics-based ABMs can be further categorized into two classes based on the modeling and representation of daily activity patterns: *daily activity pattern choice* models and *activity-scheduling process* models.

The *daily activity pattern choice* models treat daily activity patterns as a choice modeling problem and comprise a nested logit model of daily activity patterns based on the hierarchy of trip purposes and trip frequencies. The *daily activity pattern choice* models can be further classified into two groups: “individual daily activity pattern” model and “coordinated daily activity pattern” models. The “individual daily activity pattern” models, first proposed by Ben-Akiva and Bowman (1998), follow the concept of an overall daily activity-travel pattern. This approach was then summarized as the day activity schedule model by Bowman (1998) in his dissertation; its applications include the Portland Metro Model (Bowman, Bradley, Shiftan, Lawton, & Ben-Akiva, 1999), SFCTA (San Francisco County Transportation Authority) Model (Bradley, Outwater, Jonnalagadda, & Ruiter, 2001), SACSIM (Sacramento Activity-Based Travel Simulation Model) (Bradley, Bowman, & Griesenbeck, 2010), etc. The “coordinated daily activity pattern” models enhance the “individual daily activity pattern” models by incorporating intra-household interactions during activity scheduling and modeling the correlation across the activity patterns in a household. This approach has been widely applied in models such as NYBPM (New York Best Practice Model;(Chiao, Mohseni, & Bhowmick, 2006), MORPC (Mid-Ohio Regional Planning Commission) Model (Parsons Brinckerhoff, 2005), ARC (Atlanta Regional Commission) Model (Parsons Brinckerhof, 2006), etc.

The *activity-scheduling process* models consider an activity-scheduling process where a sequential decision process is employed to yield daily activity patterns. Compared to the *daily activity pattern choice* models, which model the daily activity pattern choice at the top of the hierarchy, the *activity-scheduling process* models generate daily activity patterns at the end of the model as output. Applications of this approach include PCATS (Prism-Constrained Activity Travel Simulator; (Kitamura & Fujii, 1998) and CEMDAP (Comprehensive Econometric Micro-simulator for Activity-Travel Patterns; (Bhat, Guo, Srinivasan, & Sivakumar, 2004).

Despite the wide popularity of econometrics-based ABMs, this approach is criticized for two main issues: 1) utility-maximization theory might not apply to everyone and perfect rationality does not exist in reality; 2) this approach does not reveal the fundamental decision processes and behavioral mechanisms behind these travel-related decisions (Pinjari & Bhat, 2011). Consequently, researchers are working on alternative theory approaches to better mimic individual behavior process.

2.2.3 Rule-based models

Rule-based models, also referred to as computational process models (CPM), have been proposed as an alternative approach to relaxing the behavior assumption of the utility-maximization theory in econometrics-based ABMs. CPMs use heuristic productions,

typically in the form of if-then rules, to mimic the underlying decision-making process (Garling, Kwan, & Golledge, 1994). Even though rule-based models relax the RUM theory and better represent travelers' decision processes, none of them have been implemented for operation purposes. Two factors have contributed to this: 1) rule-based models require very detailed input data that are hard to collect; 2) the production rules employed in rule-based models are not valid and may be thought to lack rigor (Zhang, 2006). Examples of rule-based ABMs include SCHEDULER (Garling et al., 1994), SMASH (Simulation Model of Activity Scheduling Heuristics; (Ettema, Borgers, & Timmermans, 1993), AMOS (Activity Mobility Simulator; (Kitamura, Lula, & Pas, 1993), ALBATROSS (A Learning-Based Transportation Oriented Simulation System; (Arentze, Hofman, Joh, & Timmermans, 1999), etc.

2.2.4 Agent-based models

The agent-based modeling approach is a class of computational models for evaluating the system operation by simulating the actions and interactions of autonomous agents in this system (Odell, 2002). Even though this modeling approach has been widely applied in fields such as economics, business, network theory, etc., the application of this approach in travel demand modeling has not thrived until recently (Buliung & Kanaroglou, 2007). The agent-based approach is related to another modeling concept—multi-agent systems—which has been implemented in rule-based ABMs. While agents in both modeling approaches follow similar behavioral rules, the agent-based approach

allows the agents to learn, adapt and evolve during interactions with the environment and other agents, which is distinct from multi-agent systems (Bhat et al., 2004). Examples of agent-based ABMs include ALBATROSS in its second version (Timmermans & Arentze, 2005), MATSIM (Multi-Agent Transport Simulation Toolkit; (Balmer, Axhausen, & Nagel, 2006), SILK (Search, Information, Learning and Knowledge; (Zhang, 2006; Zhang & Levinson, 2004) and ADAPTS (Agent-Based Dynamic Activity Planning and Travel Scheduling; (Auld & Mohammadian, 2012).

2.3 Network Supply Models

Network supply models, also referred to as traffic assignment models, simulate how travel demand interacts with network supply in the transportation system. Specifically, network supply models concern the route choices between origins and destinations in transportation networks. Based on the end goal, traffic assignment models are typically categorized into two classes: user equilibrium (UE) assignment or system optimum (SO) assignment. UE assignment is based on Wardrop's first principle, which states that no driver can unilaterally decrease his/her travel costs by switching to another route (Wardrop, 1952). SO assignment is based on Wardrop's second principle, which states that drivers cooperate with one another to minimize total system travel time. In the context of travel demand modeling, UE assignment is often employed since results of this assignment are closer to real-world situations. To reach various assignment goals, different assignment algorithms are proposed, such as incremental assignment,

capacity restraint assignment, and method of successive average (MSA). While different assignment algorithms perform differently, the main idea is to find the route, and to link volumes and travel times that satisfy the equilibrium (or system optimum) condition, usually through iterative procedures.

One major hypothesis in traffic assignment is the resolution of representing traffic flow and conditions. Earlier traffic assignment models focused on representing the average conditions over a period of analysis time, which is known as static assignment. In these models, link volumes and travel times are assumed to be the same over the analysis period. Static traffic assignment models are widely applied in conventional four-step models because the mathematical properties of these models can be easily retained (Y. Chiu et al., 2011). However, static assignment models are criticized for the following issues: 1) time-dependent interactions of travel demand and network supply are not captured; 2) static assignment models are aggregated in nature, which does not consider the movement of each; 3) congestion building-up and dissipations are not modeled; and 4) they are not able to incorporate demand management strategies and intelligent transportation system technologies (Lin et al., 2008).

To resolve the aforementioned issues, researchers began to look into the concept of dynamic traffic assignment (DTA). DTA models can be regarded as performing static assignment over a very short period while the impacts of the previous period on the current period are captured. The equilibrium condition is still realized over each period, which is known as dynamic user equilibrium (DyUE, to be distinguished from

deterministic user equilibrium, or DUE). There exist two types of DTA models based on how the network loading process is modeled: analytical and simulation-based DTA models. Analytical DAT models usually use volume-delay functions to calculating travel times in the network, while simulation-based DTA models typically use mesoscopic simulations to estimate how traffic propagates in the network. Examples of simulation-based DTA models include VISTA (Waller & Ziliaskopoulos, 1999), DYNASMART (Mahmassani, 1992), DynaMIT (Ben-Akiva, Bierlaire, Koutsopoulos, & Mishalani, 1998), DynusT (Y.-C. Chiu, Nava, Zheng, & Bustillos, 2011) and DTALite (Zhou & Taylor, 2014).

2.4 Integrated ABM and DTA Models

Today, it almost comes naturally that traffic assignment is the last step in the travel demand modeling system. However, the integration of travel demand and network supply models was not proposed until Evans (1973) first combined the gravity distribution model with the equilibrium assignment model. It has become standard practice to consider the interaction between travel demand and network supply ever since. Recent years have witnessed significant development in both demand and supply aspects. Nevertheless, the advancement in both fields was accomplished somewhat independently. Earlier ABMs were developed with a static assignment process, which generates inconsistent results and undermines the real potential of ABMs. Likewise, using travel demand in aggregate time periods as the inputs for DTA does not achieve

the capacity of DTA models (Lin et al., 2008). Therefore, the integration of ABM and DTA helps exploit the full potential for both models mainly because of two reasons: 1) both models treat time as a continuum, and 2) both models operate at the individual level.

Earlier research focused on the mathematical formulation of integrating ABM and DTA, typically through fixed-point formulation approach (Cantarella & Cascetta, 1995; Lam & Huang, 2003; Lin et al., 2008). The integration of an operational ABM and DTA model was not proposed until recently. Pendyala et al. (2017) categorized the ABM and DTA integration approaches into the sequential integration and several levels of dynamic integrations, based on the data exchange frequency between the two models. In a sequential integration paradigm, the ABM and DTA models are loosely coupled, and only communicate and exchange data at the end of a full iteration (i.e., the entire 24-hour period of a day). This sequential information exchange procedure fails to capture the impacts of network disruptions or real-time information systems on travel behavior and demand. In recognition of this limitation, a tighter integration approach is proposed. In a dynamic integration paradigm, data exchange between the ABM and DTA models occurs at a much finer time resolution (e.g., one minute), which represents network dynamics in a more realistic way. Existing dynamic integrated systems include MATSim (Balmer et al., 2009), SimTRAVEL (Pendyala et al., 2012), and POLARIS (Auld et al., 2016). While dynamic integrated modeling systems provide enhanced modeling capabilities, the majority of the integration efforts in the literature fall in the sequential integration paradigm.

Various ABM and DTA integration works have been proposed, from a conceptual framework to an operational model system, in the literature. Within the scope of this dissertation, it is not feasible to provide a comprehensive review of the ABM and DTA integration literature. Consequently, this subsection focuses on reviewing the integration of implementation-ready ABM and DTA models at the metropolitan level. Particularly, several SHRP 2 Integrated Dynamic Travel Model (C10) funded projects are reviewed in detail. In project C10A, an ABM (DaySim), a DTA model (TRANSIMS), and an emission model (MOVES; the Motor Vehicle Emission Simulator) are integrated for the Jacksonville metropolitan area in Florida (Hadi, Pendyala, Bhat, & Waller, 2014). In project C10B, DaySim and MOVES are integrated with another DTA model, DynusT, for the Sacramento metropolitan area in California (Cambridge Systematics, 2014). Other on-going C10 funded projects include the Atlanta Regional Commission's (ARC) project that integrates CT-RAMP1 with DynusT; an Ohio Department of Transportation (ODOT) project that integrates CT-RAMP2 with DynusT; San Francisco County Transportation Authority's (SFCTA) project that integrates CHAMP with FastTrips; and Baltimore Metropolitan Council's (BMC) project that integrates TourCast with DTALite. Even though different ABM and DTA models are integrated, the core integration methodologies of these projects are similar and belong to the sequential integration paradigm, except for a tighter integration that was proposed for ARC and ODOT. The main challenges stemming from these projects include: 1) excessive runtime, with each full model iteration potentially taking days to run; 2) convergence challenge, as there is no consensus on

unified convergence criteria; and 3) an efficient data exchange procedure, which is needed to improve the model communication efficiency in order to save model runtime.

2.5 Behavioral Foundation

The behavioral foundation in travel demand modeling decides not only how each model component is formulated but also how the ABM and DTA models are integrated. Even though the behavioral foundation in ABM have evolved from perfect rationality (e.g., utility-maximizing) to bounded rationality (e.g., rule-based decision-making) paradigm, the behavioral foundation in DTA has focused on perfect rationality-based UE principles. This section reviews various behavioral theories ranging from rational behavioral theory to the positive theory of travel behavior with a focus on the BUE principle.

2.5.1 Perfect rationality and bounded rationality

As briefly introduced in Section 2.3, the UE principle, based on Wardrop's first principle, assumes that travelers have the same preference and perception, as well as perfect knowledge of all alternatives. Travelers seek to maximize utility or minimize generalized cost with perfect rationality in the decision-making process (Wardrop, 1952). The simplest form of the UE principle is deterministic user equilibrium (DUE). Because of its simplicity and stability, DUE is the most popular traffic assignment

method in practice, which is also the built-in method in all commercial transportation planning software packages (Sheffi, 1985). Several iterative algorithms have been proposed to solve DUE-based traffic assignments, such as the method of convex combinations (Frank-Wolfe algorithm; (Frank & Wolfe, 1956), the method of successive averages (MSA; (Almond, 1967) and origin-based algorithm (OBA; (Bar-Gera & Boyce, 2003).

Despite being built on the utility maximization theory, stochastic user equilibrium (SUE) principle relaxes some of the behavioral restrictions applied in DUE. While DUE assumes that all travelers perceive travel cost in an identical manner, SUE assumes that travelers do not have perfect information about travel cost due to random perception errors (Daganzo & Sheffi, 1977). Since a random error component is introduced to the utility structure (e.g., the generalized cost function), which is assumed to follow normal or Gumbel distributions, SUE problems are typically in the form of discrete choice models (i.e., probit or logit formulation). SUE problems are typically solved with a discrete choice formulation, together with an iterative network loading algorithm. STOCH (logit-based) approach proposed by Dial (1971) and the simulation (probit-based) approach developed by Daganzo and Sheffi (Daganzo & Sheffi, 1977) are the two traditional approaches to performing stochastic traffic assignment. Additionally, SUE implementations also concern the route choice set generation problem, which is also known as path enumeration, because finite choice alternatives are required for discrete choice formulations. Various route choice set generation methods have been proposed with different behavioral assumptions, such as

probabilistic choice set formations (Manski, 1977), labeling approach (Ben-Akiva, Bergman, Daly, & Ramaswamy, 1984), link elimination and penalty methods (de la Barra, Perez, & Anez, 1993), etc.

To further remove the unrealistic rationality assumption (i.e., utility maximization and perfect information) in both DUE and SUE, boundedly rational user equilibrium (BRUE) was proposed with an alternative behavioral theory based on Simon's work on bounded rationality (Simon, 1955). The BRUE condition in a transportation system is accomplished when all travelers have found a satisfactory travel option. In the context of choice modeling, instead of seeking a choice alternative that maximizes the utility, BRUE travelers look for a choice alternative that reaches certain utility levels, wherein the difference between this utility threshold and utility optimum is called indifference band (IB). Mahmassani and Chang (1987) first introduced the concept of BRUE to transportation in modeling departure time choices with bottlenecks. Since then, BRUE has been widely applied in various transportation problems, including traffic safety (Sivak, 2002), transportation planning (Khisty & Arslan, 2005), route choice (Han & Timmermans, 2006), etc. Despite that the boundedly rational behavior hypothesis has been verified by numerous simulation tests and empirical studies, it has not been mathematically formulated because of its non-uniqueness and non-convexity. Di et al. (2013) proposed to solve the BRUE problem in traffic assignment by formulating a nonlinear complementarity problem (NCP) so that the mathematical properties of BRUE can be analyzed.

2.5.2 Behavioral user equilibrium

Compared to the aforementioned rational behavioral theory, the concept of behavioral user equilibrium (BUE) is based on a positive theory of travel behavior, which removes the restrictions of impractical rationality assumptions discussed above. Zhang (Zhang, 2006) first proposed the BUE principle in his dissertation. Under this principle, travelers do not possess perfect information and travel decision-making is theorized as a continual search process that emphasizes the role of information acquisition, knowledge updating, and adaptive learning. Figure 2-1 illustrates the decision-making procedure in BUE. At any specific time, an agent has a certain level of spatial knowledge about external systems (e.g., places, infrastructures, and places of interest). When the external environment changes, such as an increase in commute time due to nearby construction, the agent is no longer satisfied with the current condition. The problem-solving process is composed of procedural steps akin to a real-world problem-solving situation. The agent first examines the self-beliefs to determine the subjective search gains from the alternative search. The efforts related to the search and information acquisition are modeled as the perceived search cost. The interplay of search gain and search cost decides if a search for a travel alternative is necessary. If the agent decides not to search, they will keep the status quo and execute habitual behavior. If the search process is invoked, this agent will identify which travel dimension (e.g., departure time, mode choice, destination choice, etc.) to search. Then, rules and heuristics are used to determine alternatives and to select an alternative to execute, if this alternative is sufficiently good. Eventually, when all agents in the

system are satisfied with their travel options and stop searching for new alternatives, the system is said to reach BUE.

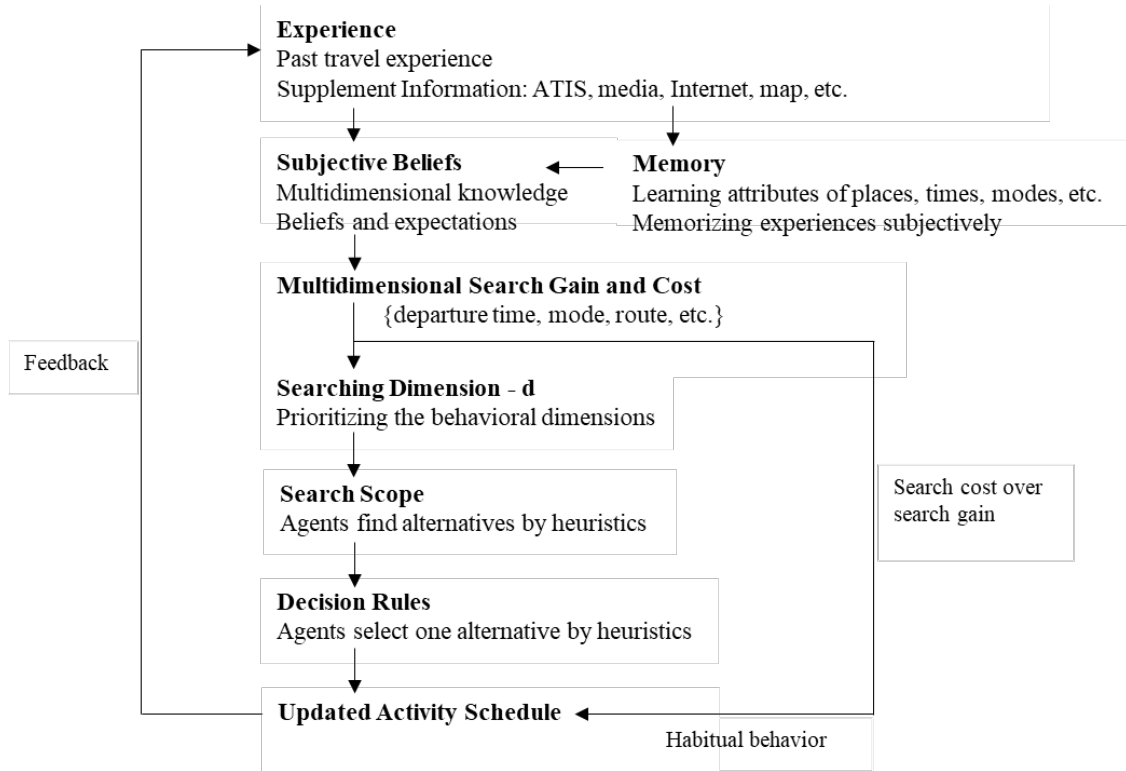


Figure 2-1 Decision-Making Process in BUE

Zhang (2006) applied the BUE theory to a route choice and traffic assignment model. Xiong (2015) later expanded the BUE theory to handle multi-dimensional travel behaviors including mode choice, departure time choice, and en-route diversion choice. Previously, the majority of travel demand models in the literature assumed that travel choices in different behavior dimensions took places in a pre-determined order. In Xiong's framework, however, travelers might choose the behavior dimension that is most rewarding instead of executing travel choices sequentially. An agent-based travel

demand model implementing the BUE theory has been developed by Zhang et al. (2013) and was applied to a case study in the I-270/-I495/I-95 corridor in the Northern Washington, D.C. metropolitan area. Even though a real-world application has been developed, the BUE theory still suffers from three main drawbacks: 1) the current BUE-based model relies on external travel demand models to provide the initial traffic condition, including population, socio-economic information, and travel patterns; 2) the BUE-based model is still trip-based, where trip chaining constraints are not considered; and 3) only three travel dimensions are included in the current model framework. Therefore, this dissertation proposes to integrate ABM and DTA models based on the BUE theory as an attempt to address these three drawbacks. The proposed integrated model is elaborated in the following chapters.

2.6 Summary

The past few decades have witnessed significant progress in travel demand modeling. From four-step models to ABMs and from econometrics-based approaches to rule-based and agent-based modeling techniques, the advancement in travel demand models is driven by emerging transportation policies, desired sensitivities to planning scenarios, and the pursuit of better behavioral realism. Meanwhile, the focus of travel demand models has shifted from pursuing aggregate-level prediction accuracy to understanding individual-level travel behavior, and from replicating observed activity-

travel decisions to explaining the underlying decision-making process. It has also shifted from the perfect rationality theory to more realistic behavioral foundations.

Similar evolution patterns have also occurred in the development of network supply models. DTA models have replaced static traffic assignment models to simulate individual-level movement and capture temporal interactions in traffic flow. However, the behavioral foundation of network supply models has not advanced significantly, as UE-based DTA models still dominate the field. Although bounded rationality-based traffic assignment models (i.e., BRUE) were proposed half a century ago, large-scale applications still face considerable challenges. This is also the case when it comes to integrating ABM and DTA models. Loose coupling between ABM and DTA models is not only the most straightforward integration approach but also the most popular one. The unrealistic behavioral foundation in the integration approach results in two major drawbacks: impractical convergence criteria and excessively long runtime. DTA models operate iteratively to reach UE condition, and then several feedback loops between ABM and DTA models must conduct to attain system-wide equilibrium. In real-world applications, it is typical to set a predefined feedback loop number due to runtime considerations, which results in model convergence never fully guaranteed.

With integration limitations summarized above, an integration approach that is based on a more realistic behavioral foundation is imperative. BUE, as a theoretically more practical principle, is a plausible behavioral foundation for ABM and DTA model integration. Different from existing integration approaches that emphasize system-wide

equilibrium criteria, BUE defines the equilibrium situation from an individual point of view; BUE is achieved when each traveler stops searching for travel decision alternatives because additional searches are no longer rewarding. This bottom-up equilibrium definition makes more sense from an agent-based modeling point of view. However, a BUE-based integration approach also faces significant challenges such as how to define search gain and search cost in the context of activity-based travel demand modeling, how to implement the BUE principle without additional data collection burdens, and how to improve the integrated model runtime systematically. Chapter 3 addresses these integration challenges and provides a theoretical framework for the BUE-based integration method.

Chapter 3 ABM and DTA Integration Framework

This chapter develops a theoretical framework that demonstrates how typical operational ABM and DTA models can be integrated based on the BUE principle. Previous operational integrated ABM and DTA models primarily follow a UE-based convergence criterion, which measures the network-wide travel time changes over iterations. In contrast, the new convergence criterion, namely BUE, measures the convergence from an individual point of view. Under the BUE principle, each agent in the system is assumed to go through a search process (i.e., an iterative process). At each search iteration, each agent chooses one of the four travel behavioral dimensions (i.e., destination choice, time-of-day choice, mode choice and route choice) to search for alternatives that could potentially improve his/her travel experiences. The BUE is reached when all agents in the network are satisfied with their travel options and stop searching. Only a portion of the population will change their travel behavior, meaning that the ABM only needs to rerun for a proportion of the population at each search iteration. Moreover, the DTA model does not need to run multiple iterations to reach the UE condition at each search iteration, which further saves the overall runtime. In addition to the runtime savings, the BUE-based integration framework employs a theoretically more advanced convergence measure, which could provide better behavioral realism.

The first section of this chapter describes a typical sequential ABM and DTA integration framework, which is used as a benchmark framework. The second section introduces the proposed BUE-based integration framework. Technical details regarding model components, workflow, and convergence criterion are also illustrated in this section. Section 3 to 4 further explains how two key concepts in the search process—search gain and search cost—are defined and calculated in the framework. Finally, the last section summarizes this chapter by discussing the advantage and value of the proposed integration framework.

3.1 A Sequential Integration Framework

This section introduces a typical sequential ABM and DTA integration approach. In this approach, the ABM and DTA models interact with each other, passing trip patterns and travel times back and forth until a measure of convergence is achieved. The general framework is illustrated in Figure 3-1. The integrated model runs in iterations, with a single “big loop” consisting of an iteration of the travel demand model, followed by a run of the DTA model. The travel demand information (e.g., agents with their activities and travel decisions, as well as characteristics) is passed to the DTA model within each big loop. Travel time information resulting from the dynamic assignment is passed back to the travel demand model for use as input (e.g., update activity pattern choices) in the next big loop. Note that DTA models themselves are run iteratively to reach UE, so there are several “small loops” of the DTA run within each “big loop”. A number of “big loops” are run until a measure of convergence is achieved, in which the change in

travel times (or some other measure) from one big loop to the next is within a specific tolerance. This methodology seeks practical convergence between the models since absolute convergence may not be achieved due to inherent differences at the various modeling levels.

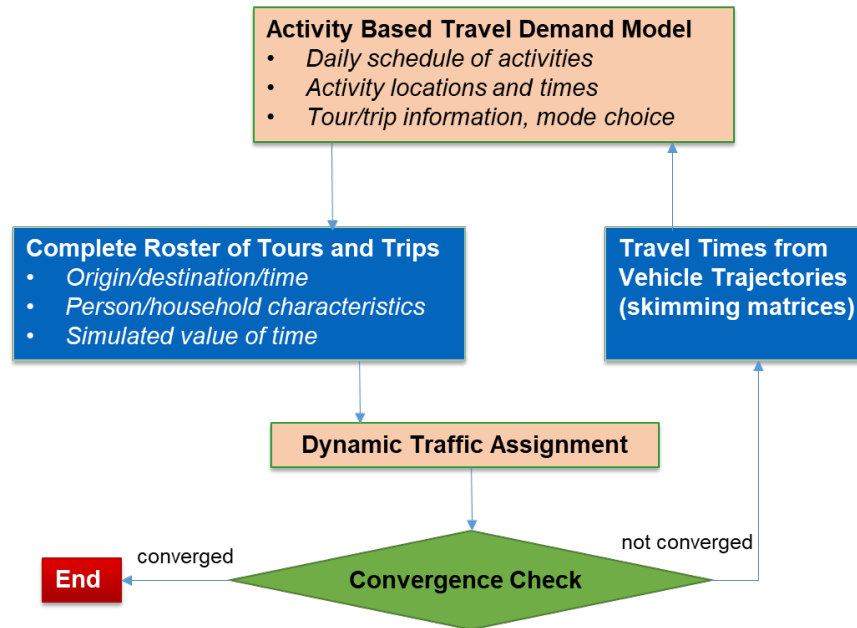


Figure 3-1 Sequential Integration Framework

3.2 A BUE-Based Integration Framework

3.2.1 Framework overview

As reviewed in Section 2.5, the BUE principle is built upon positive theory and agent-based modeling. To implement the BUE principle while integrating ABM and DTA

models, several major components are supplemented or changed on the basis of the traditional sequential integration approach. First, the proposed BUE-based integration framework is realized by means of agent-based modeling. Each traveler in the system is treated as an intelligent agent that is able to learn, adapt, and interact with the environment (e.g., transportation network). Second, the BUE principle assumes that, in contrast to the rational behavioral theory, each agent does not possess perfect information. Therefore, this positive theory of travel behavior emphasizes the role of information, searching, and learning in travel decision-making.

For a better understanding, one can draw an analogy between the proposed framework and house hunting. When a household first enters the housing market, they may have insufficient knowledge about the market and they may search several candidate houses before picking one. They could consult a real estate agent or search for information online. During the search process, the household gradually becomes familiar with the housing market and other important characteristics of different neighborhoods such as accessibility, school districts, and safety, which results in enhanced spatial knowledge about the area. The final purchase decision is based on the information and knowledge accumulated during the search process.

Similarly, the proposed BUE-based framework begins with an initial condition, which can be considered as the state that all individuals arrive at within the system the first time, with very limited knowledge about the environment (Figure 3-2). Each model iteration in the framework can be viewed as a day in real life. As days go by, agents

gradually become familiar with the system by memorizing their own travel experiences and learning through information from mass media, the internet, etc. They may search for new travel options, change their travel behavior, or adjust their own expectations until they are satisfied with the travel pattern. When all agents are satisfied with their travel options and no longer search for new alternatives, the system reaches BUE.

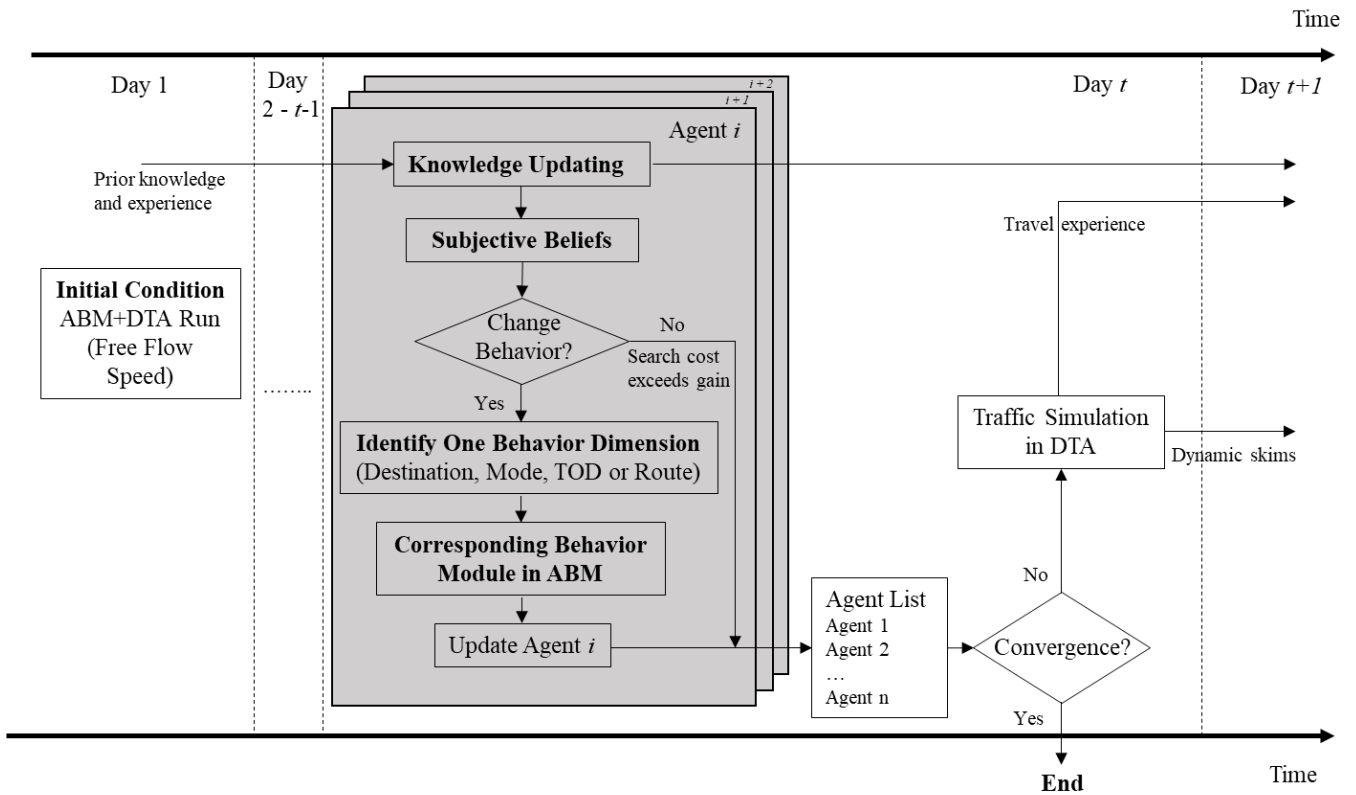


Figure 3-2 BUE-Based Integration Framework Illustration

At any given day t , an agent has an existing level of knowledge about places, activities, and transportation facilities in the system. To make the travel plan for the day, this agent i will undergo a decision-making process involving several behavioral steps.

First, this individual looks at his/her beliefs, which leads to the subjective expectation of gains from searching for alternative travel options. For instance, this individual may reduce travel time by switching to a different route or save travel cost by changing the travel mode. Self-beliefs might come from previous experiences or secondary information sources including maps, internet, media, or other individuals. Information acquisition and mental efforts involved in the search process can be generalized as the perceived search cost. The tradeoff in the subjective search gain and the perceived search cost determines whether this agent will conduct another search. The concept and calculation of search gains and search costs in the context of ABM are elaborated in the following sub-sections.

To implement the BUE principle in integrating ABM and DTA models, it is hypothesized that long-term choices (e.g., auto ownership, work location, etc.) and daily activity pattern (DAP) are not included in the day-to-day behavior adjustment process (highlighted in Figure 3-2). Specifically, long-term choice components are only executed in the first iteration and these choice results are kept unchanged throughout the rest of the model iterations. In the proposed framework, only short-term behavior dimensions are considered in the behavior adjustment phase, such as destination choice, time-of-day choice, mode choice, and route choice. In this multi-dimensional setting, agents will compare all behavior dimensions and choose the one with the highest gain/cost ratio, namely the most rewarding dimension.

If the perceived search cost exceeds the subjective search gain, this agent would decide not to search and execute habitual behavior. Otherwise, a search process is invoked. In the original BUE theory proposed by Zhang (Zhang, 2006), the search process involves a set of rules or heuristics to identify and choose travel alternatives. In the context of ABM, however, the search process can be realized by running the corresponding behavioral module in the ABM. For instance, if an individual decides to search for mode alternatives, the mode choice module in the ABM will be executed to identify and choose mode alternatives for this traveler.

When no agent in the system searches for new alternatives, the model is said to be converged and the BUE state is reached. Otherwise, the DTA model will be executed with the revised travel plans. Agents in the system will update their knowledge based on the new network travel times generated by the DTA simulator, and a new round of the search process will execute with new network information and updated self-expectations. The whole procedure iterates until BUE is reached.

3.2.2 Knowledge and learning

When the travel alternative search need arises, an individual typically uses existing knowledge to solve this problem. Specifically, it is the spatial knowledge about locations, activities, and the transportation system that help with the decision-making. In contrast to the traditional rationality-based behavior theory, where people are

assumed to have perfect information about the system, people in BUE theory only possess a certain level of the spatial knowledge, which is clearly more realistic when people make travel decisions.

Zhang (Zhang, 2006) first proposed a quantitative description of spatial knowledge, which essentially makes the decision-making process quantifiable in BUE theory. The spatial knowledge is generalized as multi-dimensional vectors, where each vector represents a particular travel dimension (e.g., mode choice, departure time choice, etc.). Suppose that an individual's perception of a particular behavioral dimension d is based on a specific attribute, such as generalized cost. This generalized cost, which is assumed to fall into several categories, is used to quantify the effect of each travel experience. For example, a person may use the following generalized cost categories: 0~5 dollars, 5~10 dollars, etc. If the generalized cost c_i corresponding to category i has been observed k_i times in prior experience, the individual's knowledge regarding this behavioral dimension can be described by a vector $K = (k_1, \dots, k_i, \dots, k_I)$. This specification of spatial knowledge is the premise of developing a quantitative travel behavior model and it is supported by empirical evidence in Zhang's research (Zhang, 2006).

In addition to designing the structure of spatial knowledge, we must understand how travelers update their spatial knowledge through learning. Bayes' theorem is applied here to describe the knowledge updating process. When a new travel option belonging to category i is experienced by a traveler, the new knowledge becomes $(k_1, \dots, k_i +$

1, ..., k_I). Let p_i denote the probability that a new round of search in dimension d would lead to an alternative with generalized cost c_i , and the individual's subjective beliefs can be represented by a vector $P = (p_1, \dots, p_i, \dots, p_I)$. These probabilities satisfy the following conditions:

$$\sum_{i=1}^I p_i = 1; \quad p_i \geq 0 \quad \forall i \in I \quad (3-1)$$

To quantitatively link spatial knowledge $K = (k_1, \dots, k_i, \dots, k_I)$ and subjective probability $P = (p_1, \dots, p_i, \dots, p_I)$, it is assumed that individuals' prior beliefs and knowledge follow a Dirichlet distribution. The probability density function is:

$$P = \frac{\Gamma(N)}{\prod_{i=1}^I \Gamma(k_i)} \cdot \prod_{i=1}^I p_i^{k_i-1} \quad (3-2)$$

where N is the total number of observations and $\Gamma(k_i) = (k_i - 1)!$ (gamma function). According to the strong law of large numbers, this assumption is equivalent to Eq. (3-3) as the sample size N grows.

$$p_i = \frac{k_i}{N} \quad (3-3)$$

Individuals' spatial knowledge $K = (k_1, \dots, k_i, \dots, k_I)$, together with the current condition, will determine their probability to search for new travel alternatives. Two important concepts regarding the search process—subjective search gain and perceived search cost—are illustrated in subsequent sections.

3.2.3 Subjective search gain

The search decision is based on an individual's past experience and subjective beliefs. Clearly, it is not feasible for a person to search for all alternative options in the real world. Consequently, the stopping criterion must be specified, which leads to the definition of two concepts: search gains and search cost. The search process continues when the individual observes benefits from the process, namely the subjective search gain exceeding the perceived search cost. Similarly, the search process stops when the search gain fails to meet search cost.

The subjective search gain is based on subjective beliefs and prior travel experiences. Search gain is also multi-dimensional. An additional search may lead to not only a decline in travel time, but also a reduction in monetary cost or improvement in subjective perception (e.g., safety, comfort, etc.). The subjective search gain sg_d is defined as the expected improvement on the generalized cost savings from an additional search, which can be calculated as:

$$sg_d = \sum_{i(\forall c_i < c)} p_i \times (c - c_i) \quad (3-4)$$

where c denotes the current generalized cost for the currently used travel option. Since travelers will select the best travel option for all experienced ones, c actually is always the lowest generalized cost c_{min}^d in dimension d . Let c^* represent the theoretically lowest generalized cost under the free-flow travel condition when all individuals initially believe there is no congestion. The subjective probability of finding a travel alternative with the generalized cost c^* becomes $1/(N + 1)$ after N searches. Therefore, Eq. (3-4) can be simplified as:

$$sg_d = \frac{c_{min}^d - c^*}{N + 1} \quad (3-5)$$

In the multi-dimensional setting, while the free-flow generalized cost c^* stays identical across all dimensions, the generalized cost c_{min}^d of the current best travel option in dimension d might be distinct in each dimension. Therefore, the traveler may search for alternatives in different dimensions at each iteration. This unordered search process is more realistic than a pre-determined decision sequence that is typically seen in traditional travel demand models.

Generally, the subjective search gain declines as the number of searches N grows. Once a better travel option is found (i.e., c_{min}^d decreases), the expected search gain also

decreases. Meanwhile, the framework is also sensitive to changes in external conditions. For instance, if new links in the network are open, travelers are more likely to search for new alternatives, since c^* decreases and their expected search gain increases.

3.2.4 Perceived search cost

The perceived search cost exists because it takes time and efforts to acquire information and search for travel alternatives. While the perceived search cost might be different for different travelers, it remains the same for a particular traveler. Furthermore, the perceived search cost is assumed to be constant for the same person throughout the search process.

Theoretically, the perceived search cost can be derived based on Eq. (3-5) by calculating both the lower and upper bounds. If a traveler stops searching after N rounds of search, the perceived search cost for this traveler must be lower than the expected search gain at search $n - 1$ and higher than the expected search gain at search n to ensure that search $n + 1$ does not happen. Therefore, we can calculate the lower and upper bounds of the search cost sc for behavioral dimension d and let the average be the estimate of the perceived search cost:

$$\begin{aligned}
sc_{d,low} &\geq g_{d,n} = \frac{c_{min,n}^d - c^*}{N + 1} \\
sc_{d,high} &\leq g_{d,n-1} = \frac{c_{min,n-1}^d - c^*}{N}
\end{aligned} \tag{3-6}$$

$$\overline{sc}_d = \frac{1}{2}(g_{d,n-1} + g_{d,n})$$

The distribution of perceived search costs among travelers can be obtained by calculating the cost for each traveler, according to Eq. (3-6). To further predict perceived search costs for all agents in the system, a subsequent regression analysis is needed to reveal the relationship between search costs and travelers' socio-demographic information. Xiong (2015) collected travel behavior data from a stated adaption survey and empirically estimated a regression model to predict the search cost. Eq. (3-7) specifies the search cost model in dimension d as:

$$\begin{aligned}
sc_d = &\beta_0 + \beta_1 c_0 + \beta_2 gender + \beta_3 purpose + \beta_4 income \\
&+ \beta_5 dist + \beta_6 peak + \beta_7 veh + \epsilon_i
\end{aligned} \tag{3-7}$$

where c_0 is the generalized cost for the initial travel experience; *gender* equals 1 if the person is male, 0 otherwise; *purpose* equals 1 if the trip purpose is mandatory, such as work and school trips, 0 otherwise; *income* is the annual household income, which includes income classes of less than \$50,000, \$50,000-\$100,000 and above \$100,000; *peak* equals 1 if the trip is in the peak period; *veh* equals 1 if the household owns more than two vehicles. It is worth noting that multicollinearity phenomena may exist in the

regression analysis; for instance, covariate trip purpose and income might be correlated. Follow-up research should attempt to address the multicollinearity issue by adding interaction terms. However, the focus of this regression analysis is for prediction purposes rather than fully explaining the relationship between the search cost and socio-demographic variables. In other words, as long as the model is able to predict the search cost with reasonable accuracy, this analysis could live with the current model specification.

In the current multi-dimensional context, the search cost is calculated independently for different behavior dimensions. However, interdependence may exist when calculating the search cost. For example, the search cost for destination choices may impact the search cost for mode choices since the distance between origin and destination obviously will affect mode choices. This issue can be resolved by fitting a multivariate multiple regression (MMR) model to capture the potential correlation among the dependent variables.

3.2.5 Model estimation

The estimation of these quantitative models can be data intensive. Traditional travel surveys only provide individual travel behavior information for a specific day or two. Longitudinal information, such as the number of days that a particular traveler takes to search for travel alternatives and traffic conditions for these days, are typically missing.

Therefore, new data sources must be explored. The BUE model was first estimated with data from a stated adaption experiment conducted by Xiong (2015). Details regarding the survey design and data collected are presented in Section 4.1. Specifically, the stated adaption experiment imitates the search process and asks participants how they would change travel behavior under various hypothetical scenarios. The participants keep switching for alternative travel options until they are satisfied with certain travel plans, and their choices and associated traffic conditions are recorded. With the generalized cost information collected for various travel dimensions, the search gain sg_d at different iteration n can be calculated according to Eq. (3-5). Information regarding when participants stop searching can also be derived from the survey; thus, the search cost sc_d can be calculated by taking the average of the lower and upper bounds according to Eq. (3-6). Lastly, the proposed regression model can be estimated using the socio-demographic information recorded by the survey and the search cost calculated above. Specifically, generalized method of moments (GMM) and two-stage least-squares (2SLS) estimators were employed in Xiong's study.

The focus of this dissertation is to develop an ABM and DTA integration framework that is so flexible that it can be easily transferred to anywhere as long as operational ABM and DTA models exist locally. However, the search cost model parameters would not be the same for different regions, as people's perceptions and travel behavior vary from place to place. Local data are needed to customize the search cost model.

Subsequent sections introduce several possible data sources for localization, as well as a model calibration approach with minimal data requirement.

3.3 Model Calibration

To implement the decision-making process in the proposed BUE integration approach, several important model parameters must be estimated. However, parameter estimation requires travel behavior data, which is typically costly and time-consuming to collect as discussed above. More importantly, the proposed integration framework focuses on integrating existing operational ABM and DTA models that are already well calibrated and validated. Consequently, additional data collection efforts are certainly not preferred. To streamline the implementation of the integration framework, as well as promote the model transferability, this study proposes to calibrate model parameters in the integration framework. It is worth noting that the search cost parameters previously estimated by Xiong (2015) are used as the initial values for the calibration process to ensure a fast convergence.

Model calibration is the process of adjusting model parameters so that the simulated model results closely represent the observed information (e.g., traffic counts, link travel times, trip production rate, etc.). This process is critical to enhancing the model performance and improving the result accuracy. The first subsection provides the technical details regarding the model calibration process, including optimization

problem formulation and the objective function. The second subsection introduces the optimization algorithm, simultaneous perturbation stochastic approximation (SPSA), and illustrates how the calibration method works.

3.3.1 Problem statement

To implement the BUE-based integration approach, model parameters must be re-estimated based on local data. Since both ABM and DTA models in this study are fully operational models, parameters in both models do not require further calibration. Based on the BUE theory, parameters in the search cost model are the only ones that call for calibration. Search costs are calculated using travelers' socio-demographic data as demonstrated in Eq. (3-7). In the search cost model, two parameters (θ_0 and θ_1) are introduced to calibrate the function in each travel dimension, as shown in Eq. (3-8). Since the search process in the BUE theory involves four travel dimensions (i.e., destination choice, departure time choice, mode choice, and route choice), a total of eight parameters must be calibrated. Note that model parameters in the destination choice dimension use mode choice parameters as the initial values, since destination choice dimension did not exist in the previous study. The reason behind this treatment is the similar behavioral trend between these two travel dimensions.

$$\begin{aligned}
 sc_d = \theta_0 \cdot \beta_0 + \theta_1 \cdot (\beta_1 c_0 + \beta_2 gender + \beta_3 purpose + \beta_4 inc \\
 + \beta_5 dist + \beta_6 peak + \beta_7 veh + \epsilon_i)
 \end{aligned}
 \tag{3-8}$$

In general, the process of a system-wide calibration of a simulation-based model is to find the parameter values that minimize the error between observed measurements and simulated measurements. Specifically, the observed measurements are traffic counts. In other words, the SPSA algorithm seeks to minimize the difference between observed traffic counts and simulated link volumes. Even though the search cost model involves four travel dimensions, only link volume information is included in the objective function since it represents the overall model performance. Local survey data can be used to validate simulated results from three other travel dimensions, such as destination choice, TOD choice, and mode choice. The calibration process is formulated as a constrained minimization optimization problem, as shown in Eq. (3-9).

$$\begin{aligned}
 & \text{minimize}_{\theta} (M_o - M_s(\theta))^2 \\
 & \text{s. t.} \\
 & M_s = f(Z; \theta) \\
 & l \leq \theta \leq u
 \end{aligned} \tag{3-9}$$

where M_o represents the observed traffic counts; M_s are simulated link volumes; θ is the vector of parameters from the model to be calibrated, and l and u are vectors of lower bounds and upper bounds for the parameters; $f(Z; \theta)$ represents the link between the simulated outputs and the simulation-based model; and Z are the inputs required to run the simulation-based model.

One should note the complexity associated with the calibration approach proposed in this study. The search cost model is concerned with four behavior dimensions; changes in each dimension would lead to changes in the overall results. It is difficult to include all four behavior dimensions in the objective function as it is often challenging to reach optimal solutions in multi-objective optimization problems. This study involves the final model results in the objective function, as typically executed in the literature. Therefore, deviations in other travel dimensions (i.e., mode choice, destination choice, and TOD choice) are expected in the study. In this study, results in these travel dimensions are validated against survey data to ensure the overall performance of the integrated model.

3.3.2 Simultaneous perturbation stochastic approximation (SPSA)

The optimization problem in this study is a simulation-based optimization problem. The simulated measurements are acquired directly from the integrated model, which makes the optimization problem non-analytical and the non-linear problem with extensive system noise. Additionally, the runtime of the integrated model is significantly long. Considering these factors, this study selected the simultaneous perturbation stochastic approximation (SPSA; (Spall, 1998) algorithm to solve the minimization problem. This algorithm has the following advantages: 1) it accounts for the simulation noise in the simulation-based model output; and 2) it only requires two function evaluations per iteration regardless of the length of the vector of parameters.

SPSA optimization algorithm has been successfully applied to various problems in the transportation sector, especially to the calibration of DTA models (Antoniou, Azevedo, Lu, Pereira, & Ben-Akiva, 2015; Lee & Ozbay, 2009; Lu, Xu, Antoniou, & Ben-Akiva, 2015; Ma, Avenue, Ave, & Zhang, 2007).

Essentially, the SPSA algorithm works by perturbing the components of the vector of parameters and computing the gradient of the objective function based on the perturbations. The implementation step of the algorithm is briefly described below:

- Step 0 Initialization and coefficient selection: in this step, the SPSA algorithm is set up with the initial values for the vector of parameters θ_0 and also the values for the non-negative algorithm coefficients a, c, A, α , and γ . In this study, the initial values of the search cost parameters θ_0 are obtained from a study conducted by Xiong (2015), which could contribute to the fast convergence in SPSA algorithm. The algorithm coefficients belong exclusively to the SPSA algorithm. At iteration k , the SPSA gain sequences $a_k = a/(A + k)^\alpha$ and $c_k = c/k^\gamma$ are critical to the performance of the SPSA algorithm and their values are discussed in the Section of calibration results.
- Step 1 Generation of simultaneous perturbation vector: a p -dimensional perturbation vector Δ_k is generated using Monte Carlo simulation. Typically, the random perturbation vector Δ_k is Bernoulli-distributed, with a probability of 0.5 for each of the two possible outcomes.

- Step 2 Objective function evaluations: evaluate the objective function twice using the perturbation vector at each iteration k : $f(\hat{\theta}_k + c_k \Delta_k)$ and $f(\hat{\theta}_k - c_k \Delta_k)$
- Step 3 Gradient approximation: compute the gradient approximation $g(\hat{\theta}_k)$ using the perturbation vector and two evaluations of the objective function computed from Step 2:

$$g(\hat{\theta}_k) = \frac{f(\hat{\theta}_k + c_k \Delta_k) - f(\hat{\theta}_k - c_k \Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \dots \\ \Delta_{kp}^{-1} \end{bmatrix} \quad (3-10)$$

where Δ_{ki} is the i th component of the perturbation vector.

- Step 4 Update vector of parameters: update the values of the vector of parameters based on the gradient descent using the approximated gradient computed from Step 3 and the step size a_k from Step 1:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k * g(\hat{\theta}_k) \quad (3-11)$$

- Step 5 Iteration or termination: return to Step 1 to continue iterating or terminate if there is negligible change between iterations in the objective function and/or values of the vector of parameters.

While this section provides technical details concerning the SPSA algorithm, the selection of SPSA coefficients and actual calibration results are illustrated in Chapter 5.

3.4 Discussion

This chapter introduces two ABM and DTA integration approaches with a focus on the BUE-based integration approach. The BUE theory was initially developed in the context of the SILK framework, where behavior adjustment modules are rule-based Artificial Intelligence (AI) models. This chapter emphasizes illustrating how the BUE theory is adjusted and improved in the context of ABM and DTA integration. Specifically, four main aspects of BUE have been enhanced:

- First, the original BUE theory requires an initial traffic condition and synthetic population, which is typically from an existing travel demand model. The BUE-based ABM and DTA integrated model, however, can provide the initial state by running the model based on free-flow travel times without relying on outside data sources.

- Second, the original BUE theory is built on the trip-based environment. Since most ABMs are tour-based models, the BUE theory has incorporated trip-chaining constraints so that it can fit into the tour-based setting. Technical details regarding this BUE improvement are illustrated when implementing the integration framework to a real-world application in Chapter 5.
- Third, only three behavior dimensions (i.e., departure time choice, mode choice and route choice) are considered in the original framework. This study incorporates a fourth dimension (destination choice) into the framework, which enhances the behavioral adjustment process.
- Last, a model calibration approach is proposed to streamline the search cost estimation process. This approach provides great transferability without additional behavior data collection efforts.

In addition to the aforementioned BUE improvements, this chapter also introduced how key elements such as search gain and search cost are calculated in the context of ABM and DTA integration. The proposed BUE-based integration framework is applied to a large-scale network in the Washington-Baltimore Metropolitan area. Chapter 5 presents details regarding the real-world model application and analyzes the application results.

Chapter 4 Data

Previous travel demand models are typically estimated with household travel survey data, which is mostly concerned with only one cross-section of the study population. However, the estimation of the proposed BUE model requires longitudinal information that is often missing in the literature. For example, to study each person's search process, we have to understand what conditions trigger the search behavior and how many searches it takes this person to become satisfied with the travel options. In other words, we must monitor this person's travel behavior over time to extract the longitudinal information needed to estimate the BUE model.

This chapter introduces two longitudinal data sources and one calibration data source for enhanced transferability purposes. First, we can design a stated adaptation experiment to record participants' behavioral responses to hypothetical traffic conditions. Section 4.1 illustrates how such experiments are designed and how longitudinal information can be inferred from the experiment. However, such experiments can be both expensive and time-consuming. As an alternative, this study develops an approach to calibrate the BUE model parameters. This calibration approach requires a previously estimated BUE model and calibrates it based on existing local data. Section 4.2 describes the data requirements for this calibration approach. Lastly, the ideal longitudinal data source is from repeated observations where the actual behavioral dynamics and the learning process can be revealed. Section 4.3 introduces

how passively collected data is a good source of individual-level longitudinal data. While some travel behavior information is easier to infer from passively collected data—such as departure time, destinations, and trip durations—other information is difficult to detect, like travel modes and trip purposes. Consequently, it requires significant data mining and processing efforts to derive all trip-related information from passively collected data. As a proof of concept, Section 4.3 demonstrates how travel mode information can be detected with a case study in the Washington-Baltimore area. While imputing all travel behavior information is beyond the scope of this dissertation, future studies should examine how passively collected data could supplement or even replace traditional behavior surveys.

4.1 Data from Behavioral Surveys

The BUE model emphasizes the role of searching and learning in explaining individual behavioral dynamics. As a result, it is no longer sufficient to estimate such a model with only cross-sectional data. For instance, to calculate the search cost for a traveler, we must know when this traveler stops searching for alternatives as discussed in Section 3.2. While this longitudinal travel behavior information is typically missing in the literature, the behavioral survey is a good alternative and could reveal the traveler's decision-making adjustment over time. Xiong (2015) designed a stated adaption experiment to collect behavior data and empirically estimated regression models to predict search costs. The stated adaption survey consisted of a number of hypothetical

questions such as “what would you do if you encountered specific conditions.”. It was carried out in an iterative fashion where the behavioral responses of a person regarding different inexperienced situations could be recorded. For instance, a survey participant first reported his/her most recent trip, together with their socio-demographic information recorded as well. Then the travel condition was assumed to alter due to changes in exogenous policy or congestion level. The survey participant had to adjust to these changes by searching for new travel alternatives. Since the search process in Xiong’s study was multi-dimensional, alternatives in travel modes, departure time, and routes were simulated. With the new information, this person had to choose between his/her habitual plan and the new alternative, and state the reasons behind this decision. The process was repeated iteratively until the participant was satisfied with the travel experience. In this way, a complete behavioral adjustment procedure of each participant can be recorded. The survey was conducted on a sample group of 110 people that performed adaptations under schemes like overall congestion increases and road-pricing scenarios.

With the longitudinal data collected from the survey, two important variables—search gain and search cost—can be empirically calculated, as discussed in Section 3.2. Additionally, the travel survey also records participants’ socio-demographic information. The regression model, which relates the search cost and socio-demographic information, therefore can be estimated. While the BUE model parameter estimation can be easily carried out using the survey data, the cost and time associated

with the survey remain a big issue. Additionally, the transferability of this method is rather weak since similar behavioral surveys must be conducted locally.

4.2 Model Calibration Data

Due to the high cost and processing complexity of the data sources mentioned above, this dissertation proposes a parameter calibration method that requires only existing local data and provides great transferability. This model calibration approach, however, requires acquisition of the initial parameter values of a previously estimated model so that the calibration can converge more rapidly. In this study, the search cost regression model parameters estimated by Xiong (2015) were used as the initial values for the calibration process.

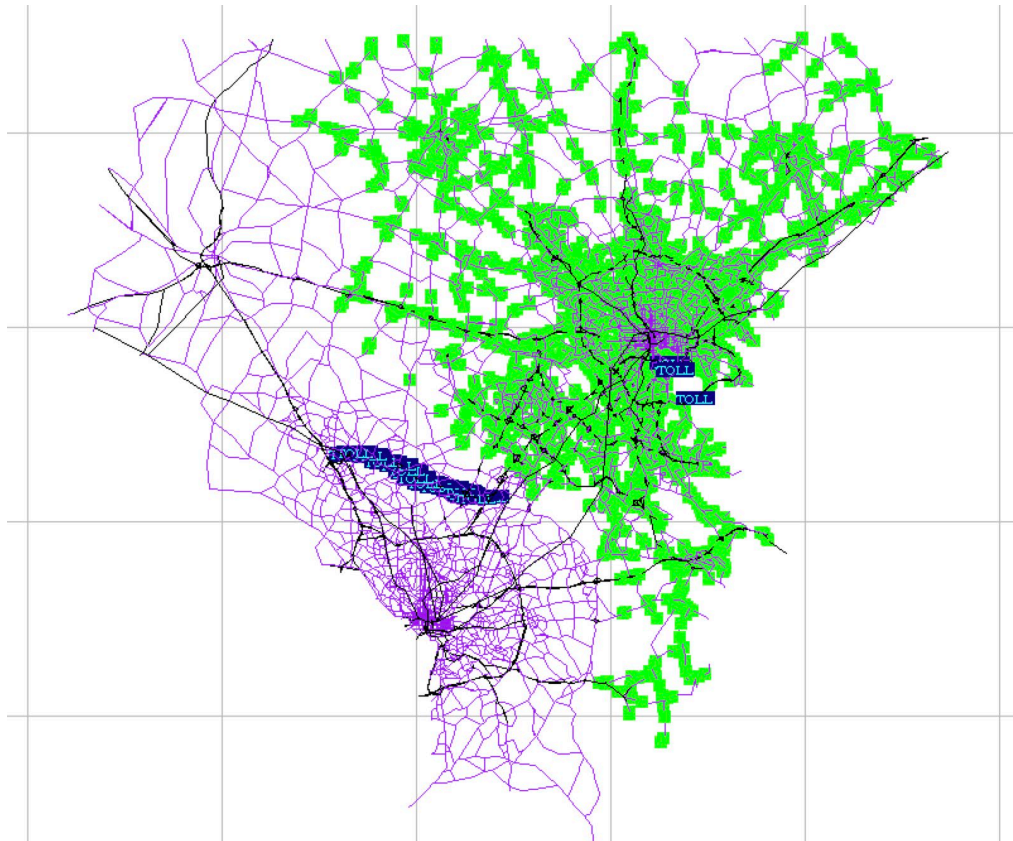


Figure 4-1 Traffic Count Sensor Location

Traffic count data are typically required for model calibration. The calibration attempts to minimize the difference between simulated link volumes and traffic counts. In this study, annual average daily traffic (AADT) information was obtained from the Maryland State Highway Administration (SHA) in 2009, covering 3,770 links within the InSITE network. As shown in Figure 4-1, the small green squares represent the location of link traffic count sensors within the network.

In addition to traffic count data, local travel survey data are also required to validate intermediate results. For instance, trip length distribution information from the travel

survey can be used to validate destination choice results. Mode share information can be used to validate mode choice results. Departure time distribution information of various trip purposes can be used to validate time-of-day model results. Specifically, this study utilizes the 2007/2008 Household Travel Survey (HTS) jointly conducted by National Capital Region Transportation Planning Board (TPB) and BMC. This travel survey collected information from 14,365 households in the Washington and Baltimore metropolitan area. A two-day travel diary was recorded for each member in the household. Detailed behavior information at the trip level is available from the HTS data. Descriptive statistics are presented in Table 4-1.

Table 4-1 HTS Descriptive Statistics

Variables	Mean	Std. dev.
Household size	2.18	1.21
# of workers	1.18	0.85
# of vehicles	1.73	1.05
# of lic. drivers	1.63	0.75
Residential location (1=urban)	0.29	0.45
Housing tenure (1=own)	0.77	0.42
# of observations	14,365	

This model calibration approach provides great transferability where existing local ABM and DTA models can easily be converted to BUE-based paradigm without additional data collection efforts. In this dissertation, SPSA (Simultaneous Perturbation Stochastic Approximation) optimization method is selected for the calibration task. Calibration and validation results are presented in Chapter 5.

4.3 Passively Collected Data

The aforementioned behavioral survey can be considered as a stated preference method, where inaccuracy might exist when people state their preferences. As a result, the perfect data would be the panel data, where travelers' travel information are actually recorded over time. Imagine that there exists a device that follows you 24/7 to record information about your every single trip over a long period of time. This ideal data source would provide you insights on how you change travel behavior so that the learning process can be modeled. While a data collection method of this kind might be impossible to implement, passively collected data could be an excellent alternative to track people's behavior dynamics.

Thanks to the popularity of smartphones today, passively collected data are within easier reach than ever. In its simplest form, passively collected data are GPS location data (trajectories) that are typically collected using smartphones. While smartphones are present at almost every time and place, the deficiency of this data source is quite obvious: we do not know the actual trip information. It is nearly impossible to bother users to provide details regarding every single taken trip. Therefore, data mining and machine learning methods are typically employed to extract useful travel behavior information. However, this approach requires significant data processing time and existing data mining methods cannot guarantee the detection of all necessary trip information. As a proof of concept, this section presents a case study in the Washington,

D.C. area, which elaborates how trip mode information can be extracted from smartphone GPS data.

4.3.1 Introduction

Thanks to the rapidly growing smartphone industry, passively collected travel data has never been so readily available. According to the Pew Research Center, the United States had around 223 million smartphone users in 2017. Roughly three-quarters of Americans (77%) now own a smartphone, with lower-income Americans and those ages 50 and older exhibiting a sharp uptick in ownership over the past year. The widespread use of Global Positioning System (GPS)-based technologies, GPS loggers, GPS-enabled phones, etc., provides an innovative but accurate approach to observe and track individuals' travel behavior. Compared to traditional data-collecting activities, GPS-based technologies play a leading role in passively collecting a large amount of accurate spatial and temporal information without spending considerable time and money. The fast development of connected vehicles and autonomous vehicle technologies ensures the continued influx of GPS data. This advent of GPS big data requires technologies and research for processing and utilization to serve our life better. To optimally use GPS data, we must be able to infer multiple trip information, such as travel modes and trip purposes. In this study, we discuss how to do this solely with passively collected GPS data and without asking people additional questions.

Mode detection based on GPS raw data drew increasing research attention in the past decade while GPS technology has been widely used to collect large-scale transportation data. Researchers have used AI techniques to handle mode detection, including decision trees (Stenneth, Wolfson, Yu, & Xu, 2011), neural networks (Byon & Liang, 2014; Gonzalez et al., 2010), Naïve Bayes and Bayesian networks (Xiao, Juan, & Zhang, 2015), random forests (Lari, 2015), etc. Wu, Yang and Jing (2016) have conducted a thorough review of existing studies on this topic. Overall, the current practices can detect car mode with high accuracy but the detection accuracy of bus/metro/subway modes is not satisfactory.

This section develops a random forest model to impute travel mode information and calculate variable importance rankings. The model is empirically tested on a GPS dataset collected through GPS-enabled smartphone devices. To effectively detect the travel mode for each trip, classification feature construction is critical in providing useful information, preferably travel mode-specific knowledge. In addition to the traditional features used in the literature (e.g., average speed, maximum speed, trip distance, etc.), this study constructed two innovative features based on land use data: the distance to the closest rail line (both underground and aboveground) and the distance to the nearest bus line. Even though Stenneth et al. (Stenneth et al., 2011) first proposed to use transportation network information in travel mode detection, they did not consider metro (underground) detection and bus line information. Within the Washington Metro system, GPS locations can be recorded at Metro stations, which

makes underground metro detection possible. To the best of my knowledge, this study is the first to use land use data to infer metro mode that is typically underground.

4.3.2 Methodology

Based on previous studies in rule-based models and decision trees (Tang, Xiong, & Zhang, 2015; Xiong & Zhang, 2013), the random forest model is selected for this mode detection research. A random forest is an ensemble of decision trees (Ho, 1995). The training packages for random forest are typically included in most existing software or platforms for artificial intelligence (e.g., RandomForestClassifier in Python or R).

The general idea of random forests is to combine multiple decision trees built from different samples generated from the training data set using the bootstrap sampling method. For classification problems, the predictor is based on the majority voting of different trees. For regression problems, the predictor is formed by taking the average of different trees. When building each decision tree, at each node, a given number of features are randomly selected and the best split is calculated from the selected features. Since no pruning step is taken for the tree, all of the trees are maximum trees.

There are two critical parameters when training the random forest model: the number of trees in the forest, or *ntree*, and the number of input variables randomly chosen at each split, or *mtry*. One benefit of random forests is that it will not over-fit as more

trees are added (Louppe, 2014). If more statistics are expected from decision trees, like variable importance, then *ntree* can be set to 1000 or more to make the statistics stable. For *mtry*, a value that equals the square root of the number of features is typically first used. Then, attempt a value twice as high and half as low and check the out-of-bag (OOB) error. OOB error refers to the mean prediction error for each training sample, using only the trees that have the sample in their bootstrap sample. It is suggested to set *mtry* higher when many noise variables are present.

4.3.3 Data collection

The data was originated from research analyzing the travel behavioral impact of the 2016~2017 Washington D.C.'s Metro SafeTrack project. SafeTrack is a series of 16 maintenance surges that address safety recommendations for the Metro system, which lead to significant service disruptions to different Metro lines in the Washington area. To assess the impact and analyze the travel behavior responses, the research authors have conducted joint web-smartphone surveys with over 2,000 Metro users. A smartphone application has been designed to record GPS location data for each survey subject. The survey app functions are illustrated in Figure 4-2:

- 1) *GPS location tracking*: the app automatically records the user's location information. The frequency of recording is automatically adjusted based on whether the user is moving or static to save battery consumption. Typically, the

time interval between two location records is 30 seconds when the user is moving and 10 ~ 30 minutes when the user is static, depending on the battery status.

- 2) *Trip information logging*: the app periodically “pushes” survey questions to record trip purposes and the travel modes for the user’s recorded trips. This information is verified by a follow-up travel diary survey and then is used in this research as the ground-truth travel modes to train the mode detection.

- 3) *Data uploading*: for the sake of battery and cellular data usage, the app will not automatically upload data to the online database unless the device is plugged in and connected to a Wi-Fi network. Alternatively, the user could manually upload survey records by pressing the button “Press to Upload”.

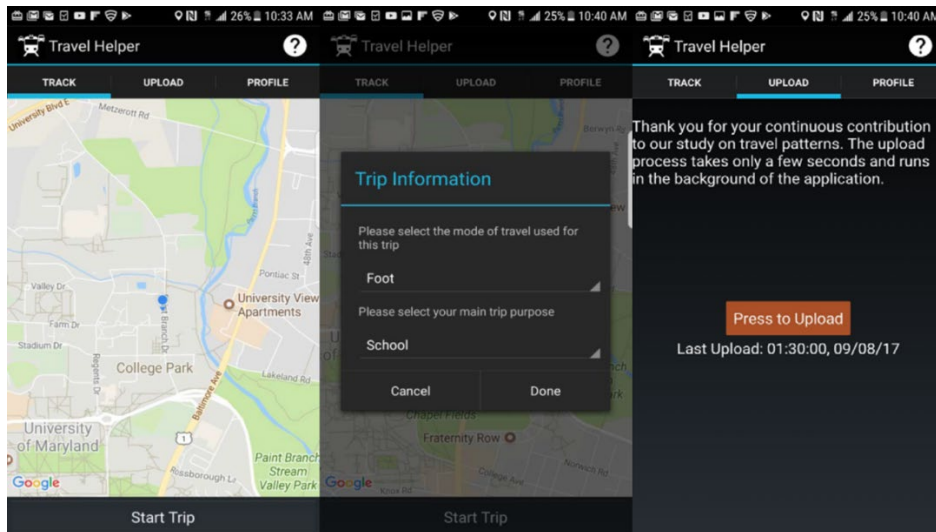


Figure 4-2 The User Interface of Smartphone GPS Data Collection App

A total of 865 trips are specified with travel mode information and these data are used for mode detection modeling in this study. Of these 865 trips, 19.31% are auto trips, 15.84% are bus trips, 52.94% are metro trips and 12.37% are walk trips. Since this survey was targeted towards Metro users, a high percentage of Metro trips are captured. During the survey, only three trips are reported as bike trips; these trips are excluded from this study due to the small sample size.

4.3.4 Data processing

Data Filtering:

In this study, the location point data collected in this study has information including latitude, longitude, the instantaneous speed, accuracy and timestamp. The collected raw GPS location data are filtered based on two criteria: accuracy and the average speed between two successive location points. Accuracy indicates the closeness of a measured location to the real location of the device at the time of the measurement, which is vital in assessing the quality of the location data. The authors first filter the data based on the accuracy and remove location data with accuracy that is larger than 500 meters as an attempt to get rid of inaccurate data points. To further eliminate infeasible travel patterns, a location point is removed if the average speed between this point and last point is faster than 150 meters per hour. This is to discard data noises (e.g., sudden “jumps” in location) and improve data quality.

Trip End Identification:

To impute travel mode information, trip end information must be extracted from a series of GPS location points. The trip end identification method in this study is similar to the approach proposed by Tang, Pan, & Zhang (2018). A trip end is identified as the first and last location point in a stay region. In this study, a stay region is defined as the region where the user has stayed longer than a time threshold T , within a distance range of D and under a speed limit V . A set of successive location points $P_l = \{p_0, p_1, \dots, p_k\}$ are labels as a stay region if they satisfy the following constraints:

$$\begin{aligned}\Delta d_{0i} &\leq D, \forall i \in P_l \\ \Delta t_{0k} &\geq T \\ v_i &\leq V, \forall i \in P_l\end{aligned}\tag{4-1}$$

where Δd_{0i} denotes the distance difference between the first location p_0 and any location p_i in the location set, Δt_{0k} is the time difference between the first and last location points, and v_i represents any speed at the location p_i . Consequently, location p_0 is the trip end of the last recorded trip and p_k will be the trip start of the following trip.

The Construction of Classification Features:

As described in the modeling section, typical trajectory features are employed in our empirical testing. The following table summarizes the trajectory features.

Table 4-2 Mode Detection Data Description of Trajectory Features

Variables	Descriptions
Trip distance	The trip distance is computed as the sum of the distances between two successive location points in this trip
Trip time	The difference between the timestamps of the trip start and the trip end
OD Euclidean distance	The shortest Euclidean distance between the origin and destination of the trip
Average speed	The average speed is calculated as the trip distance divided by the trip time
Max. instantaneous speed	The maximum value in the set of instantaneous speeds directly collected by the smartphone app during the trip
Average data record	The number of data points recorded during the trip divided by the trip time

These features are selected to differentiate the modes as much as possible. For instance, the average speed can be used to distinguish walk mode from other modes. The maximum instantaneous speed further helps differentiate walk trips from auto or bus trips that encounter severe traffic congestion, which makes the average speed of those trips close to walking trips. The overall data recording frequency can be used to identify Metro trips as other travel modes typically do not suffer from significant GPS disruptions.

4.3.5 Empirical results

The random forests model results are shown in Table 4-3. The overall model accuracy is 91.33%, with the highest 96.11% accuracy for Metro and lowest 86.15% for bus. Metro trips are generally easy to detect for two reasons. First, underground Metro trips could only obtain GPS signals at Metro stations, meaning that the total number of data points recorded in Metro trips is much smaller than the other modes given the same travel distance. Therefore, the average data record feature could easily distinguish Metro trips from other modes. Second, the distance to the Metro line system further helps detect Metro trips. However, bus trips are relatively difficult to identify and often confused with driving trips. Even with land use information, the detection accuracy could not reach 90%. Future studies should focus on seeking new features that could discover bus trips.

Table 4-3 Overall Model Statistics

Accuracy:	0.9133			
95% CI:	(0.8834, 0.9376)			
P-Value [Acc > NIR]:	< 2.2e-16			
Statistics by Class:				
	Auto	Metro	Bus	Walk
Sensitivity	0.8804	0.9719	0.7667	0.8571
Specificity	0.9860	0.9502	0.9564	0.9825
Pos Pred Value	0.9419	0.9603	0.7302	0.8571
Neg Pred Value	0.9698	0.9646	0.9638	0.9825
Balanced Accuracy	0.9371	0.9608	0.8673	0.9167

In this study, the prediction accuracy of 10-fold cross-validation is used to measure the performance of the random forests model. For each round of the validation, 10 random seeds are used to ensure the stability of the validation results. Table 4-4 summarizes the prediction accuracy of the proposed model.

Table 4-4 Confusion Matrix

		Detected Travel Mode				Recall:
		Car	Metro	Bus	Walk	
Reported Travel Mode	Car	149	1	10	2	92.55%
	Metro	3	466	3	3	97.90%
	Bus	7	14	98	4	79.67%
	Walk	0	4	2	99	94.29%
Precision:		93.71%	96.08%	86.73%	91.67%	91.33%

Compared to existing studies on mode detections, the level of accuracy that the proposed model emits is among the top. According to the review paper by Wu et al. (Wu et al., 2016), most studies reported an overall mode detection accuracy at around 88%~93%, while the reported highest level of accuracy is 96% (Lari, 2015). Nevertheless, it is worth noting that this research is focused on the mode detection of likely Metro users. Unlike typical studies that largely draw data from auto trips (e.g., about 80% of the testing data are driving trips in Lari(Lari, 2015), the testing data has relatively more balanced distribution among car, metro, bus, and walk. Also, in most research papers, classifying auto trips has much higher accuracy compared to detecting bus or metro trips (Nitsche, Widhalm, Breuss, Brändle, & Maurer, 2014). It is not yet possible to draw a conclusion on which models perform the best without extensive

examination and testing using the same datasets. However, the high prediction accuracy resulted from the random forests model clearly shows its potential in handling mode detection, especially for its generalization power on the trivial differences among car, metro, and bus trajectories.

4.4 Summary

This chapter introduces two ways to derive the longitudinal data required to estimate the proposed BUE-based integrated model: behavioral surveys and passively collected data. Additionally, a calibration approach can be applied when a previously estimated BUE model exists. The data requirement for this calibration approach is specified in this chapter. While this chapter reveals only travel mode information from passively collected data, the methodology provided is general enough to detect other trip-related information. Tang (2018) applied a similar methodology to infer trip purpose information from smartphone GPS data. Ongoing research in our research group is also examining how random forest models could impute traveler's socio-demographic information. With the popularization of smart devices and maturity of data mining methodologies, passively collected data has the potential to replace traditional travel surveys and provide data for advanced travel demand models.

Chapter 5 Integrated Model Application in the Washington-Baltimore Area

This chapter showcases a model demonstration. The proposed BUE-based ABM and DTA integration framework is applied to a large-scale network in the Washington-Baltimore metropolitan area. The Washington, D.C. area ranks No.1 in traffic congestion according to the 2015 Urban Mobility Scorecard (Dooley, 2015), which makes it highly challenging to model the massive traffic in this large-scale network. The traditional sequential ABM and DTA integrated model for the Washington-Baltimore metropolitan area would take 30 hours per iteration on average, including 30 DTA iterations. The long runtime remains a significant hinderance for real-world implementations. However, the BUE-based integration approach could accelerate the integrated model while maintaining the result accuracy. On the contrary, the advanced behavioral theory foundation proposed in the framework could lead to enhanced behavior realism. The real-world integration results are analyzed and discussed in this chapter.

This chapter is organized as follows. The first section introduces two major components of the integration application: InSITE ABM and DTALite. Section 5.2 illustrates the overall integration process regarding software implementation. Section 5.3 demonstrates the advanced computing technology employed to accelerate InSITE and DTALite including multi-threading and tree-based route storage. Section 5.4 analyzes

and discusses real-world model results. The last section provides conclusions and summarizes this chapter.

5.1 Major Model Components

5.1.1 BMC InSITE ABM

InSITE is an activity-based model system composed of interconnected, discrete choice models representing choices at distinct dimensions (e.g., travel mode, destination) that focus on decisions related to daily activity and mobility for a typical weekday. InSITE adopts the day activity-schedule approach, where a daily activity schedule is defined through the concepts of activity pattern and activity schedule. The activity pattern defines the participation in activities as primary and secondary. Primary activities are the anchors (e.g., home-to-work trip and work-to-home trip represent a tour, with work as the primary activity) of a tour, and secondary activities are intermediate stops within a particular tour (i.e., stopping for shopping during the work-to-home half of the tour). The activity schedule adds detailed information to the activity pattern about tours, such as the timing, travel mode, destination of primary activities, and the stops for secondary activities within the tours.

The model covers an area that includes the entire Baltimore Metropolitan Council (BMC) region, plus the District of Columbia and the Maryland portion of the region

covered by the Metropolitan Washington Council of Governments (MWCOG). The proportion of Maryland in the model region consists of Baltimore City and Anne Arundel, Baltimore, Carroll, Harford, Howard, Frederick, Montgomery, and Prince George's Counties.

InSITE models travel for a typical weekday. The choices made by households and individual travelers are simulated using probabilities from a series of logit models. As shown in Figure 5-1, the model begins by simulating long-term choices that are made before the travel day, including auto ownership, workplace location for workers, school location for students, transit pass ownership, and E-ZPass toll transponder ownership.

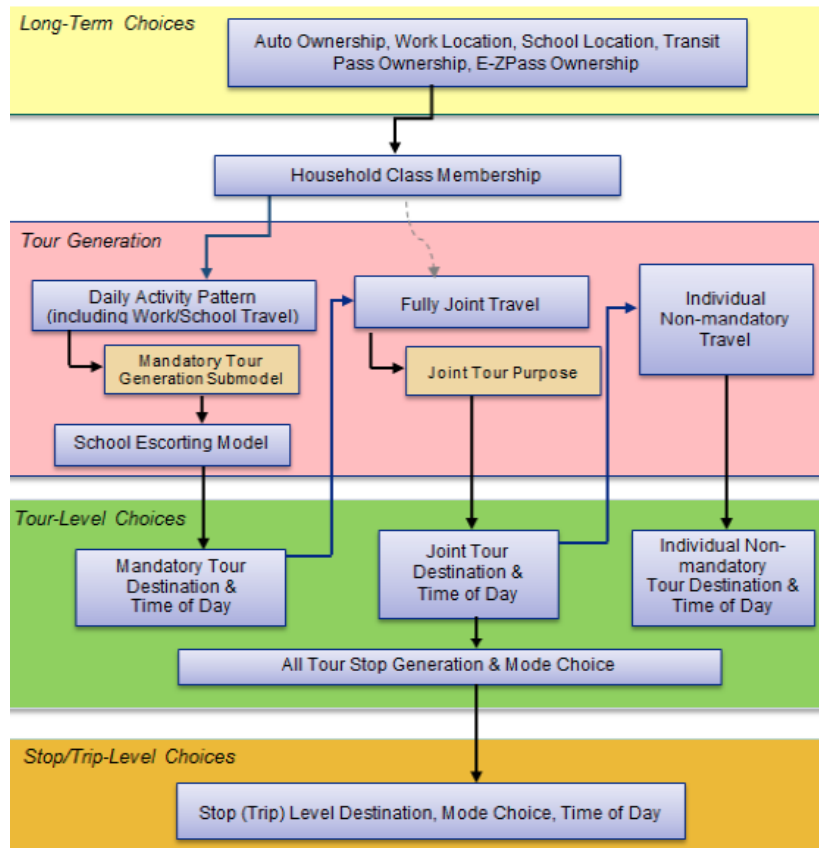


Figure 5-1 InSITE Model Structure

Another feature of InSITE is a class membership model that is applied prior to the daily activity pattern model. The class membership model determines which segment an individual household belongs to via a multinomial logit model. Each class membership model alternative represents a distinct segment, and the model uses attributes of the household (such as household size and income) to generate probabilities that the household is a member of a specific class. The parameters of the daily activity pattern model, the fully joint tour model and the class membership model were estimated simultaneously since there are components of each model that affect one another. A more in-depth look at the class membership model is documented by Lemp (2014).

The next set of models estimates a daily activity pattern for each person. Whether the person has work tours (with or without stops), school or university tours, non-mandatory activity tours only, or does not travel within the region, is simulated. If the simulated pattern has mandatory (work or school) tours, they are generated. For students making school tours, InSITE simulates whether they are escorted by a household member and if so, by whom. At this point, the destinations and times of day for the mandatory tours are simulated.

Next, fully joint tours among household members are simulated, including who participates, the activity purpose, the destination, and the time of day. After the details of the mandatory and joint tours are known, individual, non-mandatory tours are simulated, with their destinations and time of day. The final tour level models are the generation of stops for each tour and tour level mode choice.

The stop- and trip-level choices are simulated next. These include destination choice for each stop, the times for the stops, and mode choice for the trips between stops (conditional on tour mode). From the results of these models, auto and transit trip tables are assembled for input to highway and transit assignment, respectively.

5.1.2 DTALite

DTALite, an open-source, light-weight, mesoscopic DTA simulation package; in conjunction with the Network eXplorer for Traffic Analysis (NeXTA) graphic user interface (GUI), it has been developed to provide transportation planners, engineers, and researchers with a theoretically rigorous and computationally efficient traffic network modeling tool. DTALite adopts a new software architecture and algorithm design to facilitate the most efficient use of emergent parallel (multi-core) processing techniques and exploit the unprecedented parallel computing power newly available on both laptops and desktops.

The overall structure of DTALite, illustrated in Figure 5-2, integrates the four major modeling components, highlighted in yellow. They are:

- 1) time-dependent shortest path finding, based on a node-link network structure;
- 2) vehicle/agent attribute generation, which combines an origin-destination demand matrix with additional time-of-day departure time profile to generate trips;
- 3) dynamic path assignment module, which considers major factors affecting agents' route choice or departure time choice behavior, such as (i) different types of traveler information supply strategies (e.g., historical, pre-trip, and/or en-route information,

- and variable message signs), and (ii) road pricing strategies where economic values are converted to generalized travel time, and;
- 4) a class of queue-based traffic flow models that can accept essential road capacity reduction or enhancement measures, such as work zones, incidents and ramp meters. The queue-based traffic simulation model in DTALite only requires basic link capacity and free-flow speed for operation, which are readily available from static traffic assignment models. By using simple input parameters, in addition to possible connections with common signal data interfaces, the proposed simulation package may enable state DOTs and regional MPOs to rapidly apply advanced DTA methodologies for large-scale regional networks, subareas or corridors. Additionally, the modularized system design may help serve future needs by simplifying the process for transportation researchers and software developers to continue to build upon and expand its range of capabilities.

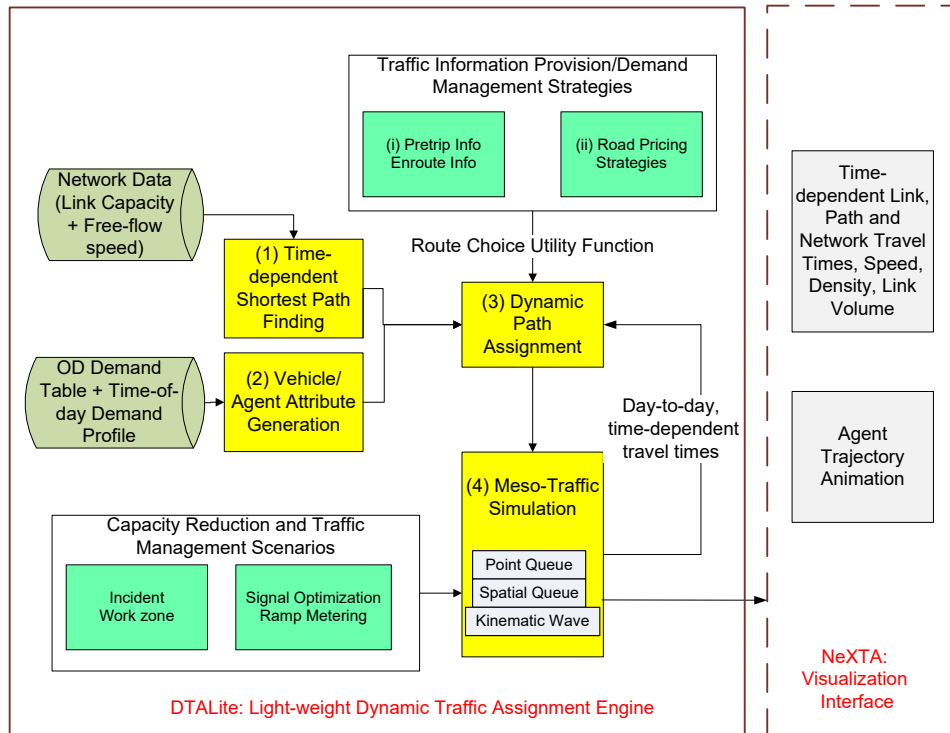


Figure 5-2 DTALite System Architecture with Key Modeling Components
 (Source: (Zhou & Taylor, 2014))

The traffic assignment and simulation modules are fully integrated and iterated to either capture day-to-day user response or find steady-state equilibrium conditions. Within this simulation-assignment framework, the rich set of output data include traffic measures of effectiveness (MOEs) at different spatial and temporal scales, ranging from network, corridor-level, and specific links. Typical speed, volume, and density measures, as well as agent-based trajectories, can be visualized through the NEXTA user interface. Based on the design structure and queue-based mesoscopic traffic simulation model, DTALite has considerable potential for generalizing the modeling framework into the field of real-time traffic state estimation and prediction.

As a powerful mesoscopic DTA simulator, DTALite is often applied to large-scale networks. Existing path-based methods only store link volumes of the current iteration. When the origin-destination (OD) demand volume changes, DTALite must calculate from the beginning, which is extremely time-consuming. Currently a tree-based version of DTALite featuring rapid re-optimization is in development. The new DTALite version has the following features that will dramatically improve the computing speed: 1) storage of all the shortest path tree for all iterations; 2) re-optimization function, which finds the new shortest path tree based on the existing base tree; and 3) large-scale parallel computing capability (i.e., OPENMP and MPI). This tree-based DTALite model could achieve a major speed boost: the Maryland statewide network (i.e., 1674 zones, 170,000 links) runtime is only 3 seconds per iteration on an 8-thread machine.

5.2 Model Integration Implementation

5.2.1 Overall process

The overall integration structure is illustrated in Figure 5-3. To execute various model components in one platform, a Python program is developed as a wrapper to call different modules. In addition to the major components InSITE ABM and DTALite, the integrated model also includes several intermediate modules such as search process module, convergence check module, agent update module and skim script.

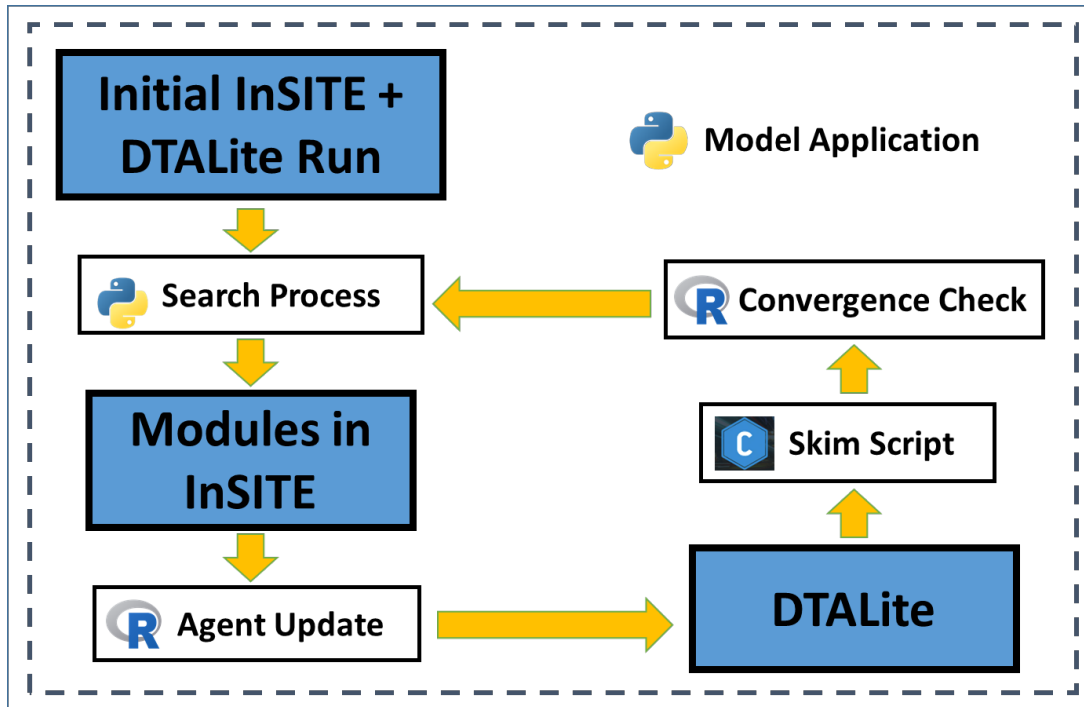


Figure 5-3 Integration Framework Structure

The framework starts with an InSITE and DTALite run as the initial condition, which imitates the state when all agents arrive at the system for the first time and are unfamiliar with the environment. Next, the search process module calculates the search gain and search cost for every agent in the system and determines whether an agent would search for alternative travel options, and which behavioral dimension to search if this agent decides to search for alternatives. The integrated model then calls different behavioral modules in InSITE ABM to calculate new travel options for the agents that choose to search. In this study, four behavioral dimensions are considered: destination choice, time-of-day choice, mode choice and route choice. The agent trip roster is then updated with new trip options and fed to DTALite to simulate updated travel time on the network. Since InSITE and DTALite are developed in different coding languages,

a skim script module is built to convert the dynamic DTALite skim file format to be compatible with InSITE. Finally, the convergence check module checks if the model has converged by measuring how many agents in the system are still searching. The next model iteration will start if the convergence has not been reached.

5.2.2 Information exchange between models

In the BUE-based integration framework, the ABM and DTA model exchange information when the model run is complete. At the end of each InSITE run, the program will output a complete roster of trip information and traveler characteristics (e.g., origin/destination, time of travel, value of time (VOT), etc.). One feature of InSITE is that each traveler in the population has a simulated VOT obtained from the VOT population density functions. These simulated VOTs are passed from InSITE to DTALite to use in its route choice models, which could also support the analysis of tolling scenarios. Compared to previous integration efforts, where the ABM outputs three separate files containing information regarding the trip, person, and household, respectively, this DTA roster combines all information in one file, which improves the data exchange efficiency. This roster later becomes the input to DTALite so that the network representation can be simulated.

The main data passed from DTALite to InSITE is related to highway travel times. In the integrated model, DTALite simulates highway travel dynamically over the entire

day from beginning to end. Since InSITE uses 48 aggregate time periods of 30 minutes in length, DTALite creates link travel times for each 30-minute period by averaging the times experienced by the vehicles (from trip trajectories) in the DTA during the period. It should be noted that free flow travel times are employed for the initial iteration. After the initial iteration, the travel time inputs to InSITE can be unique for each half-hour period.

5.2.3 Towards a tour-based BUE model

The original BUE theory was developed based on a trip-based diagram. In other words, no trip chaining constraints were explicitly considered previously. However, InSITE ABM is a tour-based travel demand model. Consequently, the BUE theory needs to be revised accordingly. Specifically, the following key assumptions are proposed regarding implementing the search theory in a tour-based environment.

- Tours are independent
- Mandatory tours (work/school) will not change destination choices
- If non-mandatory tours change destinations, stops conditional on the tour will change accordingly
- If tours/stops change TOD, time constraints apply (based on minimum activity duration)

- If tours change TOD/modes, the stops conditional on the tour will change accordingly
- Travel modes of stops are the same as tours

5.3 Accelerating InSITE ABM and DTALite

One major motivation of this dissertation is the excessively long model runtime that hinders the large-scale implementation of the integrated ABM and DTA model. Even though the proposed BUE-based integration framework has significantly reduced the integrated model runtime, there is still potential for further runtime reduction in the individual model component. This section illustrates how advanced computing technologies are implemented to speed up InSITE ABM and DTALite.

5.3.1 Multiprocessing in InSITE ABM

InSITE, as an econometrics-based ABM, calculates travel options for each traveler in the model using discrete choice models. In other words, InSITE model runtime is positively related to the number of travelers in the model. Realizing this fact, a parallel computing program is developed to speed up the InSITE ABM. In this dissertation, the whole population is divided into sub-groups so that multiprocessing functions can be performed to run discrete choice models on these population sub-groups in parallel. Specifically, a Python program is used to call multiple InSITE runs simultaneously and

each of the InSITE runs processes one population sub-group. The multi-processing feature is realized by utilizing Python package Subprocess. Python scripts are attached in Appendix A for further reference.

The model speed is constrained by CPU power. The current single-process version of the InSITE ABM occupies around 25% of CPU power in the UMD workstation. As a result, the multi-processed InSITE can theoretically run as much as four times faster than the single-processed InSITE. Due to multi-processing, the model runtime is positively related to the CPU power. The more cores/processors a machine has, the faster the multi-processed InSITE could run on this machine. Consequently, the multi-processed InSITE could run even faster with a machine that has more CPU capability than the current UMD workstation.

5.3.2 Time-dependent skim generation in DTALite

Even though DTALite only performs one iteration under the BUE-based integration framework, it must provide time-dependent skims to the InSITE ABM for each iteration. The travel time skims are at 30-minute intervals and require 48 skim tables to generate every iteration. With the built-in skim calculation function in DTALite, this skim generation process takes around 2.5 hours per iteration. For a 48-hour integrated model run with 15 iterations, DTALite alone consumes 37.5 hours, which takes about 78% of the overall integration model runtime.

This dissertation develops a Python program to produce time-dependent skims much faster. This Python program takes advantage of the time-dependent link travel time from the “output_LinkMOE.csv” file, which is readily available from DTALite. The program calculates the time-dependent skims directly using the time-dependent link travel time information based on the shortest path algorithm, which runs much more efficiently than the built-in skim generation function in DTALite. This Python program shortens the skim generation time from 2.5 hours to around 45 minutes per iteration, which saves 26 hours for the overall integrated model run.

5.4 Calibration and Validation Results

As introduced in Section 3.3 Model Calibration, this dissertation develops a model calibration approach to re-estimate the search cost model parameters. SPSA algorithm is selected to solve the optimization problem. The sum of the squared differences between the observation information and model results is used as the objective function. SPSA seeks to minimize the objective function by the simultaneous perturbation of model parameters. This study uses parameter values previously estimated from data in the same region as the initial values, which could lead to quicker model convergence.

5.4.1 SPSA calibration

An important aspect of the calibration is the selection of the five SPSA algorithm coefficients: a , c , A , α and γ . Spall (Spall, 1998) has suggested the default values for these coefficients. However, coefficients a and c must be adjusted locally based on the objective function and the average of gradients to ensure an appropriate step size. Several rounds of calibrations were conducted; Table 5-1 summarizes values of the SPSA coefficients used in this study.

Table 5-1 SPSA Coefficients

Coefficient	Value
A	100
a	3E-11
c	0.1
α	0.602
γ	0.101

The SPSA algorithm is coded in Python and the code is attached in Appendix B for the reader's reference. As introduced in Section 3.3, eight parameters (θ_0 and θ_1 in four behavior dimensions) are calibrated in this calibration process. The initial search cost model coefficients β are inherited from the parameter values in the previous study conducted by Xiong (2015). The initial values of θ_0 and θ_1 are 1.0.

The algorithm convergence is reported in Figure 5-4. Thirty iterations were conducted in this study, largely due to the runtime constraints. Two objective function evaluations are required in the SPSA algorithm, which makes 60 integrated model runs in the calibration process. Each integrated model run takes around 24 hours and details regarding runtime breakdown are provided in the subsequent section. Fortunately, the initial state achieved by performing InSITE and DTALite, which takes around four hours, does not need to repeat for each iteration. In other words, each SPSA iteration takes about 20 hours. The calibration process was terminated at iteration 30 when the overall percent difference between observed and simulated traffic counts falls within 10%. Although the objective function represents the sum of squared differences between simulated link travel time and traffic counts, only the percent difference is reported in Figure 5-4 since it is easier to observe the true model performance.

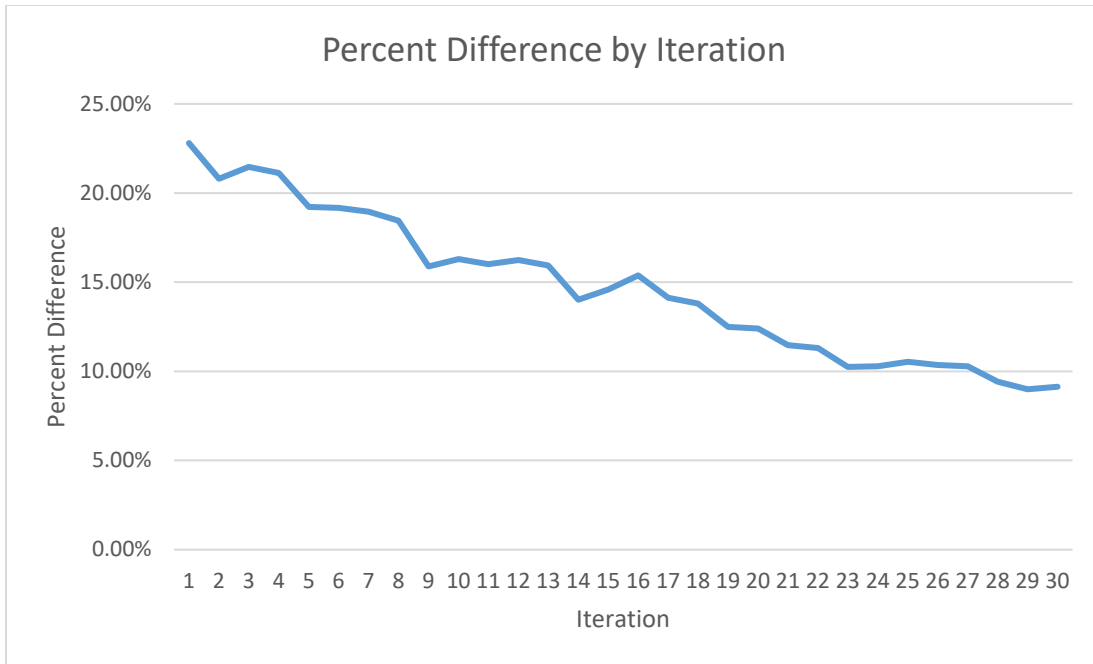


Figure 5-4 SPSA Convergence Over Iterations

After 30 iterations, the calibrated SPSA parameters θ_0 and θ_1 and final calibrated search cost model coefficients β are shown in Table 5-2 and Table 5-3, respectively. Using the calibrated parameter values, the perceived search cost can be empirically calculated for each traveler in the system.

Table 5-2 Calibration Results

Parameters	Search Cost in Mode	Search Cost in Destination	Search Cost in TOD	Search Cost in Route
θ_0	1.643	0.802	1.342	0.874
θ_1	0.964	2.509	0.792	1.497

Table 5-3 Calibrated Search Cost Model Parameters

Variables:	Search Cost in Mode	Search Cost in Destination	Search Cost in TOD	Search Cost in Route
Constant	2.203	1.076	0.540	0.336
Generalized cost	0.022	0.058	0.006	0.001
Gender (female)	0.013	0.035	0.128	0.147
Purpose (work/school)	-0.097	-0.253	-0.072	0.147
Income (<\$50k)	0.181	0.472	-0.215	-0.448
Income (\$50k - \$100k)	0.082	0.213	-0.226	-0.310
Distance	-0.019	-0.050	-0.006	-0.009
Peak-hour travel	0.155	0.404	0.089	0.015
Number of vehicles	-0.085	-0.221	0.236	-0.052

As indicated in Section 3.3, the calibration problem in this study is quite complex considering the multi-dimensional nature associated with the search cost model. Changes in one travel dimension could lead to changes in the final model results. This proposes a calibration algorithm and proves that the algorithm is able to reduce the overall model result errors, which would be considered a success in the calibration process.

5.4.2 Validation results

Even though the objective function of the calibration algorithm is to minimize the difference between observed traffic counts and simulated link volumes, this section validates results in other behavior dimensions (i.e., mode choice, destination choice, and TOD) to have a comprehensive assessment of the model performance. As illustrated in Section 3.3, 2007/2008 HTS data are used to validate model results. The

survey data are expanded based on the survey weights in the modeling area to be comparable with model results. All the model results and observations are reported at the trip level.

Figure 5-5 presents the aggregate share of the observed and estimated trip travel mode choice. Results suggest that the proposed integrated model could replicate the observed travel mode share fairly good. However, the proposed model overestimates the single-occupancy-vehicle (SOV) share by over 3% and slightly underestimates other modes. It is important to note is that the travel mode school bus, which is explicitly modeled in InSITE, is considered part of the transit mode.

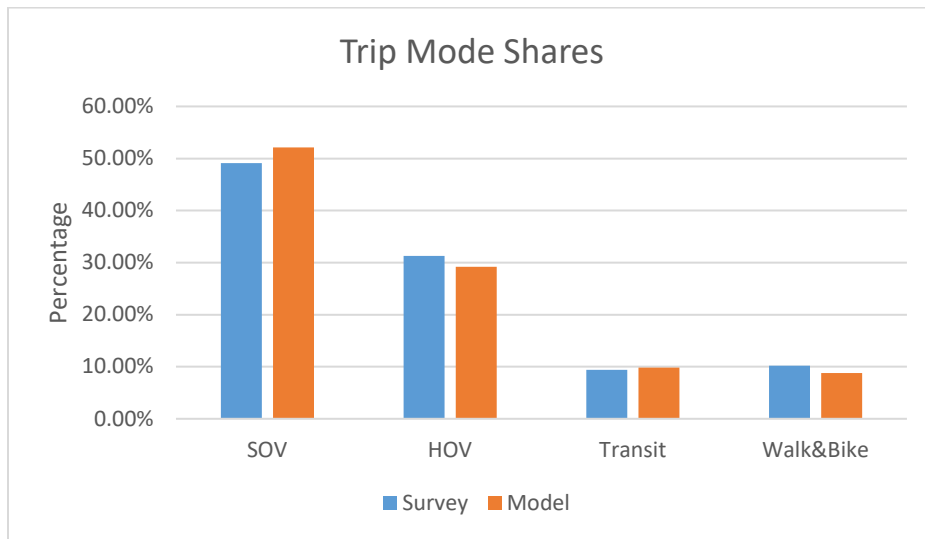


Figure 5-5 Observed and Simulated Trip Mode Shares

Figure 5-6 demonstrates the observed and simulated trip length distribution at a five-mile interval. In summary, the integrated model tends to overestimate the trip length as

higher percentages are seen in both 5-9.99 and 10-14.99 bins. Despite the errors, the integrated model does reflect the overall trip length distribution as observed in HTS.

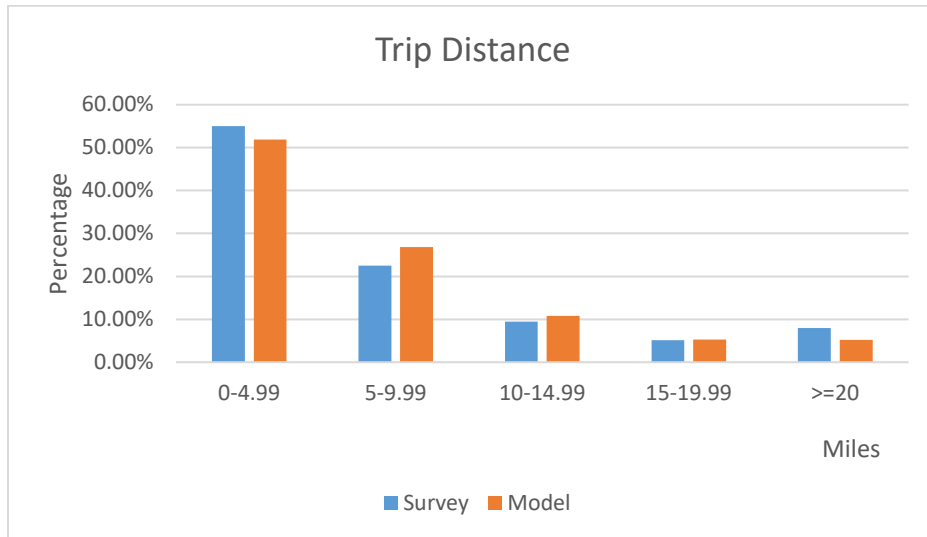


Figure 5-6 Observed and Simulated Trip Distance Distributions

Figure 5-7 reports the observed and simulated trip time-of-day distribution. The results suggest that the integrated model overestimates trip departures in both morning and afternoon peaks. In general, the integrated model tends to underestimate the congestion level in the network based on the validation results in Figure 5-5, Figure 5-6, and Figure 5-7. Consequently, travelers in the proposed model are likely to drive alone, take longer paths, and depart during rush hours as a result of the underestimated congested travel time.

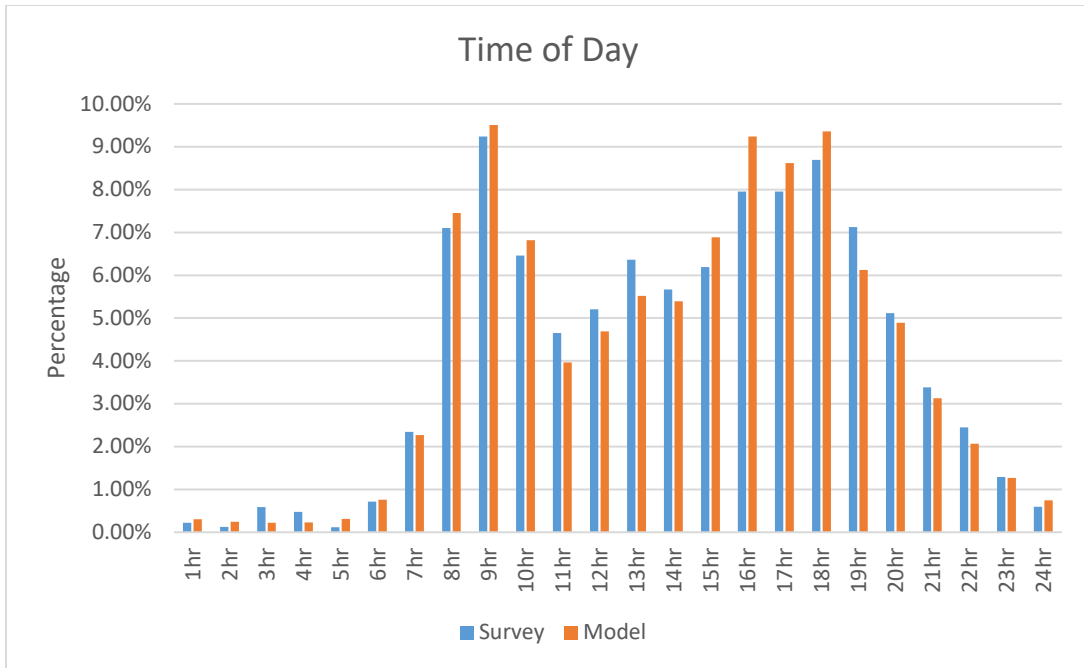


Figure 5-7 Observed and Simulated TOD Distributions

In addition to survey data, this study also validates assignment results with traffic counts. The primary focus of the assignment validation is the ability of the integrated model assignment to reproduce observed daily traffic volumes. This validation can be considered a model system validation, since it will be impacted by the travel models embodied in both InSITE and DTALite.

Table 5-4 Validation by Function Class

Function Class	Percent Difference
Interstate	4.75%
Freeway	-6.24%
Primary Arterial	-7.78%
Minor Arterial	9.06%
Collector & Others	16.65%

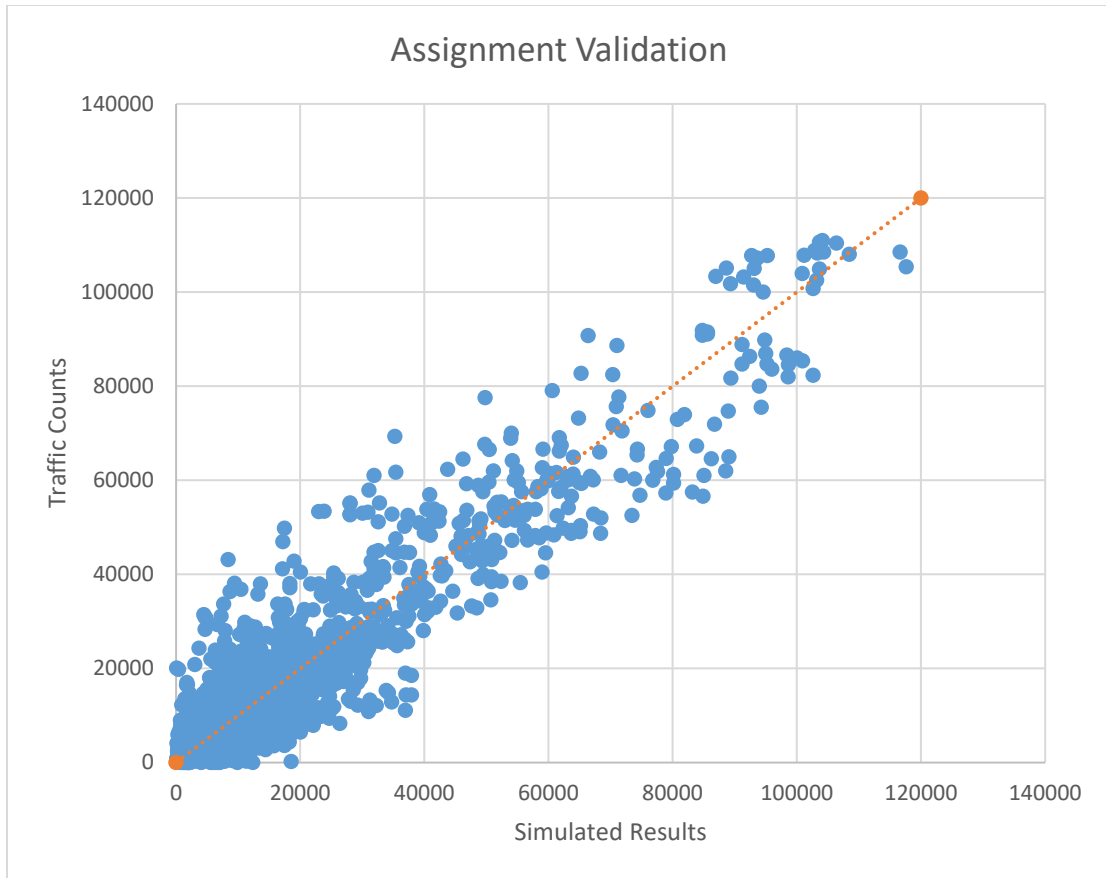


Figure 5-8 Assignment Results Compared with Observations

Table 5-4 reports the validation results by function classes. The percent difference for the majority of function classes is within 10%. It is worth noting that the model tends to overestimate the volume on lower facility types such as minor arterials and collectors. This is probably because these facilities are underrepresented in the network, meaning that not all local roads are included in the network. Consequently, more traffic is diverted to minor arterials and collectors. However, assignment results at various function classes are within the accepted accuracy range in general. Figure 5-8 plots the simulated assignment results against traffic counts. The dashed line represents the 45-degree line where simulated results match the observed link volumes exactly. The

overall assignment performance could replicate the observed pattern, with some outliers. To further analyze the validation results, Figure 5-9 illustrates the geographic distribution of the over- and underestimated links. Blue links represent links that are underestimated while red links represent overestimated ones. Overall, the model tends to underestimate traffic volumes on the Baltimore beltway but overestimate on I-70 and northern I-95 towards Baltimore.

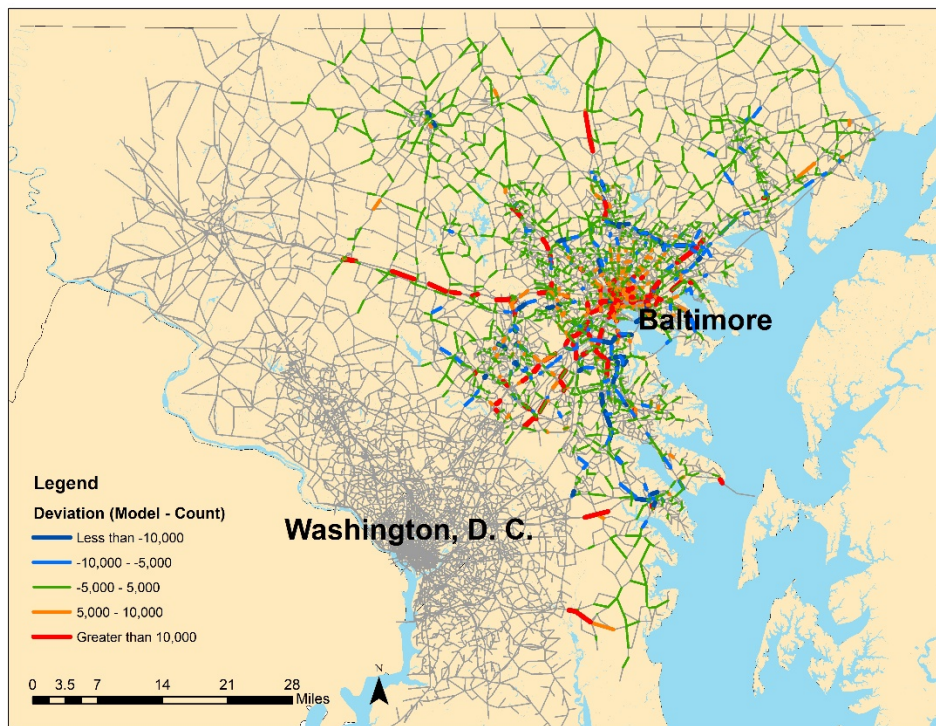


Figure 5-9 Locations of Over- and Underestimated Links

5.5 Application Results

With the help of longitudinal information, rich behavior dynamics can be observed in the BUE-based integrated model. The unique search process embedded in the proposed model acts like a comprehensive database where each behavioral decision during the search process is recorded for every agent in the system. Figure 5-9 demonstrates how the search gain and search cost ratio evolves during the search process for a specific agent. Initially, all travel dimensions are rewarding for this agent (i.e., the gain/cost ratio is larger than 1). This agent decides to search for alternatives in the most rewarding dimension, which is the departure time choice. At iteration 2, this agent determines to search for destination alternatives since destination choice is the dimension with the highest gain/cost ratio. The search continues as the process iterates. Finally, this agent stops searching at iteration 7 when the gain/cost ratios in all dimensions fall below 1, meaning that another round of searching is no longer rewarding and this agent is satisfied with the current travel plan. However, if external conditions alter (e.g., construction work in the neighborhood), this agent might change his/her aspiration level and start the search process again. Figure 5-9 only describes the search dynamics for one particular travel. Eventually, when all travelers in the system stop searching for new alternatives, the BUE condition has been reached.

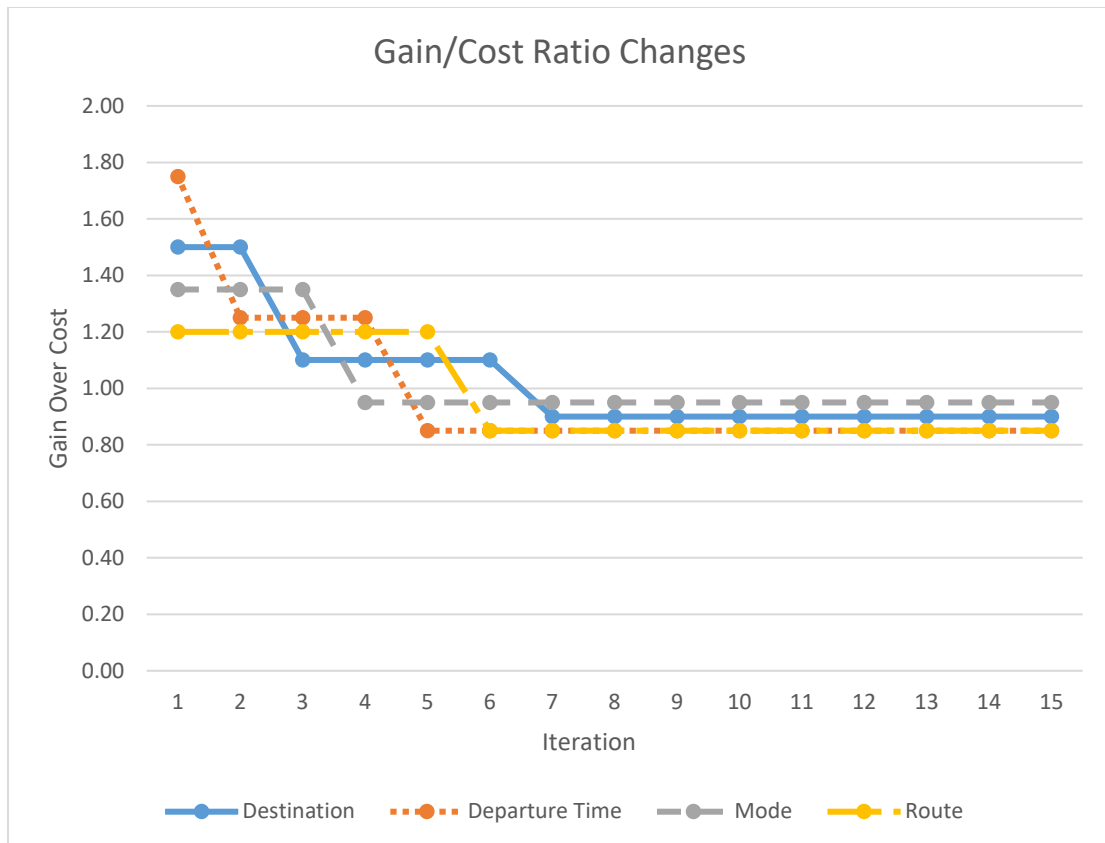


Figure 5-10 Changes of Gain/Cost Ratio over Iterations

Figure 5-10 illustrates the convergence of multi-dimensional behaviors where the percentage of the population searching for alternatives in four dimensions are reported. The preliminary results suggest that people tend to search for alternative routes at the beginning. Later, departure time options are explored. The low percentage of people search for destination or mode alternatives. The result indicates that the BUE is reached after around 11 iterations when no one in the system is searching for new alternatives. This is also in line with the travel behavior observed in reality. For instance, it would take a person a few days (no more than ten days) to find the satisfied travel pattern if this person is new to a place or turbulence occurs in the external transportation system.

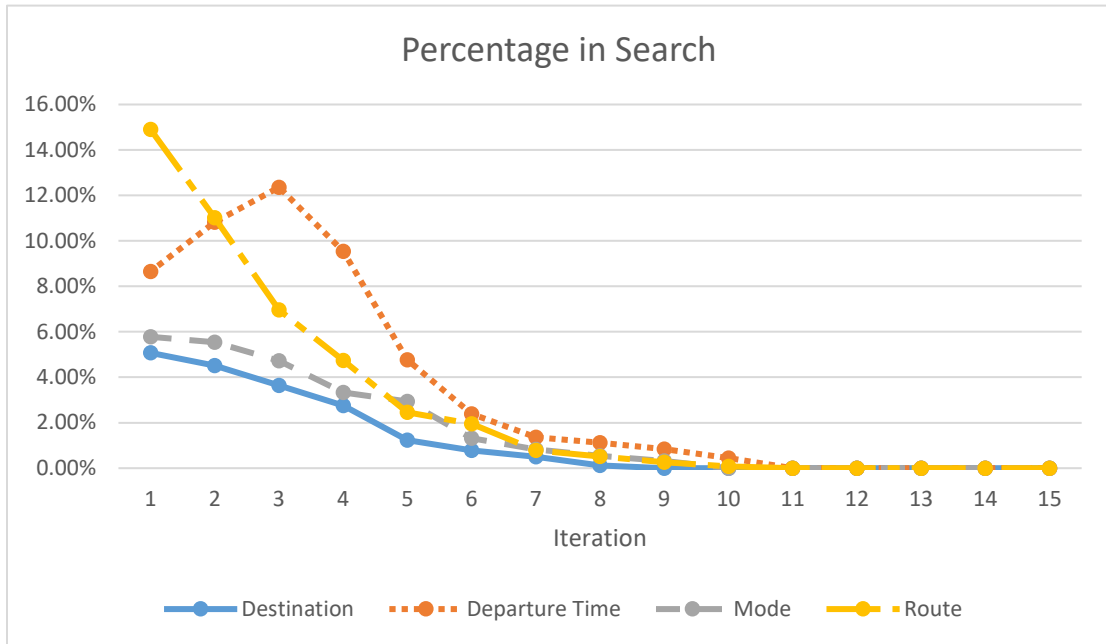


Figure 5-11 Convergence of the Integrated Model

In addition to the model convergence, more behavior dynamics in various travel dimensions can be observed from the proposed model. Figure 5-11 illustrates the mode share changes at the trip level in the search process. At the first iteration, individuals tend to drive since free flow travel time was used in the system. As the search process iterates, individuals tend to switch to other mode alternatives when the traffic congestion is taken into consideration.

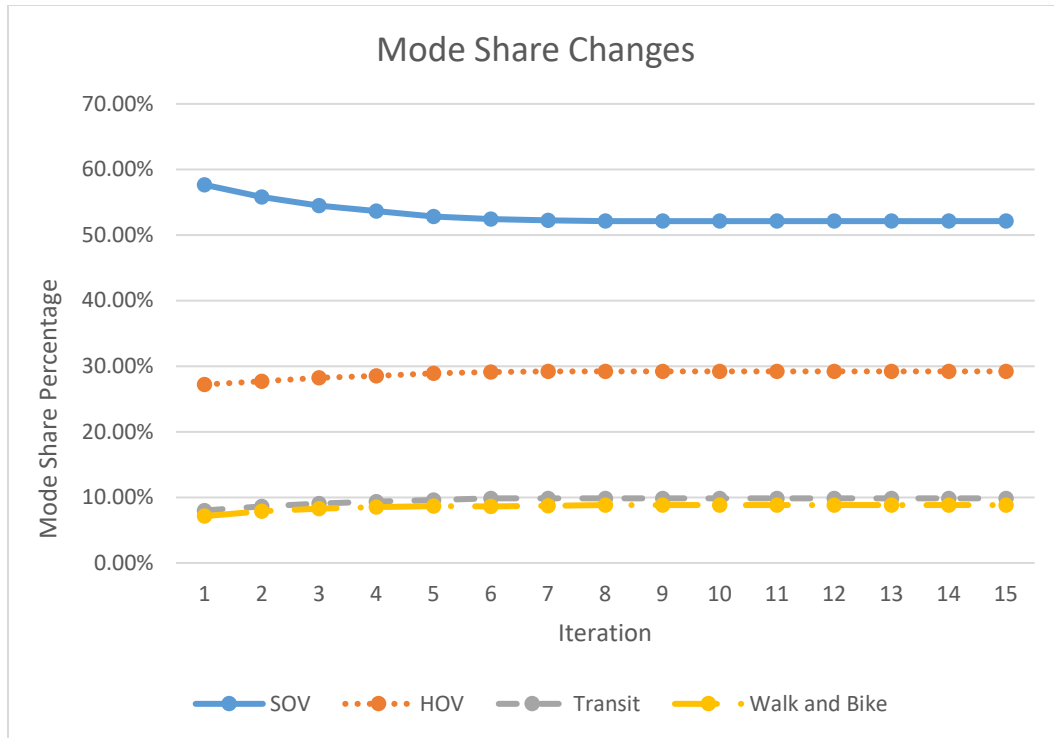


Figure 5-12 Mode Share over Iteration

5.6 Practical Implementation of Fast Convergence and Runtime

One major motivation of this dissertation is the excessively long runtime associated with the integrated ABM and DTA model. With the implementation of BUE theory, plus the speed boost in both InSITE and DTALite, the integrated model runtime has reduced significantly. Currently, a complete 15-iteration BUE-based integrated model takes 24 hours. A detailed runtime breakdown for a 15-iteration model run is shown in Table 5-5.

Table 5-5 A 15-Iteration Model Runtime Breakdown

	Runtime (hrs)
InSITE ABM	9
DTALite	11
Search process	1
CUBE skim conversion	3
Total	24

Previously, a sequentially integrated InSITE and DTALite model would take 30 hours per iteration. Currently, a BUE-based InSITE and DTALite integrated model only consumes 1.6 hours per iteration on average. Even though the BUE-based integrated model would require more iterations to converge, the overall runtime has decreased greatly. A detailed runtime comparison between a sequentially integrated model and BUE-based integrated model is presented in Table 5-6. With accelerating InSITE ABM and DTALite, and by implementing the BUE approach, the overall model runtime has been improved by almost 19 times.

Table 5-6 Runtime Comparison

	Sequential ABM-DTA Integration (hrs)	BUE-Based Integration (hrs)
1-Iteration ABM	16	0.6
1-Iteration DTA	2.5	0.75
1-Iteration Integrated Model	30	1.6
A Complete Run with 15 Feedback Loops	450	24

Bringing down the integrated model runtime to a time frame of one day has two major practical values. First, it allows for many more practitioners to actually start using this

modeling tool for application purposes without having to wait days or even weeks to see the model results. The sequentially integrated InSITE and DTALite model developed for the BMC area in the SHRP2 program takes more than a week to run; it is difficult for BMC to use this model because of the runtime. Second, only with a runtime of one day or so could the proposed calibration approach be possible. As described in Section 3.2, the calibration algorithm takes 30 iterations. In other words, the calibration process with the sequential integration would take around 562 days, which makes it impossible to implement. In general, the runtime improvement itself is a major contribution of this dissertation as it allows agencies to actually implement the integrated AMB and DTA model within a reasonable time frame.

5.7 Summary

This chapter applies the proposed BUE-based ABM and DTA integrated framework to the Washington-Baltimore metropolitan area network. InSITE ABM and DTALite models are integrated based on the BUE theory. Brief introductions on InSITE and DTALite models are given. A Python wrapper is developed to execute various components in the framework and enables efficient data exchange. To enhance the BUE theory to a tour-based environment, several rules are proposed in this chapter. To further accelerate the major model components, InSITE ABM and DTALite, the advanced computing technology (multi-processing) and fast time-dependent skim generation program are implemented. Multi-processed InSITE ABM now runs four

times faster than the original version. The new time-dependent skim generation method saves about 26 hours for the overall integrated model runtime.

In addition to the significant runtime savings, various model results are also provided. This chapter first reports the calibration and validation results. SPSA coefficients and calibrated search cost model parameters are reported. Model results in various behavior dimensions after calibration are validated against HTS data. The validation results suggest that the integrated model could replicate observed travel patterns but tend to underestimate the congestion level in the network. The overall SPSA algorithm performance indicates that the calibration process can reduce the percent difference between simulated link volumes and traffic counts to below 10%. Specific traffic assignment results and validation by function classes are also reported. Considering the complexity associated with the calibration approach, the performance of the calibration algorithm is considering satisfactory.

One major advantage of the proposed integrated model is to record the multi-dimensional travel dynamics. Rich behavior dynamics at both the individual level and the system level are presented in this chapter. At the individual level, the search gain and cost ratio over model iterations is shown, which could reveal the travel decision dynamics for a specific traveler with regards to when this person starts/stops searching for alternatives and what travel dimensions this traveler searches at each iteration. At the system level, the model convergence and mode share changes over iterations are also exhibited. Information regarding what travel dimensions people tend to search and

when the system reaches BUE can be easily identified. With the implementation of the BUE theory, the analysis capability of the integrated model has been enhanced.

Chapter 6 Conclusions

6.1 Research Summary

Advances in information technology and modeling methodology drives innovations in travel demand modeling. More and more new travel demand models have been developed in recent decades. One major motivation of this dissertation is a practical challenge in the travel demand modeling field: how to properly integrate ABM and DTA models. Specific challenges include an extremely long model runtime and slow model convergence. To resolve this modeling challenge, this dissertation proposes to employ an innovative behavioral theory, BUE, to link ABM and DTA models.

The BUE-based integration approach proposed in this dissertation contributes to the literature mainly in two ways. Firstly, the BUE theory provides an alternative way to look at the decision-making process when modeling travel demand. With imperfect information and satisficing behavior, BUE theory employs an agent-based modeling approach that emphasizes the role of learning and searching behavior involved in the travel decision-making process. Travelers in the system no longer follow a predefined decision-making order (e.g., activity generation, destination choice, TOD choice, mode choice, and route choice). The multi-dimensional search mechanism in BUE theory better describes the complex process of making travel decisions. Furthermore, the agent-based modeling approach enables the capture of more behavior dynamics. As

demonstrated in Section 5.5, the behavior dynamics of an individual or the whole system can be recorded, which increases the analysis capability of the proposed integrated model.

Second, the BUE-based integration approach reduces the model runtime significantly as a result of a new convergence definition. Previous integration methods measure the model convergence from a network-wide point of view. DTA models must run iteratively to reach the user equilibrium condition first. The integrated model then needs to run iteratively to ensure that congestion is properly reflected. The BUE-based integration approach, however, measures the model convergence from an individual point of view, thanks to the agent-based modeling method. The model reaches behavioral user equilibrium when all agents are satisfied with their travel options and no longer search for travel alternatives. In this way, the ABM model only needs to execute for a proportion of the agents who search for travel alternatives and the DTA model only runs once for each integrated model iteration. Consequently, the model runtime is decreased substantially.

To implement the BUE theory in integrating ABM and DTA models, a Python program is developed to mimic the search process in the decision-making procedure. The Python program calculates the search gain and search cost associated with the search process for each traveler in the system and decides whether to search for new travel alternatives. The Python program acts like a program wrapper that also calls components in ABM and DTA models. Intermediate steps are also developed to convert data formats

required by different model components and to ensure efficient data exchange processes.

The BUE-based integration framework has been applied to the Washington-Baltimore metropolitan area. InSITE ABM is integrated with DTALite in this region based on the BUE theory. To further accelerate the integrated model, advanced computing technologies are developed to boost both InSITE and DTALite. The integrated InSITE and DTALite model only takes about 24 hours to run compared to weeks in a typical sequential integrated model. In addition to runtime savings, model calibration and validation results suggest acceptable assignment results. Various model results illustrating behavior dynamics are also presented in the dissertation to further demonstrate expedited analysis capability.

Lastly, this dissertation demonstrates how passively collected data can provide valuable information for developing cutting-edge travel demand models. To keep pace with the big data era, new travel demand models must take advantage of this big data source to supplement or even replace traditional travel surveys. Passively collected data are easier to obtain and typically come with large volumes. Researchers need to develop advanced data mining and machine learning models to extract useful facts from this data source. This dissertation illustrates how mode information can be inferred from smartphone GPS data. Even though more information must be detected from passively collected data, such as trip purposes and socio-demographic information, this

dissertation sheds light on a promising data source for advanced travel demand modeling purposes.

6.2 Discussions and Future Research

As travel demand models evolve, impractical behavior assumptions must be relaxed and a travel decision-making process that is closer to a real-world situation must be developed. The BUE principle attempts to provide an alternative theory to explain people's travel behavior other than the normative theory. Admittedly however, limitations exist when applying the BUE principle to integrate ABM and DTA models. One limitation is that major components in typical ABMs, such as discrete choice models, are based on the normative theory. The normative theory emphasizes the assumption of perfect information and complete rationality, which contradicts the positive theory embedded in the BUE principle. One can argue that the positive theory only describes the overall decision-making process while the normative theory explains the behavior in each specific travel dimension. This is also similar to the situation in hybrid approach-based ABMs, which attempt to combine econometric-based models with agent-based models. The underlying behavior assumptions are not consistent in either of those ABMs. Furthermore, econometric-based models have been widely applied in the field of travel demand modeling and established high credibility. It might take a leap of faith to replace the entire econometric-based models with positive theory-based models.

In this dissertation, only travel mode information is detected from passively collected data. Future research should explore more valuable trip-related information from passively collected data. Researchers have been working on imputing missing information such as trip purposes and users' socio-demographic data (Tang, Pan, & Zhang, 2018; Zhu, Gonder, & Lin, 2017). Passively collected data could completely replace the traditional travel-diary type of surveys if trip purpose and socio-demographic information are accurately predicted. With the wide coverage of passively collected data, this data source could play a much more critical role in advanced travel demand analysis in the future.

Recognizing the value of passively collected data, the Federal Highway Administration (FHWA) has supported a series of Exploratory Advanced Research programs to look at how data collected from smartphones and other smart devices could supplement or even replace traditional travel surveys. The low cost associated with the data collection and easy access to reaching a large group of the population are advantages that traditional survey approaches cannot provide. Especially for methodologies like BUE theory that emphasizes modeling people's behavior dynamics, panel data or repeated observations are preferred. Traditional surveys focus only on one cross section while passively collected data can easily reveal repeated observations for an object over a long period. Rich information regarding the travel behavior adjustment process can be detected from this data sources. The model calibration approach proposed in this study is only a short-term solution. For a long-term point of view, longitudinal behavior data

are still needed to estimate the search cost model parameters. Passively collected data has the potential to provide this type of information without expensive behavioral surveys.

Appendices

Appendix A: Integration Implementation Code

```
""""
BUE-based Integrated InSITE and DTALite

@author: dyang114
""""

import pandas as pd
import numpy as np
import os
import subprocess
from simpledbf import Dbf5
import shutil

itern_num = 15

dir_integrated = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/"
dir_insite1 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model Run
1/dist/bin/"
dir_insite2 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model Run
2+/dist/bin/"
dir_insite1_1 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model
Run 1_1/dist/bin/"
dir_insite1_2 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model
Run 1_2/dist/bin/"
dir_insite1_3 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model
Run 1_3/dist/bin/"
dir_insite1_4 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model
Run 1_4/dist/bin/"
dir_insite1_5 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model
Run 1_5/dist/bin/"
dir_data1 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model Run
1/dist/data/"
dir_data2 = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/InSITE Model Run
2+/dist/data/"
dir_R = "C:/Users/carrion/Documents/R/R-3.3.2/bin/Rscript.exe"
dir_Cube = "C:/Program Files (x86)/Citilabs/CubeVoyager/runtp.exe"
```

```

dir_BUE = "G:/MITAMS InSITE ABM-DTALite Clean Set Up/Di_BUE/"

def RunInSITE(start,end):
    print "Running Customized InSITE from %d to %d" %(start, end)
    os.chdir(dir_insite2)
    subprocess.call([dir_insite2+"ComponentLauncher.exe", "%d" %start, "%d"
%end])

def RunDTALite():
    print "Running DTALite"
    os.chdir(dir_data2)
    subprocess.call(dir_data2+"DTALite.exe")

def RunCUBE():
    print "Running CUBE Script"
    cmd = [dir_Cube,dir_data2+"ConvertToCubeSkims.s"]
    subprocess.call(cmd)

def RunR(module):
    print "Running R Script " + module
    cmd = [dir_R, "--vanilla", dir_BUE+"agent_filter_"+module+".R"]
    subprocess.call(cmd)

"""=====
#Run InSITE with free flow condition
print 'Running InSITE with free flow speed (1st iteration)'

cwd = os.path.join(os.path.dirname(__file__))
scenario_dir = os.path.abspath(os.path.dirname(cwd))
multithread_dir = "%s\\Cube\\"%scenario_dir

#Multiprocessing of InSITE
dirlist = [dir_insite1_1, dir_insite1_2, dir_insite1_3, dir_insite1_4, dir_insite1_5]
processes = [subprocess.Popen([multithread_dir +
dirlistelement],cwd=multithread_dir) for dirlistelement in dirlist]
exitcodes = [p.wait() for p in processes]

#Run DTALite
RunDTALite()

#Convert DATLite skims to CUBE skims
RunCUBE()

#Test model convergence

```

RunR('ConvergenceScript')

```
"""=====
#Analysis DTA outputs and run BUE analysis (2+ iteration)
print 'Initialization: seize population information and calculate search cost'
os.chdir(dir_data2)
shutil.copyfile('persons.dbf','persons_temp.dbf')
person_info_dbf = Dbf5('persons_temp.dbf')
person_info = person_info_dbf.to_dataframe()

search_outcome_d = pd.read_csv('search_outcome_d.csv')
search_outcome_m = pd.read_csv('search_outcome_m.csv')
search_outcome_t = pd.read_csv('search_outcome_t.csv')
search_outcome_r = pd.read_csv('search_outcome_r.csv')
search_outcome_d = search_outcome_d.set_index('person_tour_stop')
search_outcome_m = search_outcome_m.set_index('person_tour_stop')
search_outcome_t = search_outcome_t.set_index('person_tour_stop')
search_outcome_r = search_outcome_r.set_index('person_tour_stop')
search_count_d = pd.read_csv('search_count_d.csv')
search_count_m = pd.read_csv('search_count_m.csv')
search_count_t = pd.read_csv('search_count_t.csv')
search_count_r = pd.read_csv('search_count_r.csv')
search_count_d = search_count_d.set_index('person_tour_stop')
search_count_m = search_count_m.set_index('person_tour_stop')
search_count_t = search_count_t.set_index('person_tour_stop')
search_count_r = search_count_r.set_index('person_tour_stop')
search_dimension = pd.read_csv('output_test_dimension.csv')
search_dimension['id'] = search_dimension['person_tour_stop']
search_dimension = search_dimension.set_index('id')

for iterate in range(2,iter_n_num+1):
    #print iterate
    shutil.copyfile('TripModes_modified.dbf','TripModes_modified_temp.dbf')
    input_agent = pd.read_csv('input_agent_test.csv')
    output_agent = pd.read_csv('output_agent.csv')
    trip_mode_dbf = Dbf5('TripModes_modified_temp.dbf')
    trip_mode = trip_mode_dbf.to_dataframe()
    trip_mode['person_tour_stop'] = trip_mode['personId'].map(str) + '_' +
trip_mode['tourId'].map(str) + '_' + trip_mode['halfTour'].map(str) + '_' +
trip_mode['stopId'].map(str)
    input_agent['person_tour_stop'] = input_agent['person_id'].map(str) + '_' +
input_agent['tour_id'].map(str) + '_' + input_agent['half_tour_id'].map(str) + '_' +
input_agent['stop_id'].map(str)
    trip_mode['person_tourId'] = trip_mode['personId'].map(str) + '_' +
trip_mode['tourId'].map(str)
```

```

agent_join =
trip_mode.merge(input_agent[['person_tour_stop','agent_id']],how='left',left_on='pers
on_tour_stop',right_on='person_tour_stop')
agent_join = agent_join.merge(output_agent[['agent_id',' distance','
travel_time_in_min']],how='left',left_on='agent_id',right_on='agent_id')

#Fill Nan with travel time from TripModes
agent_join[' travel_time_in_min'] = agent_join['
travel_time_in_min'].fillna(agent_join['travTime'])
agent_join[' distance'] = agent_join[' distance'].fillna(agent_join['distance'])
agent_combine =
agent_join.merge(person_info[['PERSONID','AGE','GENDER','HHINC5S']],how='le
ft',left_on='personId',right_on='PERSONID')
agent_combine = agent_combine.sort_values('personId')
agent_combine['id'] = agent_combine['person_tour_stop']
agent_combine = agent_combine.set_index('id')

#convert purpose & income to dummy
agent_combine['purpose_dummy'] = np.where(agent_combine['tourId']<3,1,0)
depart_time = agent_combine['dep_time']
agent_combine['peak'] = np.where((depart_time >= 420 and num <= 540) or
(num >= 1020 and num <= 1140),1,0)
agent_combine = pd.get_dummies(data=agent_combine, columns=['HHINC5S'])

agent_combine['cost_dest'] = 1.076 + 0.058 * agent_combine['travTime'] + 0.035 *
agent_combine['GENDER'] + 0.472 * agent_combine['HHINC5S_0.0'] + 0.213 *
agent_combine['HHINC5S_1.0'] -0.050 * agent_combine[' distance'] -0.253 *
agent_combine['purpose_dummy'] + 0.404 * agent_combine['peak'] + 0.296 *
agent_combine['cars']
agent_combine['cost_TOD'] = 0.54 + 0.006 * agent_combine['travTime'] + 0.128 *
agent_combine['GENDER'] -0.215 * agent_combine['HHINC5S_0.0'] - 0.226 *
agent_combine['HHINC5S_1.0'] -0.006 * agent_combine[' distance'] -0.072 *
agent_combine['purpose_dummy'] + 0.089 * agent_combine['peak'] + 0.154 *
agent_combine['cars']
agent_combine['cost_mode'] = 2.203 + 0.022 * agent_combine['travTime'] + 0.013
* agent_combine['GENDER'] + 0.181 * agent_combine['HHINC5S_0.0'] + 0.082 *
agent_combine['HHINC5S_1.0'] -0.019 * agent_combine[' distance'] -0.097 *
agent_combine['purpose_dummy'] + 0.155 * agent_combine['peak'] + 0.114 *
agent_combine['cars']
agent_combine['cost_route'] = 0.336 + 0.001 * agent_combine['travTime'] + 0.147
* agent_combine['GENDER'] -0.448 * agent_combine['HHINC5S_0.0'] -0.310 *
agent_combine['HHINC5S_1.0'] -0.009 * agent_combine[' distance'] +0.147 *
agent_combine['purpose_dummy'] + 0.015 * agent_combine['peak'] + 0.172 *
agent_combine['cars']

```

```

search_outcome_d = search_outcome_d.merge(agent_combine[['
travel_time_in_min']],how='left',left_index=True,right_index=True)
search_outcome_m = search_outcome_m.merge(agent_combine[['
travel_time_in_min']],how='left',left_index=True,right_index=True)
search_outcome_t = search_outcome_t.merge(agent_combine[['
travel_time_in_min']],how='left',left_index=True,right_index=True)
search_outcome_r = search_outcome_r.merge(agent_combine[['
travel_time_in_min']],how='left',left_index=True,right_index=True)
search_outcome_d = search_outcome_d.rename(columns={'
travel_time_in_min':'DTA_time_%d'%iterate})
search_outcome_m = search_outcome_d.rename(columns={'
travel_time_in_min':'DTA_time_%d'%iterate})
search_outcome_t = search_outcome_d.rename(columns={'
travel_time_in_min':'DTA_time_%d'%iterate})
search_outcome_r = search_outcome_d.rename(columns={'
travel_time_in_min':'DTA_time_%d'%iterate})

agent_combine['gain_dest_%d'%iterate] = None
agent_combine['gain_TOD_%d'%iterate] = None
agent_combine['gain_mode_%d'%iterate] = None
agent_combine['gain_route_%d'%iterate] = None
search_dimension['dimension_%d'%iterate] = 0

for index, row in agent_combine.iterrows():
    agent_combine.loc[index,'gain_dest_%d'%iterate] = (-
row['travTime']*gain_percent +
search_outcome_d.loc[row['person_tour_stop']].max()/(search_count_d.loc[row['per
son_tour_stop'],'count'] +1)
    agent_combine.loc[index,'gain_TOD_%d'%iterate] = (-
row['travTime']*gain_percent +
search_outcome_t.loc[row['person_tour_stop']].max()/(search_count_t.loc[row['pers
on_tour_stop'],'count'] +1)
    agent_combine.loc[index,'gain_mode_%d'%iterate] = (-
row['travTime']*gain_percent +
search_outcome_m.loc[row['person_tour_stop']].max()/(search_count_m.loc[row['pe
rson_tour_stop'],'count'] +1)
    agent_combine.loc[index,'gain_route_%d'%iterate] = (-
row['travTime']*gain_percent +
search_outcome_r.loc[row['person_tour_stop']].max()/(search_count_r.loc[row['pers
on_tour_stop'],'count'] +1)

    if (
        agent_combine.loc[index,'gain_dest_%d'%iterate] >
agent_combine.loc[index,'cost_dest'] and

```

```

        agent_combine.loc[index,'gain_dest_%d'%iterate] -
agent_combine.loc[index,'cost_dest'] >
agent_combine.loc[index,'gain_TOD_%d'%iterate] -
agent_combine.loc[index,'cost_TOD'] and
        agent_combine.loc[index,'gain_dest_%d'%iterate] -
agent_combine.loc[index,'cost_dest'] >
agent_combine.loc[index,'gain_mode_%d'%iterate] -
agent_combine.loc[index,'cost_mode'] and
        agent_combine.loc[index,'gain_dest_%d'%iterate] -
agent_combine.loc[index,'cost_dest'] >
agent_combine.loc[index,'gain_route_%d'%iterate] -
agent_combine.loc[index,'cost_route']
    ):
        search_count_d.loc[row['person_tour_stop'],'count'] += 1
        search_dimension.loc[row['person_tour_stop'],'dimension_%d'%iterate] = 1

    elif (
        agent_combine.loc[index,'gain_TOD_%d'%iterate] >
agent_combine.loc[index,'cost_TOD'] and
        agent_combine.loc[index,'gain_TOD_%d'%iterate] -
agent_combine.loc[index,'cost_TOD'] >
agent_combine.loc[index,'gain_dest_%d'%iterate] -
agent_combine.loc[index,'cost_dest'] and
        agent_combine.loc[index,'gain_TOD_%d'%iterate] -
agent_combine.loc[index,'cost_TOD'] >
agent_combine.loc[index,'gain_mode_%d'%iterate] -
agent_combine.loc[index,'cost_mode'] and
        agent_combine.loc[index,'gain_TOD_%d'%iterate] -
agent_combine.loc[index,'cost_TOD'] >
agent_combine.loc[index,'gain_route_%d'%iterate] -
agent_combine.loc[index,'cost_route']
    ):
        search_count_t.loc[row['person_tour_stop'],'count'] += 1
        search_dimension.loc[row['person_tour_stop'],'dimension_%d'%iterate] = 2

    elif (
        agent_combine.loc[index,'gain_mode_%d'%iterate] >
agent_combine.loc[index,'cost_mode'] and
        agent_combine.loc[index,'gain_mode_%d'%iterate] -
agent_combine.loc[index,'cost_mode'] >
agent_combine.loc[index,'gain_TOD_%d'%iterate] -
agent_combine.loc[index,'cost_TOD'] and
        agent_combine.loc[index,'gain_mode_%d'%iterate] -
agent_combine.loc[index,'cost_mode'] >

```

```

agent_combine.loc[index,'gain_dest_%d'%iterate] -
agent_combine.loc[index,'cost_dest'] and
    agent_combine.loc[index,'gain_mode_%d'%iterate] -
agent_combine.loc[index,'cost_mode'] >
agent_combine.loc[index,'gain_route_%d'%iterate] -
agent_combine.loc[index,'cost_route']
):
    search_count_m.loc[row['person_tour_stop'],'count'] += 1
    search_dimension.loc[row['person_tour_stop'],'dimension_%d'%iterate] = 3

elif (
    agent_combine.loc[index,'gain_route_%d'%iterate] >
agent_combine.loc[index,'cost_route'] and
    agent_combine.loc[index,'gain_route_%d'%iterate] -
agent_combine.loc[index,'cost_route'] >
agent_combine.loc[index,'gain_TOD_%d'%iterate] -
agent_combine.loc[index,'cost_TOD'] and
    agent_combine.loc[index,'gain_route_%d'%iterate] -
agent_combine.loc[index,'cost_route'] >
agent_combine.loc[index,'gain_dest_%d'%iterate] -
agent_combine.loc[index,'cost_dest'] and
    agent_combine.loc[index,'gain_route_%d'%iterate] -
agent_combine.loc[index,'cost_route'] >
agent_combine.loc[index,'gain_mode_%d'%iterate] -
agent_combine.loc[index,'cost_mode']
):
    search_count_r.loc[row['person_tour_stop'],'count'] += 1
    search_dimension.loc[row['person_tour_stop'],'dimension_%d'%iterate] = 4

search_outcome_d['DTA_time_%d'%iterate]=None
search_outcome_m['DTA_time_%d'%iterate]=None
search_outcome_t['DTA_time_%d'%iterate]=None
search_outcome_r['DTA_time_%d'%iterate]=None

agent_combine['gain_dest_%d'%iterate] = None
agent_combine['gain_TOD_%d'%iterate] = None
agent_combine['gain_mode_%d'%iterate] = None
agent_combine['gain_route_%d'%iterate] = None

test_switch = search_dimension.copy()
test_dest = test_switch[test_switch['dimension_1']==1].copy()
test_TOD = test_switch[test_switch['dimension_1']==2].copy()
test_mode = test_switch[test_switch['dimension_1']==3].copy()
test_route = test_switch[(test_switch['dimension_1']==4) &
(test_switch['tripMode']<4)].copy()

```



```

    MTOD_agent = test_TOD[(test_TOD['tourId']==1) | (test_TOD['tourId']==2)]
    MTOD_agent = MTOD_agent[MTOD_agent['tourPurp'] ==
MTOD_agent['purpose']]
    MTOD_agent.to_csv('MTOD_agent.csv',index=False,header=True)
    FJD_agent = test_dest[(test_dest['tourId']==10) | (test_dest['tourId']==11)]
    FJD_agent = FJD_agent[FJD_agent['tourPurp'] == FJD_agent['purpose']]
    FJD_agent.to_csv('FJD_agent.csv',index=False,header=True)
    FJTOD_agent = test_TOD[(test_TOD['tourId']==10) |
(test_TOD['tourId']==11)]
    FJTOD_agent = FJTOD_agent[FJTOD_agent['tourPurp'] ==
FJTOD_agent['purpose']]
    FJTOD_agent.to_csv('FJTOD_agent.csv',index=False,header=True)
    INMD_agent = test_dest[(test_dest['tourId']==50) | (test_dest['tourId']==51) |
(test_dest['tourId']==52)]
    INMD_agent = INMD_agent[INMD_agent['tourPurp'] ==
INMD_agent['purpose']]
    INMD_agent.to_csv('INMD_agent.csv',index=False,header=True)
    INMTOD_Escort_agent = test_TOD[(test_TOD['tourId']==50) |
(test_TOD['tourId']==51) | (test_TOD['tourId']==52)]
    INMTOD_Escort_agent =
INMTOD_Escort_agent[INMTOD_Escort_agent['tourPurp'] == 1152]
    INMTOD_Escort_agent =
INMTOD_Escort_agent[INMTOD_Escort_agent['tourPurp'] ==
INMTOD_Escort_agent['purpose']]
    INMTOD_Escort_agent.to_csv('INMTOD_Escort_agent.csv',index=False,hea
der=True)
    INMTOD_agent = test_TOD[(test_TOD['tourId']==50) |
(test_TOD['tourId']==51) | (test_TOD['tourId']==52)]
    INMTOD_agent = INMTOD_agent[INMTOD_agent['tourPurp'] != 1152]
    INMTOD_agent = INMTOD_agent[INMTOD_agent['tourPurp'] ==
INMTOD_agent['purpose']]
    INMTOD_agent.to_csv('INMTOD_agent.csv',index=False,header=True)
    TMC_agent = test_mode[test_mode['tourPurp'] == test_mode['purpose']]
    TMC_agent = TMC_agent[(TMC_agent['tourId']!=90) &
(TMC_agent['tourId']!=91)]
    TMC_agent.to_csv('TMC_agent.csv',index=False,header=True)
    WBD_agent = test_dest[(test_dest['tourId']==90) | (test_dest['tourId']==91)]
    WBD_agent = WBD_agent[WBD_agent['tourPurp'] ==
WBD_agent['purpose']]
    WBD_agent.to_csv('WBD_agent.csv',index=False,header=True)
    WBTOD_agent = test_TOD[(test_TOD['tourId']==90) |
(test_TOD['tourId']==91)]
    WBTOD_agent = WBTOD_agent[WBTOD_agent['tourPurp'] ==
WBTOD_agent['purpose']]

```

```

    WBTOD_agent.to_csv('WBTOD_agent.csv',index=False,header=True)
    WBMC_agent = test_mode[test_mode['tourPurp'] == test_mode['purpose']]
    WBMC_agent = WBMC_agent[(WBMC_agent['tourId']==90) |
(WBMC_agent['tourId']==91)]
    WBMC_agent.to_csv('WBMC_agent.csv',index=False,header=True)
    STOPD_Home_agent = test_dest[test_dest['tourPurp'] != test_dest['purpose']]
    STOPD_Home_agent =
STOPD_Home_agent[STOPD_Home_agent['purpose']!=2048]
    STOPD_Home_agent =
STOPD_Home_agent[(STOPD_Home_agent['tourId']!=90) &
(STOPD_Home_agent['tourId']!=91)]
    STOPD_Home_agent.to_csv('STOPD_Home_agent.csv',index=False,header=
True)
    STOPD_Work_agent = test_dest[test_dest['tourPurp'] != test_dest['purpose']]
    STOPD_Work_agent =
STOPD_Work_agent[STOPD_Work_agent['purpose']!=2048]
    STOPD_Work_agent =
STOPD_Work_agent[(STOPD_Work_agent['tourId']==90) |
(STOPD_Work_agent['tourId']==91)]
    STOPD_Work_agent.to_csv('STOPD_Work_agent.csv',index=False,header=
True)
    STOPTOD_Home_agent = test_TOD[test_TOD['tourPurp'] !=
test_TOD['purpose']]
    STOPTOD_Home_agent =
STOPTOD_Home_agent[STOPTOD_Home_agent['purpose']!=2048]
    STOPTOD_Home_agent =
STOPTOD_Home_agent[(STOPTOD_Home_agent['tourId']!=90) &
(STOPTOD_Home_agent['tourId']!=91)]
    STOPTOD_Home_agent.to_csv('STOPTOD_agent.csv',index=False,header=
True)
    STOPTOD_Work_agent = test_TOD[test_TOD['tourPurp'] !=
test_TOD['purpose']]
    STOPTOD_Work_agent =
STOPTOD_Work_agent[STOPTOD_Work_agent['purpose']!=2048]
    STOPTOD_Work_agent =
STOPTOD_Work_agent[(STOPTOD_Work_agent['tourId']==90) |
(STOPTOD_Work_agent['tourId']==91)]
    STOPTOD_Work_agent.to_csv('STOPTOD_Work_agent.csv',index=False,he
ader=True)
    TRIPMC_agent = test_mode[test_mode['tourPurp'] != test_mode['purpose']]
    TRIPMC_agent = TRIPMC_agent[TRIPMC_agent['purpose']!=2048]
    TRIPMC_agent.to_csv('TRIPMC_agent.csv',index=False,header=True)
    test_route.to_csv('Route_agent.csv',index=False,header=True)

```

#Run R to update InSITE input files

```

RunR('MTOD')
RunR('FJD')
RunR('FJTOD')
RunR('INMD')
RunR('INMTOD')
RunR('TMC')
RunR('WBD')
RunR('WBTOD')
RunR('WBMC')
RunR('STOPD')
RunR('STOPTOD')
RunR('TRIPMC')

#Run InSITE to update agents' choices
RunInSITE(16,16)
RunInSITE(19,19)
RunInSITE(20,20)
RunInSITE(23,24)
RunInSITE(25,26)
RunInSITE(28,28)
RunInSITE(31,31)
RunInSITE(32,32)
RunInSITE(34,34)
RunInSITE(36,36)
RunInSITE(37,37)
RunInSITE(38,38)

#Update Trip Modes file to reflects changes
RunR('Trip_Modes_update')

#Generate new input_agent file for the next iteration
RunInSITE(39,39)

#Update agents with route changes
os.chdir(dir_data2)
input_agent_new = pd.read_csv('input_agent_test.csv')
input_agent_new = input_agent_new.drop('path_node_sequence', 1)

agent_joint =
input_agent_new.merge(output_agent[['agent_id','path_node_sequence']],how='left',left_on='agent_id',right_on='agent_id')
    agent_joint['person_tour_stop'] = agent_joint['person_id'].map(str) + '_' +
agent_joint['tour_id'].map(str) + '_' + agent_joint['half_tour_id'].map(str) + '_' +
agent_joint['stop_id'].map(str)

```

```
test_route['person_tour_stop'] = test_route['personId'].map(str) + '_' +
test_route['tourId'].map(str) + '_' + test_route['halfTour'].map(str) + '_' +
test_route['stopId'].map(str)
route_list=list(set(test_route['person_tour_stop']) &
set(agent_joint['person_tour_stop']))
agent_joint = agent_joint.set_index('person_tour_stop')
agent_joint.loc[route_list,'path_node_sequence'] = None
agent_joint.to_csv('input_agent.csv')

#Run DTALite to generate time-dependent skims
RunDTALite()

#Convert DATLite skims to CUBE skims
RunCUBE()

#Test model convergence
RunR('ConvergenceScript')
```

Appendix B: SPSA Code

```
"""
A class to implement Simultaneous Perturbation Stochastic Approximation.

@author: Di Yang
"""
import pdb
import numpy as np
from InSIATE_DTALite import *

class SimpleSPSA (object):
    """Simultaneous Perturbation Stochastic Approximation."""
    # These constants are used throughout
    alpha = 0.602
    gamma = 0.101

    def __init__(self, loss_function, a_par=3e-11, noise_var=0.101, args=(),
min_vals=None, max_vals=None, \
        param_tolerance=None, function_tolerance=None, max_iter=1000):
        """The constructor requires a loss function and any required extra
arguments. Optionally, boundaries as well as tolerance thresholds can
be specified.
:param loss_function: The loss (or cost) function that will be minimised.
    Note that this function will have to return a scalar value, not a vector.
:param a_par: This is the ``a`` parameter, which controls the scaling of
    the gradient. It's value will have to be guesstimated heuristically.
:param noise_var: The noise variance is used to scale the approximation
    to the gradient. It needs to be >0.
:param args: Any additional arguments to ``loss_function``.
:param min_vals: A vector with minimum bounds for parameters
:param max_vals: A vector with maximum bounds for parameters
:param param_tolerance: A vector stating the maximum parameter change
    per iteration.
:param function_tolerance: A scalar stating the maximum change in
    ``loss_function`` per iteration.
:return: None
"""
        self.args = args
        self.loss = loss_function
        self.min_vals = min_vals
        self.max_vals = max_vals
```

```

self.param_tolerance = param_tolerance
self.function_tolerance = function_tolerance
self.c_par = noise_var
self.max_iter = max_iter
self.big_a_par = 100
self.a_par = a_par

def calc_loss(self, theta):
    """Evaluate the cost/loss function with a value of theta"""
    retval = self.loss(theta, * self.args)
    return retval

def minimise(self, theta_0, ens_size=1, report=1):
    """The main minimisation loop. Requires a starting value, and optionally
    a number of ensemble realisations to estimate the gradient.
    :param theta_0: The starting value for the minimiser
    :param ens_size: Number of realisations to approximate the gradient.
    :return: A tuple containing the parameters that optimise the function,
            the function value, and the number of iterations used.
    """
    n_iter = 0
    num_p = theta_0.shape[0]
    # print "Starting theta=", theta_0
    theta = theta_0
    j_old = self.calc_loss(theta)
    # Calculate the initial cost function
    theta_saved = theta_0*100
    while (np.linalg.norm(theta_saved-theta)/np.linalg.norm(theta_saved) >=
           1e-8) and (n_iter < self.max_iter):
        # The optimisation carried out until the solution has converged, or
        # the maximum number of iterations has been reached.
        theta_saved = theta # Store theta at the start of the iteration
                           # as we may well be restoring it later on.
        # Calculate the ak and ck scalars. Note that these require
        # a degree of tweaking
        ak = self.a_par/(n_iter + 1 + self.big_a_par)**self.alpha
        ck = self.c_par/(n_iter + 1)**self.gamma
        ghat = 0. # Initialise gradient estimate
        for j in np.arange(ens_size):
            # This loop produces ``ens_size`` realisations of the gradient
            # which will be averaged. Each has a cost of two function runs.
            # Bernoulli distribution with p=0.5
            delta = (np.random.randint(0, 2, num_p) * 2 - 1)
            # Stochastic perturbation, innit
            theta_plus = theta + ck*delta

```

```

theta_plus = np.minimum(theta_plus, self.max_vals)
theta_plus = np.maximum(theta_plus, self.min_vals)
theta_minus = theta - ck*delta
theta_minus = np.maximum(theta_minus, self.min_vals)
theta_minus = np.minimum(theta_minus, self.max_vals)
# Funcion values associated with ``theta_plus`` and
# ``theta_minus``
j_plus = self.calc_loss(theta_plus)
j_minus = self.calc_loss(theta_minus)
# Estimate the gradient
ghat = ghat + (j_plus - j_minus)/(2.*ck*delta)
# Average gradient...
ghat = ghat/float(ens_size)
# The new parameter is the old parameter plus a scaled displacement
# along the gradient.
not_all_pass = True
this_ak = (theta*0 + 1)*ak
theta_new = theta
while not_all_pass:
    out_of_bounds = np.where(np.logical_or(\
        theta_new - this_ak*ghat > self.max_vals,
        theta_new - this_ak*ghat < self.min_vals))[0]
    theta_new = theta - this_ak*ghat
    if len(out_of_bounds) == 0:
        theta = theta - this_ak*ghat
        not_all_pass = False
    else:
        this_ak[out_of_bounds] = this_ak[out_of_bounds]/2.
# The new value of the gradient.
j_new = self.calc_loss(theta)
# Be chatty to the user, tell him/her how it's going...
if n_iter % report == 0:
    print "\tIter %05d" % n_iter, j_new, ak, ck
    print "\tTheta %s" %theta
# Functional tolerance: you can specify to ignore new theta values
# that result in large shifts in the function value. Not a great
# way to keep the results sane, though, as ak and ck decrease
# slowly.
if self.function_tolerance is not None:
    if np.abs(j_new - j_old) > self.function_tolerance:
        print "\t No function tolerance!", np.abs(j_new - j_old)
        theta = theta_saved
        continue
    else:
        j_old = j_new

```

```

# You can also specify the maximum amount you want your parameters
# to change in one iteration.
if self.param_tolerance is not None:
    theta_dif = np.abs(theta - theta_saved)
    if not np.all(theta_dif < self.param_tolerance):
        print "\t No param tolerance!", theta_dif < \
            self.param_tolerance
        theta = theta_saved
        continue
# Ignore results that are outside the boundaries
if (self.min_vals is not None) and (self.max_vals is not None):
    i_max = np.where(theta >= self.max_vals)[0]
    i_min = np.where(theta <= self.min_vals)[0]
    if len(i_max) > 0:
        theta[i_max] = self.max_vals[i_max]*0.9
    if len(i_min) > 0:
        theta[i_min] = self.min_vals[i_min]*1.1
    n_iter += 1
return theta, j_new, n_iter

```

```

def run_spsa(p_in):
    min_theta = [0.01] * 32
    max_theta = [100.0] * 32
    opti = SimpleSPSA(errfunc, min_vals=min_theta, max_vals=max_theta)
    (xsol, j_opt, niter) = opti.minimise(theta0)
    print xsol, j_opt, niter

```

```

if __name__ == "__main__":
    theta0 = np.ones(32)
    run_spsa(theta0)

```


Bibliography

- Adler, T., & Ben-Akiva, M. (1979). A Theoretical and Empirical Model of Trip Chaining Behavior. *Transportation Research Part B: Methodological*, 13(3), 243–257. <https://doi.org/10.1080/14736480490443003>
- Almond, J. (1967). TRAFFIC ASSIGNMENT WITH FLOW-DEPENDENT JOURNEY TIMES. In *Proceedings of the Third International Symposium on the Theory of Traffic Flow*. Operations Research Society of America.
- Antoniou, C., Azevedo, C. L., Lu, L., Pereira, F., & Ben-Akiva, M. (2015). W-SPSA in Practice: Approximation of Weight Matrices and Calibration of Traffic Simulation Models. *Transportation Research Procedia*, 7, 233–253. <https://doi.org/10.1016/j.trpro.2015.06.013>
- Arentze, T. A., Hofman, F., Joh, C. H., & Timmermans, H. (1999). The development of ALBATROSS: some key issues. *Traffic and Mobility, Springer, Berlin, Heidelberg.*, 57–72.
- Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., & Zhang, K. (2016). POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. *Transportation Research Part C: Emerging Technologies*, 64, 101–116. <https://doi.org/10.1016/j.trc.2015.07.017>
- Auld, J., & Mohammadian, A. K. (2012). Activity planning processes in the Agent-based Dynamic Activity Planning and Travel Scheduling (ADAPTS) model.

- Transportation Research Part A: Policy and Practice*, 46(8), 1386–1403.
<https://doi.org/10.1016/j.tra.2012.05.017>
- Balmer, M., Axhausen, K. W., & Nagel, K. (2006). An Agent Based Demand Modeling Framework for Large Scale Micro-Simulations. In *85th annual meeting of the Transportation Research Board* (Vol. 4, pp. 1–17).
<https://doi.org/10.3141/1985-14>
- Balmer, M., Rieser, M., Meister, K., Charypar, D., Lefebvre, N., Nagel, K., & Axhausen, K. (2009). MATSim-T: Architecture and simulation times. *Multi-Agent Systems for Traffic and Transportation Engineering*, 57–78.
<https://doi.org/10.1140/epjb/e2008-00153-6>
- Bar-Gera, H., & Boyce, D. (2003). Origin-based algorithms for combined travel forecasting models. *Transportation Research Part B: Methodological*, 37(5), 405–422. [https://doi.org/10.1016/S0191-2615\(02\)00020-6](https://doi.org/10.1016/S0191-2615(02)00020-6)
- Ben-Akiva, M. E. (1973). *Structure of passenger travel demand models*. Doctoral dissertation, Massachusetts Institute of Technology.
- Ben-Akiva, M. E., Bergman, M. J., Daly, A. J., & Ramaswamy, R. (1984). Modeling inter-urban route choice behaviour. In *Proceedings of the 9th International Symposium on Transportation and Traffic Theory* (pp. 299–330). Utrecht, The Netherlands: VNU Science Press.
- Ben-Akiva, M. E., Bierlaire, M., Koutsopoulos, H., & Mishalani, R. (1998). DynaMIT: a simulation-based system for traffic prediction. In *DACCORD Short Term Forecasting Workshop* (pp. 1–12).
- Ben-Akiva, M. E., & Bowman, J. L. (1998). Activity Based Travel Demand Model

Systems. *Equilibrium and Advanced Transportation Modelling*, 27–46.

https://doi.org/10.1007/978-1-4615-5757-9_2

Bhat, C., Guo, J., Srinivasan, S., & Sivakumar, A. (2004). Comprehensive Econometric Microsimulator for Daily Activity-Travel Patterns. *Transportation Research Record: Journal of the Transportation Research Board*, 1894, 57–66.
<https://doi.org/10.3141/1894-07>

Bowman, J. L. (1998). *The day activity schedule approach to travel demand analysis*. Doctoral Dissertation, Massachusetts Institute of Technology. Retrieved from <http://www.jbowman.net/theses/1998.Bowman.Thesis.pdf>

Bowman, J. L., Bradley, M., Shiftan, Y., Lawton, T. K., & Ben-Akiva, M. (1999). Demonstration of an Activity-Based Model for Portland. *World Transport Research*, 3, 171–184.

Bradley, M., Bowman, J. L., & Griesenbeck, B. (2010). SACSIM: An applied activity-based model system with fine-level spatial and temporal resolution. *Journal of Choice Modelling*, 3(1), 5–31. [https://doi.org/10.1016/S1755-5345\(13\)70027-7](https://doi.org/10.1016/S1755-5345(13)70027-7)

Bradley, M., Outwater, M., Jonnalagadda, N., & Ruiter, E. (2001). Estimation of an activity-based micro-simulation model for San Francisco. In *80th Annual Meeting of the Transportation Research Board, Washington DC* (pp. 1101–1118).

Buliung, R. N., & Kanaroglou, P. S. (2007). Activity–Travel Behaviour Research: Conceptual Issues, State of the Art, and Emerging Perspectives on Behavioural Analysis and Simulation Modelling. *Transport Reviews*, 27(2), 151–187.

<https://doi.org/10.1080/01441640600858649>

Byon, Y.-J., & Liang, S. (2014). Real-Time Transportation Mode Detection Using Smartphones and Artificial Neural Networks: Performance Comparisons Between Smartphones and Conventional Global Positioning System Sensors. *Journal of Intelligent Transportation Systems*, 18(3), 264–272.

<https://doi.org/10.1080/15472450.2013.824762>

Cambridge Systematics, I. (2014). *Dynamic, Integrated Model System: Sacramento-Area Application*. No. SHRP 2 REPORT S2-C10B-RW-1.

Cantarella, G. E., & Cascetta, E. (1995). Dynamic Processes and Equilibrium in Transportation Networks: Towards a Unifying Theory. *Transportation Science*, 29(4), 305–329. <https://doi.org/10.1287/trsc.29.4.305>

Castiglione, J., Bradley, M., & Gliebe, J. (2014). *Activity-Based Travel Demand Models: A Primer*. No. SHRP 2 Report, S2-C46-RR-1.

Chapin, F. S. (1974). *Human activity patterns in the city: Things people do in time and in space*. Chicago: Wiley-Interscience.

Chiao, K.-A., Mohseni, A., & Bhowmick, S. (2006). Lessons Learned from the Implementation of New York Activity- Based Travel Model. In *Innovations in Travel Modelling Conference Proceedings 42* (Vol. 2, pp. 1–207).

Chicago Area Transportation Study. (1959). *Final report in three parts*. Printed by authority of the State of Illinois.

Chiu, Y.-C., Nava, E., Zheng, H., & Bustillos, B. (2011). *DynusT user's manual*.

Chiu, Y., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., & Hicks, J. (2011). *Dynamic Traffic Assignment: A Primer. Transportation Network*

- Modeling Committee*. <https://doi.org/10.1016/j.trd.2016.06.003>
- Daganzo, C., & Sheffi, Y. (1977). On Stochastic Models of Traffic Assignment. *Transportation Science*, *11*(3), 253–274. Retrieved from <http://pubsonline.informs.org/doi/abs/10.1287/trsc.11.3.253>
- de la Barra, T., Perez, B., & Anez, J. (1993). Multidimensional path search and assignment. In *PTRC Summer Annual Meeting*. University of Manchester, United Kingdom.
- Di, X., Liu, H. X., Pang, J. S., & Ban, X. (2013). Boundedly rational user equilibria (BRUE): Mathematical formulation and solution sets. *Transportation Research Part B: Methodological*, *57*, 300–313. <https://doi.org/10.1016/j.trb.2013.06.008>
- Dial, R. B. (1971). A probabilistic multipath traffic assignment algorithm which obviates pathenumeration. *Transportation Research*, *5*(2), 83–111.
- Dijst, M. (1995). *Het elliptisch leven*. Doctoral dissertation. KNAG, Utrecht.
- Dijst, M., & Vidakovic, V. (1997). Individual action space in the city. In Ettema. *Activity-Based Approaches to Travel Analysis*.
- Domencich, T. A., & McFadden, D. (1975). *Urban travel demand-a behavioral analysis*. No. Monograph.
- Dooley, E. (2015). Here's How Much Time Americans Waste in Traffic. Retrieved from <http://abcnews.go.com/US/time-americans-waste-traffic/story?id=33313765>
- Ettema, D., Borgers, A., & Timmermans, H. (1993). Simulation model of activity scheduling behavior. *Transportation Research Record*, 1–11. <https://doi.org/10.3141/1551-12>

- Evans, S. P. (1973). *Some applications of optimisation theory in transport planning*.
 Doctoral dissertation, University College, London.
- FHWA. (1967). *Guidelines for Trip Generation Analysis*. Available from FHWA,
 HHP-22, Washington, D.C. 20590.
- Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval
 Research Logistics (NRL)*, 3 (1-2), 95–110.
- Fried, M., Havens, J., & Thall., M. (1977). *Travel behavior-A synthesized theory*. No.
 Project 8-14 Final Rpt.
- Garling, T., Kwan, M.-P., & Golledge, R. (1994). Computational-Process Household
 Activity Modelling, 28(5), 355–364.
- Gonzalez, P. A., Weinstein, J. S., Barbeau, S. J., Labrador, M. A., Winters, P. L.,
 Georggi, N. L., & Perez, R. (2010). Automating mode detection for travel
 behaviour analysis by using global positioning systems-enabled mobile phones
 and neural networks. *IET Intelligent Transport Systems*, 4(1), 37.
<https://doi.org/10.1049/iet-its.2009.0029>
- Grant, M., Bauer, J., Plakson, T., & Mason, J. (2010). Advancing Metropolitan
 Planning for Operations, 80. Retrieved from
http://ntl.bts.gov/lib/33000/33700/33775/fhwa_hop_10_026.pdf
- Hadi, M., Pendyala, R., Bhat, C., & Waller, T. (2014). *Dynamic, Integrated Model
 System: Jacksonville-Area Application*. No. SHRP 2 REPORT S2-C10A-RW-1.
- Hägerstrand, T. (1970). What About People in Regional Science? *Papers in
 Regional Science*, 24(1), 7–24. <https://doi.org/10.1111/j.1435-5597.1970.tb01464.x>

- Han, Q., & Timmermans, H. (2006). Interactive Learning in Transportation Networks with Uncertainty, Bounded Rationality, and Strategic Choice Behavior: Quantal Response Model. *Transportation Research Record*, 1964(1), 27–34.
<https://doi.org/10.3141/1964-04>
- Ho, T. K. (1995). Random Decision Forests, *Vol. 1*, 278–282.
- Huigen, P. (1986). *Binnen of buiten bereik?: een sociaal-geografisch onderzoek in ZuidwestFriesland*. Nederlandse Geografische Studies 7. Amsterdam/Utrecht: KNAG/Universiteit van Utrecht.
- Jones, P. M. (1977). *New approaches to understanding travel behavior: the human activity approach*. Oxford University.
- Jones, P. M., Dix, M. C., Clarke, M. I., & G., I. (1983). *Understanding travel behaviour*. University of Oxford. Transport Studies Unit.
- Khisty, C. J., & Arslan, T. (2005). Possibilities of steering the transportation planning process in the face of bounded rationality and unbounded uncertainty. *Transportation Research Part C: Emerging Technologies*, 13(2), 77–92.
<https://doi.org/10.1016/j.trc.2005.04.003>
- Kitamura, R., & Fujii, S. (1998). Two Computational Process Models of Activity-Travel Behavior. *Theoretical Foundations of Travel Choice Modeling*, 251–279.
- Kitamura, R., Lula, C. V., & Pas, E. I. (1993). AMOS: An activity-based, flexible and truly behavioral tool for evaluation of TDM measures. In *21st PTRC Summer Annual Meeting*. University of Manchester, United Kingdom.
- Kwan, M.-P. (1997). GISICAS: AN ACTIVITY-BASED TRAVEL DECISION SUPPORT SYSTEM USING A GIS-INTERFACED COMPUTATIONAL-

PROCESS MODEL. *Activity-Based Approaches to Travel Analysis*.

- Lam, W. H. K., & Huang, H.-J. (2003). Combined Activity/Travel Choice Models: Time-Dependent and Dynamic Versions. *Networks and Spatial Economics*, 3, 323–347.
- Lari, A. (2015). Automated Transportation Mode Detection Using Smart Phone Applications via Machine Learning: Case Study Mega City of Tehran. *Transportation Research Board 94th Annual Meeting*, 6147.
- Lee, J., & Ozbay, K. (2009). New calibration methodology for microscopic traffic simulation using enhanced simultaneous perturbation stochastic approximation approach. ... *Research Record: Journal of the Transportation ...*, 2124, 233–240. <https://doi.org/10.3141/2124-23>
- Lemp, J. (2014). A Latent Class Model of Daily Activity Patterns and Joint Activity Participation. In *5th TRB Conference on Innovations in Travel Modeling*.
- Lenntorp, B. (1977). Paths in space-time environments: A time-geographic study of movement possibilities of individuals. *Environment and Planning A*, 9(8), 961–972.
- Lenntorp, B. (1978). A time-geographic simulation model of individual activity programmes. *Human Activity and Time Geography*, 2, 162–180.
- Lin, D.-Y., Eluru, N., Waller, S., & Bhat, C. (2008). Integration of Activity-Based Modeling and Dynamic Traffic Assignment. *Transportation Research Record: Journal of the Transportation Research Board*, 2076, 52–61. <https://doi.org/10.3141/2076-06>
- Lu, L., Xu, Y., Antoniou, C., & Ben-Akiva, M. (2015). An enhanced SPSA algorithm

- for the calibration of Dynamic Traffic Assignment models. *Transportation Research Part C: Emerging Technologies*, 51, 149–166.
<https://doi.org/10.1016/j.trc.2014.11.006>
- Ma, J., Avenue, O. S., Ave, C., & Zhang, H. M. (2007). Calibration of Micro Simulation with Heuristic Optimization Methods University of California , Davis School of Transportation Engineering 1239 Siping Road Submitted to AHB45 (1), TRB Committee on Traffic Flow Theory and Characteristics , Simulation Sub-, 45(1), 1–18.
- Mahmassani, H. (1992). Dynamic traffic assignment and simulation for advanced network informatics (DYNASMART). In *2nd International Seminar on Urban Traffic Networks*.
- Mahmassani, H., & Chang, G.-L. (1987). On Boundedly Rational User Equilibrium in Transportation Systems. *Transportation Science*, 21(2), 89–99.
<https://doi.org/10.1287/trsc.21.2.89>
- Manski, C. F. (1977). The structure of random utility models. *Theory and Decision*, 8(3), 229–254. <https://doi.org/10.1007/BF00133443>
- McFadden, D. (1978). Modeling the choice of residential location. *Transportation Research Record*, 673.
- Nitsche, P., Widhalm, P., Breuss, S., Brändle, N., & Maurer, P. (2014). Supporting large-scale travel surveys with smartphones - A practical approach. *Transportation Research Part C: Emerging Technologies*, 43, 212–221.
<https://doi.org/10.1016/j.trc.2013.11.005>
- Odell, J. (2002). Objects and agents compared. *Journal of Object Technology*, 1(1),

41–53. <https://doi.org/10.5381/jot.2002.1.1.c4>

Parsons Brinckerhof. (2006). *Progress report for the year 2005, regional transportation plan major update project for the Atlanta Regional Commission, general modeling task 13 (activity/tour-based models)*.

Parsons Brinckerhoff. (2005). *The MORPC travel demand model validation and final report*.

Pendyala, R., Konduri, K., Chiu, Y.-C., Hickman, M., Noh, H., Waddell, P., ...

Gardner, B. (2012). Integrated Land Use-Transport Model System with Dynamic Time-Dependent Activity-Travel Microsimulation. *Transportation Research Record: Journal of the Transportation Research Board*, 2303, 19–27.

<https://doi.org/10.3141/2303-03>

Pendyala, R., You, D., Garikapati, V., Konduri, K., & Zhou, X. (2017). Paradigms for integrated modeling of activity-travel demand and network dynamics in an era of dynamic mobility management. *The 96th Annual Meeting of the Transportation Research Board, Accepted for Presentation Only*, (August 2016).

Pinjari, A. R., & Bhat, C. R. (2011). Activity-based travel demand analysis. *A Handbook of Transport Economics, Edward Elg(1)*, 213–248.

<https://doi.org/10.4337/9780857930873.00017>

Rasouli, S., & Timmermans, H. (2014). Activity-based models of travel demand: Promises, progress and prospects. *International Journal of Urban Sciences*, 18(1), 31–60. <https://doi.org/10.1080/12265934.2013.835118>

Rossi, T. F., & Shiftan, Y. (1997). Tour Based Travel Demand Modeling in the U.S. *IFAC Proceedings Volumes*, 30(8), 381–386. <https://doi.org/10.1016/S1474->

6670(17)43853-5

- Rossi, T., & Systematics, C. (2012). Integrated Activity Based Model and Traffic Simulation in the Sacramento Region – the Home Stretch.
- Sheffi, Y. (1985). Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. *Transportation Science*, 19(4), 463–466.
<https://doi.org/10.1287/trsc.1090.0306>
- Simon, H. A. . (1955). A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics*, 69(1), 99–118.
- Sivak, M. (2002). How common sense fails us on the road: contribution of bounded rationality to the annual worldwide toll of one million traffic fatalities. *Transportation Research Part F: Traffic Psychology and Behaviour*, 5(4), 259–269.
- Spall, J. C. (1998). An Overview of the Simultaneous Perturbation Method for Efficient Optimization. *Johns Hopkins Apl Technical Digest Volume: 19, Issue: 4, 19(4)*, 482–492.
- Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011). Transportation mode detection using mobile phones and GIS information. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11*, 54. <https://doi.org/10.1145/2093973.2093982>
- Tang, L., Pan, Y., & Zhang, L. (2018). Trip purpose imputation based on long-term GPS data. In *2018 Transportation Research Board Annual Meeting. Jan 8-11, 2018, Washington D.C.*
- Tang, L., Xiong, C., & Zhang, L. (2015). Decision tree method for modeling travel

- mode switching in a dynamic behavioral process. *Transportation Planning and Technology*, 38(8), 833–850. <https://doi.org/10.1080/03081060.2015.1079385>
- Timmermans, H., & Arentze, T. (2005). Albatross Version 2: A Learning-Based Transportation Oriented Simulation System. *European Institute of Retailing and Services Studies, Eindhoven, The Netherlands*.
- Waller, S. T., & Ziliaskopoulos, A. K. (1999). Visual Interactive System for Transportation Algorithms. In *78th Annual Meeting of the Transportation Research Board, Washington, D.C.*
- Wardrop, J. G. (1952). Road Paper: Some Theoretical Aspects of Road Traffic Research. In *Proceedings of the institution of civil engineers* (p. 1(3): 325-362).
- Wu, L., Yang, B., & Jing, P. (2016). Travel Mode Detection Based on GPS Raw Data Collected by Smartphones: A Systematic Review of the Existing Methodologies. *Information*, 7(4), 67. <https://doi.org/10.3390/info7040067>
- Xiao, G., Juan, Z., & Zhang, C. (2015). Travel mode detection based on GPS track data and Bayesian networks. *Computers, Environment and Urban Systems*, 54, 14–22. <https://doi.org/10.1016/j.compenvurbsys.2015.05.005>
- Xiong, C. (2015). On Agent-Based Modeling : Multi- Dimensional Travel Behavioral Theory , Procedural Models and Simulation-Based Applications.
- Xiong, C., & Zhang, L. (2013). A descriptive bayesian approach to modeling and calibrating drivers' en route diversion behavior. *IEEE Transactions on Intelligent Transportation Systems*, 14(4), 1817–1824. <https://doi.org/10.1109/TITS.2013.2270974>
- Zhang, L. (2006). *Search, Information, Learning, and Knowledge in Travel Decision-*

making: A Positive Approach for Travel Behavior and Demand Analysis.

Doctoral dissertation, University of Minnesota.

Zhang, L. (2011). Behavioral Foundation of Route Choice and Traffic Assignment.

Transportation Research Record: Journal of the Transportation Research

Board, 2254, 1–10. <https://doi.org/10.3141/2254-01>

Zhang, L., Chang, G.-L., Zhu, S., Xiong, C., Du, L., Mollanejad, M., ... Mahapatra,

S. (2013). Integrating an Agent-Based Travel Behavior Model with Large-Scale

Microscopic Traffic Simulation for Corridor-Level and Subarea Transportation

Operations and Planning Applications. *Journal of Urban Planning and*

Development, 139(2), 94–103. [https://doi.org/10.1061/\(ASCE\)UP.1943-](https://doi.org/10.1061/(ASCE)UP.1943-)

5444.0000139

Zhang, L., & Levinson, D. (2004). Agent-Based Approach to Travel Demand

Modeling: Exploratory Analysis. *Transportation Research Record*, 1898(1), 28–

36. <https://doi.org/10.3141/1898-04>

Zhou, X., & Taylor, J. (2014). DTALite: A queue-based mesoscopic traffic simulator

for fast model evaluation and calibration. *Cogent Engineering*, 1(1), 1–19.

<https://doi.org/10.1080/23311916.2014.961345>

Zhu, L., Gonder, J., & Lin, L. (2017). Prediction of Individual Social-Demographic

Role Based on Travel Behavior Variability Using Long-Term GPS Data, 2017.