March 2016

# Adaptive Route Choice in Stochastic Time-Dependent Networks: Routing Algorithms and Choice Modeling

Jing Ding-Mastera
*University of Massachusetts - Amherst*

# ADAPTIVE ROUTE CHOICE IN STOCHASTIC TIME-DEPENDENT NETWORKS: ROUTING ALGORITHMS AND CHOICE MODELING

A Dissertation Presented

by

JING DING-MASTERA

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2016

Civil and Environmental Engineering
Department

# ADAPTIVE ROUTE CHOICE IN STOCHASTIC TIME-DEPENDENT NETWORKS: ROUTING ALGORITHMS AND CHOICE MODELING

A Dissertation Presented

by

JING DING-MASTERA

Approved as to style and content by:

Song Gao, Chair

Moshe Ben-Akiva, Member

John Collura, Member

Jenna Marquard, Member

Richard Palmer, Department Head
Civil and Environmental Engineering
Department

# ABSTRACT

## ADAPTIVE ROUTE CHOICE IN STOCHASTIC TIME-DEPENDENT NETWORKS: ROUTING ALGORITHMS AND CHOICE MODELING

FEBRUARY 2016

JING DING-MASTERA

BE, TIANJIN UNIVERSITY, TIANJIN, CHINA

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Song Gao

Transportation networks are inherently uncertain due to random disruptions; meanwhile, real-time information potentially helps travelers adapt to realized traffic conditions and make better route choices under such disruptions. Modeling adaptive route choice behavior is essential in evaluating Advanced Traveler Information Systems (ATIS) and related policies to better provide travelers with real-time information. This dissertation contributes to the state of the art by estimating the first latent-class routing policy choice model using revealed preference (RP) data and providing efficient computer algorithms for routing policy choice set generation. A routing policy is defined as a decision rule applied at each link that maps possible realized traffic conditions to decisions on the link to take next. It represents a traveler's ability to look ahead in order to incorporate real-time information not yet available at the time of decision.

A case study is conducted in Stockholm, Sweden and data for the stochastic time-dependent network are generated from hired taxi Global Positioning System (GPS) traces

through the methods of map-matching and non-parametric link travel time estimation. A latent-class Policy Size Logit model is specified with two additional layers of latency in the measurement equation. The two latent classes of travelers are policy users who follow routing policies and path users who follow fixed paths. For the measurement equation of the policy user class, the choice of a routing policy is latent and only its realized path on a given day can be observed. Furthermore, when GPS traces have relatively long gaps between consecutive readings, the realized path cannot be uniquely identified.

Routing policy choice set generation is based on the generalization of path choice set generation methods, and utilizes efficient implementation of an optimal routing policy (ORP) algorithm based on the two-queue data structure for label correcting. Systematic evaluation of the algorithm in random networks as well as in two large scale real-life networks is conducted. The generated choice sets are evaluated based on coverage and adaptiveness. Coverage is the percentage of observed trips included in the generated choice sets based on a certain threshold of overlapping between observed and generated routes, and adaptiveness represents the capability of a routing policy to be realized as different paths over different days. It is shown that using a combination of methods yields satisfactory coverage of 91.2%. Outlier analyses are then carried out for unmatching trips in choice set generation. The coverage achieves 95% for 100% threshold after correcting GPS errors and breaking up trips with intermediate stops, and further achieves 100% for 90% threshold.

The latent-class routing policy choice model is estimated against observed GPS traces based on the three different sample sizes resulting from coverage improvement, and the estimates appear consistent across different sample sizes. Estimation results show the policy user class probability increases with trip length, and the latent-class routing policy choice model fits the data better than a single-class path choice model or routing policy choice model. This suggests that travelers are heterogeneous in terms of their ability and willingness to plan ahead and utilize real-time information. Therefore, a fixed path model as commonly used in the literature may lose explanatory power due to its simplified assumptions on network stochasticity and travelers' utilization of real-time information.

# CONTENTS

# LIST OF TABLES

**Table** **Page**

# LIST OF FIGURES

**Figure** **Page**

# CHAPTER 1

# INTRODUCTION

## 1.1 Background and Motivation

Traffic congestion has become a severe problem around the world. According to the 2015 Urban Mobility Report by Texas Transportation Institute (TTI), traffic congestion in the United States (U.S.) cost 160 billion dollars in 2014. Individual travelers suffer from congestion which consumes personal time and increase anxiety. According to TTI, in order to arrive on time for important trips, travelers must now plan for 60 minutes to make a trip that takes 20 minutes in light traffic. As a result, industry also suffers from congestion because of reduced productivity of employees and increased scheduling and shipping costs. This burden of costs are then inevitably passed onto the consumers. Furthermore, congestion induce impacts to the environment from wasted fuel and emissions.

There are generally two types of congestion, recurrent and non-recurrent. Recurrent congestion is caused by the insufficient design capacity of a roadway segment under normal traffic flows. The design capacity is based on the projected flows at the time of design and may not meet the actual flows at a later time due to prediction errors. Non-recurrent congestion is due to random disruptions that reduce the capacity, such as incidents, vehicle breakdowns, bad weather, work zones, special events and so forth. Transportation systems are frequently subject to these random disruptions resulting in variable and unpredictable traffic conditions. Thus non-recurrent congestion contributes significantly to the total congestion. It is estimated by the Federal Highway Administration (FHWA) that 50% of the congestion in the U.S. is due to unexpected disruptions to the system.

An important characteristic of congested transportation networks is that traffic variables such as travel time and flows are stochastic and time-dependent from time to time and day to day. For example, travel time on a weekday could be very different from that on weekends, and travel time during morning peak could be very different from that during mid-day. This characteristic comes from both recurrent and non-recurrent congestion. Non-recurrent congestion contributes through random disruptions to the networks while re-current congestion contributes through the fluctuations in number of trips and the spread of trips over departure times. Travelers with flexible time may shift their departure times to avoid peak hours. and travelers with non-work trip purposes may cancel a trip on a particular day. Traffic variables not only are stochastic and time-dependent, but also have link-wise and time-wise dependency. For example, when an incident happens on the highway, the link travel times around the incident location and within the incident duration are correlated. Thus when modeling transportation networks, it is important to capture these characteristics.

It is widely acknowledged that expanding the current infrastructure to provide increased capacity, which is typically financially and environmentally constrained, is no longer the only solution for congestion. New measures to relieve congestion are based on the concept of making the best use of the current infrastructure, which is the underlying idea of Intelligent Transportation Systems (ITS). The federal government and many states have developed such measures to mitigate congestion. One example is a new strategy toward incident management which provides advanced training of emergency response teams in an effort to reduce the disruption time. While such strategies may help reduce the impacts of disruptions, a more effective solution may be to provide the travelers with information regarding the disruptions. With the fast development of sensor and telecommunication technologies, real-time information is increasingly available for travelers and system operators to make better decisions in such an uncertain system with random disruptions. As a result, Advanced

Traveler Information Systems (ATIS) have been identified as a potential strategy toward reducing congestion and improving system reliability.

There are various information types and information access types. Information can be classified as a priori information and online information. A priori information is available before a trip, which provides the average values and variability of the traffic variables across days, e.g. the average travel time through a tunnel is 1 minute, but the travel time could be very high due to incidents which happen once a week on average. Online information is available during a trip, which provides the values of traffic variables on a specific day and time, e.g. currently, the travel time through the tunnel is 30 minutes due to a crash. When the network is stochastic and time-dependent, online information can be very different from a priori information and that is when online information is beneficial in assisting travelers avoiding the congestion. ATIS can provide both a priori and online information to travelers. There are also different types of information access. Smartphone apps such as Google Traffic and Waze usually contain a database of all roadways and can provide travelers with travel times on multiple alternative routes. Variable message signs (VMS) usually provide travelers with travel times of the roadway segment they are traveling on. Different assumptions should be made based on different types of information access.

The success of ATIS is based on the idea that an better informed traveler can make a better decision, and collectively many travelers' better decisions can reduce congestion. For instance, if information regarding delays on the highway due to an incident is timely provided to the travelers who plan to take this route, they can make rerouting decisions which result in a relief in congestion. However, if travelers do not adapt to real-time information and wait in line for a long time, then congestion certainly cannot be alleviated by providing real-time information. It is therefore crucial to study whether an individual traveler makes adaptive route choice decisions utilizing real-time information and how they make such decisions, and furthermore, what is the network-level impact if many travelers make adaptive route choice decisions.

From a network-level planning point of view, the route choices made by travelers have immense effects on where congestion takes place and the collective of all travelers' choices determines the traffic load and distribution, and how we develop transportation policies to handle the ever growing travel demand. As part of regional transportation planning activities, travel demand modeling (TDM), which estimates travel behavior and travel demand for a future time frame, is a powerful tool to assist decision makers in making informed decisions on proposed plans and policies. Route choice modeling is an essential part in TDM, but the route choice models applied in TDM are usually very basic with simplified assumptions of fixed path choice behavior. In a congested network, however, the traffic conditions affect travelers route choice behavior, which in turn affects the traffic conditions. Although this interaction between supply and demand can be captured by a conventional dynamic traffic assignment (DTA) model in a deterministic and static network with fixed path choice assumptions, it cannot be captured in a stochastic time-dependent network without a more realistic route choice model. The development of ATIS in assisting travelers' decisions has renewed the interest of developing more sophisticated route choice models to model the real-time information effect and adaptive route choice behavior.

While it may seem simplistic to implement ATIS, the effects to the transportation network must be evaluated. For example, the installation of VMS informing travelers of delays and alternative routes due to an incident on a highway is expected to reduce congestion and hence system costs. Travelers route choice decision with information are generally better from those without information, and collectively better decisions by many travelers would result in less congestion. Yet it is also anticipated that not all drivers will utilize the information being provided. Furthermore, the induced traffic diversions could have a direct impact to the local roadways. Therefore, a crucial component in designing and evaluating ATIS is an understanding of travelers adaptive route choice behavior in response to a wide range of traveler information access in a network with dynamic and random traffic conditions. This understanding could be utilized to predict the effectiveness of ATIS and provide justification of its costs associated with installation and operation. The cost-benefit analysis of ATIS requires a model which allows travelers to make adaptive route choice decisions with the real-time information and derives metrics on network performance such as total travel time and travel time reliability. These metrics can then be translated to monetary values and compared to those in the network without ATIS.

Then what are the scopes of adaptive route choice behavior? A traveler makes decisions based on his or her knowledge of the available alternatives and their attributes. This knowledge is periodically updated by both personal experience and exogenous information, and as a result the decisions might be revisited and revised. In other words, a traveler "adapts" to the decision environment. The time scale at which route choice adaption happens can be broadly divided into two types: day-to-day and within-day. In a day-to-day context, a traveler's route choice today might be different from yesterday due to information collected yesterday during the trip. In a within-day context, route choice could be revised en route, e.g., taking a detour upon receiving information on a crash on the bridge along the original route. This dissertation focuses on within-day adaptive route choice, where the real-time information reflects travel conditions at or close to the decision time. For a recent review of empirical day-to-day route choice studies, please refer to [Balakrishna et al., 2013].

Within-day route choice is arguably the most researched area of traveler response to ATIS, which predicts the route a traveler would take when traveling between an Origin-Destination (OD) pair. Three types of route choice models (fixed path models, adaptive path models, and routing policy choice models) have been studied based on three types of travelers (non-adaptive, reactive, and strategic, respectively). Early on, most studies focused on fixed path models. They assume that a traveler is non-adaptive who chooses a fixed path at the origin of a trip and follows it till the end, not accounting for any real-time information provided en-route. Travelers' route choice behavior in an uncertain network with real-time information will conceivably be different from that in a deterministic network.

With real-time information provided en-route, travelers could make route choice decisions at intermediate nodes based on the current situation in order to avoid delay downstream. Thus over time, there was a new focus on adaptive path models. An adaptive path model assumes that a traveler is reactive and route choice is a series of path choices at each node. A reactive traveler behaves as if no additional information or diversions would happen in the future and choose from the set of paths from the current node to the destination, even though in reality they might switch to another path at the next node. Although an adaptive path model could account for diversions from an initial chosen path, it assumes that travelers are simply reactive to information on-the-spot and do not plan ahead for real-time information that will be available later in the trip. Most recently, a few studies focused on routing policy choice models, which assume travelers are strategic in planning ahead for future events and travelers have some expectations for the real-time information downstream. Both reactive and strategic travelers adapt to real-time information at intermediate nodes; the difference between a reactive and strategic traveler lies on whether the decision at an intermediate node takes into account future information availability and diversion possibilities. A strategic traveler fully considers the future information availability and the possible actions they might take at all future nodes. Therefore they decide what next link to take but not the next path to destination. However, studies on routing policy choice models have only been carried out with stated preferences (SP) data, where individuals state their preference under experimental conditions such as surveys. Therefore, there has been a gap in the literature that this dissertation now addresses: the estimation of a routing policy choice model under real-time information using revealed preference (RP) data, where individuals reveal their preference through the choices already made such as observed GPS traces. A recent overview of models that account for real-time information en-route can be found in [Abdel-Aty & Abdalla, 2006], [Balakrishna et al., 2013], and [Chorus et al., 2006].

## 1.2 Literature Review

### 1.2.1 Optimal Routing Policy Algorithms

The optimal routing policy (ORP) calculation is a building block of any choice set generation algorithm. There are only a limited number of studies exploring the ORP problem in STD networks with explicitly represented time and various assumptions on network stochasticity and information availability. Some studies investigate real-life networks, but have simplified assumptions on network stochasticity and information availability. Others study such complicated problems but are not so practical as to be applied to real-life networks.

[Hall, 1986] studied for the first time the time-dependent version of the ORP problem and demonstrates in an STD network, adaptive decision rules are more effective than fixed paths. [Chabini, 2000] produced a dynamic programming algorithm based on the concept of decreasing order of time, which is optimal in the sense that no algorithms with better worst-time complexity exist. The algorithm is later described in [Gao, 2004]. [Miller-Hooks & Mahmassani, 2000a] developed a label-correcting algorithm for determining the adaptive least-expected time hyperpaths from all nodes to a select destination assuming time-wise and link-wise stochastically independent link travel time random variables. [Miller-Hooks, 2001] compared this algorithm with the dynamic programming algorithm in [Chabini, 2000] computationally. It is Similar to the TDLTP (Time-dependent Least-time Problem) algorithm in [Ziliaskopoulos & Mahmassani, 1993] for determining least-time paths in deterministic, time-dependent networks. In [Boyles, 2009], a label-correcting algorithm is proposed for the online shortest path problem in cyclic graphs and is demonstrated on a medium-sized transportation network. [Boyles & Waller, 2011] described a network contraction procedure and demonstrated adaptive routing algorithms on three networks of varying size. In [Boyles, 2012], an approach is presented for replacing a regional network with a smaller one, and is applied to a regional network representing the Austin, Texas metropolitan area. [Boyles & Waller, 2010a] and [Boyles & Waller, 2010b] studied traffic assignment problems in real-life network.

[Gao & Chabini, 2006a] studied the ORP problems in STD networks, where link travel times are modeled as random variables with time-dependent distributions. It established the first framework for ORP problems in STD networks, providing a comprehensive taxonomy of the studied problem, based on information access and network stochastic dependency, designing an algorithm (Algorithm DOT-SPI) to one of the variants, particularly pertinent in traffic networks, where the network dependency and the value of information are taken into account. However, Algorithm DOT-SPI is not intended to be deployed in practice, as in practice it is difficult to obtain the a priori joint realization and the run time of the algorithm is high due to the fact that the problem variant it solves is an intrinsically difficult problem.

Based on Algorithm DOT-SPI, Algorithm LC-CDPI has been designed in [Ding & Gao, 2012], which is the first algorithm practical for large-scale networks that considers complete time-wise and link-wise stochastic dependency. This algorithm is introduced in Section 3

### 1.2.2 Choice Set Generation

The first step in route choice modeling is choice set generation. For real-life networks, there may exist a large number of routes/routing policies for an origin-destination (OD) pair. Thus a subset of reasonable alternatives needs to be generated. For path choice set generation there are deterministic and stochastic algorithms [Frejinger, 2007]. Commonly used deterministic methods include link elimination [Azevedo & Martins, 1993], [Srinivasan & Dhakar, 2013], [Rieser-Schussler & Axhausen, 2012], link penalty [de la Barra & Anez, 1993], labeling [Ben-Akiva & Ramaswamy, 1984], constrained k-shortest paths [Van der Zijpp & Fiorenzo-Catalano, 2005], and branch-and-bound ( [Friedrich & Wekeck, 2001] for public transportation networks, [Hoogendoorn-Lanser, 2005] for multi-modal networks and [Prato & Bekhor, 2006] for private transportation networks). Stochastic methods include simulation [Ramming, 2002], [Prato, 2011], and doubly stochastic choice set generation [Bovy & Fiorenzo-Catalano, 2006]. [Bekhor & Ramming, 2006] provides a comprehensive comparison of a large number of path choice set generation algorithms using a data set from Boston. [Prato, 2012] suggests that transportation modelers should implement stochastic path generation techniques with average variance of its distribution parameters, and correction for unequal sampling probabilities of the alternative routes in order to obtain satisfactory results for coverage of "postulate chosen route", and reproduction of "true model estimates". A number of routing policy choice set generation algorithms are investigated, which are generalizations of the link elimination and simulation methods for path choice set generation.

### 1.2.3 Route Choice Models

#### 1.2.3.1 Fixed Path Models

Fixed path models refer to probabilistic route choice models in deterministic networks which differ from deterministic route choice models. In deterministic route choice models, the probability of choosing a path is 100%. Deterministic route choice models can also be applied in STD networks with attributes being random variables. This dissertation focuses on probabilistic route choice models.

Once the choice sets are generated, the next step in route choice modeling is the estimation of a route choice model which predicts the route a traveler would take when traveling between an OD pair. Most route choice models are only based on deterministic networks. They assume that a traveler makes a complete route choice at the origin of a trip and do not account for any real-time information provided en-route. For example, early on the most widely used model is the Multinomial Logit (MNL) due to its attractive closed-form formula. However, the MNL model assumes the error terms are identically and independently distributed (i.i.d) which are not true in networks with many overlapping paths. To address this issue, the C-Logit model [Cas, 1996] and Path Size Logit model

[Ben-Akiva & Bierlaire, 1999] are proposed based on the logit model. Notably, the Path Size Logit model has been successfully implemented based on RP data in Dynamic Traffic Assignment (DTA) in a subnetwork of Beijing, China [Ben-Akiva et al., 2012]. Later on, more complicated models are developed such as Error Component model [Bolduc & Ben-Akiva, 1991], Multinomial Probit [Yai et al., 2002], latent route choice models with network-free data [Mahmassani, 2001], models assuming a universal choice set estimated based on a sampling approach [Frejinger et al., 2009], and models assuming a universal choice set estimated trough repeated link choices based on a dynamic discrete choice approach [Fosgerau et al., 2013]. Due to the complexity of these models, however, they have yet to be applied in routing policy choice models in uncertain networks based on RP data.

#### 1.2.3.2 Adaptive Path Models

Some researchers model adaptive route choice behavior by successively estimating a sequence of non-adaptive path choice models at each node and updating the attributes of alternative paths to the destination to reflect real-time information. In principle all fixed path models mentioned in 1.2.3.1 can be applied this way. The simulation-based traffic prediction models in DynaMIT [Ben-Akiva et al., 2011] and DYNASMART [Fosgerau et al., 2011] are such examples, which update the path choice at intermediate decision nodes according to the latest network travel times. These models are calibrated over aggregate measurements such as counts and speeds, in which route choice parameters are among the calibration variables. In this regard, adaptive path models have been calibrated as a part of traffic prediction models in real-life networks.

A large body of research on route choice in response to real-time information focuses on binary route switchings in real life, e.g., [Polydoropoulou et al., 1996], [Chatterjee & McDonald, 2004], [Peeta & Ramos, 2006], [Tsirimpa et al., 2007], or more advanced hypothetical ATIS in SP surveys, in which travelers are directly asked about their preferences for route choice in hypothetical situations, e.g., [Mahmassani & Liu, 1999], [Srinivasan & Mahmassani, 2003], [Abdel-Aty & Abdalla, 2004], [Peeta & Yu, 2005], [Bogers et al., 2005], [Abdel-Aty & Abdalla, 2006]. In all of these studies, travelers are assumed to respond to real-time information on the spot, and the complete decision process is a series of path choices, each of which is based on updated traffic conditions revealed by real-time information at the time of decision. The implicit assumption is that a traveler is myopic and cannot look ahead for future information and such behavior is adaptive path choice.

#### 1.2.3.3 Routing Policy Choice Models

Recent studies investigate the response before the information is received for travelers with looking-ahead abilities. They find that a traveler does not need to commit to a particular route, but instead can decide later at a switching point based on then revealed traffic conditions, and choose the route with a shorter travel time for the remaining trip. The option value of downstream real-time information thus could potentially make a collection of alternatives that share a common starting link more attractive than other links out of the same decision node. Therefore, the travelers respond to the information upstream of the actual point where it is received. Such travelers are said to take routing policies, which loosely speaking, are decision rules applied at each link that map possible realized traffic conditions to decisions on the link to take next. It represents travelers' ability to look ahead in order to incorporate real-time information not yet available at the time of decision. The first routing policy choice model was developed in [Gao, 2005] estimated based on synthetic data and a simplified network. Later [Gao et al., 2008] studied two types of models that account for travelers' adaptation to real-time information: an adaptive path model and a routing policy choice model also based on synthetic data and a simplified network. Empirical studies of the routing policy choice to this date have only been carried out with SP data, e.g. [Razo & Gao, 2010], [Razo & Gao, 2013], [Tian et al., 2011].

#### 1.2.3.4 Route Choice without Choice Set Generation

Some studies avoid choice set generation in the context of route choice model estimation and assume a universal choice set. [Frejinger et al., 2009] presents a new paradigm for choice set generation by assuming that the choice sets contain all paths connecting each OD pair, and a sampling approach was proposed to generate subsets of paths suitable for model estimation. Despite the fact that this approach avoids the bias in model estimation, the application is not feasible when calculating the probability of routes in the universal choice sets. [Fosgerau et al., 2013] proposes a dynamic discrete choice approach for consistently estimating route choice model parameters based on path observations through repeated link choices. The approach does not require choice set generation or sampling. It currently only applies to non-adaptive path choices.

### 1.2.4 Travel Time Reliability

Many different approaches exist for modeling reliability (variability). The most common approach is to assume that travelers see reliability as a direct source of inconvenience, similar to the way travel time is viewed. Furthermore, different model forms and measures of reliability have been explored to study the values of reliability. Reliability ratio is the ratio of reliability and travel time. Studies have shown that model forms influence the reliability ratio ( [Yan, Yan], [Prato et al., 2014]) and so does the method of measuring reliability. The most common measure is travel time standard deviation [Small et al., 1995]; others include the difference between the 90th percentile and the 50th percentile of the travel time distribution [Lam & Small, 2001], the difference between the 80th percentile and the 50th percentile [Small et al., 2005], and the coefficient of variation [Small et al., 1995]. A large number of studies have estimated the value of the reliability ratio. Results range from about 0.10 to over 3.00 with differences across model types, measures of reliability, data types, time periods, and trip purposes. ( [Small et al., 1999], [Batley & Ibanez, Batley & Ibanez], [Li et al., 2010], [Prato et al., 2014], [Yan, Yan], [Lam & Small, 2001], [Ghosh, 2001]). Table 1.2 shows a summary of some recent reliability ratio studies.

The estimated ratios in RP studies ( [Lam & Small, 2001], [Small et al., 1995], [Liu et al., 2007], [Li et al., 2010]) are generally higher than the ones from SP studies ( [Small et al., 1995], [Small et al., 1999], [Small et al., 2005]). [Ghosh, 2001] and [Yan, Yan] find RP estimates to be of higher value in comparison to SP estimates. [Small et al., 2005] concluded that SP studies underestimate the values compared to those in RP studies. [Hensher, 2010] investigation suggests that part of the difference may be due to the method in which the SP data is used in model estimation. The SP experiment design, particularly the way of presenting of travel time reliability, also has a critical influence on the valuation of travel time reliability [Batley & Ibanez, Batley & Ibanez]. In addition, some studies found the trip purposes affects reliability ratio. For example, [Li et al., 2010] showed that the disutility incurred when arriving early is higher than arriving later for non-commuters; while commuters would pay much more to avoid a late arrival, given the consequence of being late. Furthermore, some other studies show that time periods affect reliability ratio. ( [Prato et al., 2014], [Liu et al., 2007]) showed that the value of reliability and travel time are higher for the peak period due to possible penalties for being late and consequently possible time pressure.

**Table 1.1.** Summary of model studies

| | **Fixed Path Models** | | **Adaptive Path Models** | | **Routing Policy Models** | |
|---|---|---|---|---|---|---|
| | SP Data | RP Data | SP Data | RP Data | SP Data | RP Data |
| Hypothetical Simplified Networks | Yes | Yes | Yes | Yes | Yes | Yes |
| Real-Life Networks | Yes | Yes | Yes | Yes | Yes | **No** |

**Table 1.2.** Summary of travel time reliability studies

| Paper | Data Type | Measure of Reliability | Coefficient/ Value of Time ($/hour) | Coefficient/ Value of Reliability | Reliability Ratio |
|---|---|---|---|---|---|
| Small et al (1995) | SP | Coefficient of Variation | -0.1051 | -0.3463 | NA |
| Small et al (1995) | SP | Standard Deviation | -0.0996 | -0.1263 | 1.27 |
| Small et al (1999) | SP | Standard Deviation | 3.92 | 12.6 | 3.23 |
| Ghosh (2001) | SP/RP | 90th-50th percentile | 11.81(Joint) 36.06(RP) | 12.50(Joint) 47.51(RP) | 1.06(Joint) 1.32(RP) |
| Hensher (2001) | SP | Uncertainty or contingency[1] | NZ$8.7 | NZ$5.0 | 0.57 |
| Lam and Small (2001) | RP | 90th-50th percentile | 22.87 | 15.12(Male) 31.91 (Female) | 0.66(Male) 1.39 (Female) |
| Yan (2002) | RP | 80th-50th percentile | 16.40-17.16 | 18.52-33.39 | 1.10-1.95[2] |
| Small et al (2005) | RP | 80th-50th percentile | 21.46 | 19.56 | 0.91 |
| Hollander (2006) | SP(bus) | Standard Deviation | £4.2 | £0.42 | 0.1 |
| Lui (2007) | RP | 80th-50th percentile | 7.7-31.44 | 21.34-39.99 | 0.93-3.16[3] |
| Batley and Ibanez (2009) | SP(Rail) | Standard Deviation | £15.4 | £31.8 | 2.07 |
| Li et al (2010) | SP | Standard Deviation | 22.52 (Commute) 8.18 (Non-commute) | 35.87 (Commute) 21.04 (Non-commute) | 1.59 (Commute) 2.57 (Non-Commute) |
| Prato et al (2014) | RP | 90th-50th percentile | DDK148.67 (Peak) DDK76.55 (Off-peak) | DDK228.95 (Peak) DDK115.59 (Off-peak) | 1.54 (Peak) 1.51 (Off-peak) |

[1]Uncertainty of travel time is defined as the extra time the traveler allowed for in order to ensure he/she arrived at the destination at the planned time. It is equivalent to a contingency time

[2]Varies across model types (ordered-logit with different ways of categorizing the period [0,1])

[3]Varies across time periods (5am-10am)

## 1.3 Challenges and Contributions

In the literature, fixed path models and adaptive path models introduced in Section 1.1 have been estimated in both hypothetical simplified networks and real-life networks based on both SP and RP data. Routing policy choice models have also been estimated in hypothetical simplified networks based on SP data. However, the estimation of routing policy choice models in real-life networks under real-time information using RP data remains an unexplored area.

This dissertation thus contributes to the state of art by conducting the first RP study of routing policy choice using Global Positioning System (GPS) data, specifically in the following two aspects:

- Estimation of the first adaptive route choice model using RP data, where a potential adaptive traveler is provided with real-time information. A latent-class, latent-choice, latent-path Policy Size Logit model is used to capture strategic behavior. This is a significant step toward building a more accurate traffic prediction model under real-time information.

- The design and implementation of computer algorithms for choice set generation in real-life large STD networks, where link travel times are stochastically dependent random variables. Such algorithms are not only critical to the model estimation in the current research, but can also potentially be applied in other route choice and traffic network research due to its generality.

RP studies of adaptive route choice (adaptive path and routing policy choice) in real life networks impose challenges in data collection. Three major types of data are needed: travelers' chosen routes, network conditions, and travelers' information access. The first two types of data are increasingly more available due to the advent of GPS technologies. For example, [Papinski et al., 2009] compared the planned and actually chosen routes (observed by GPS) of travelers and found that 20% of surveyed travelers switch routes for various reasons (one of them being ATIS). There have been a large number of GPS data collection efforts throughout the world, e.g., [Spissu & Sanjust, 2011], [Pillat & Friedrich, 2011] [Rieser-Schussler & Axhausen, 2012], especially with the ever increasing popularity of GPS-enabled smart phones. A number of route choice models have been estimated using GPS data, e.g., [Frejinger & Bierlaire, 2007]. The third type of data, however, is not available from passive GPS traces especially in a dynamic context where the information content changes over time and space, and must be collected through surveys or other monitoring devices such as video cameras. In the case study of this dissertation, taxi GPS traces from Stockholm, Sweden are used to generate chosen routes and link travel times.

Another challenge of RP routing policy choice studies is computational. The underlying network for adaptive route choice is conceivably more complicated than that for a fixed path choice model, as travel times are usually represented as time-dependent random variables to support modeling the dynamic and adaptive nature of the behavior. The alternative in the choice set is a routing policy, and the choice set generation thus requires repetitive executions of an ORP algorithm, which in general is much more time consuming than shortest path algorithms in deterministic networks. There have been a large number of algorithmic studies which generate optimal routing decisions depending on traffic conditions revealed by real-time information in a stochastic network, e.g., [Miller-Hooks & Mahmassani, 2000b], [Waller & Ziliaskopoulos, 2002], [Gao & Chabini, 2006b], [Gao & Huang, 2012]. An efficient ORP algorithm applicable in large-scale real-life networks is developed and applied in this dissertation ( [Ding & Gao, 2012]).

## 1.4 Dissertation Structure

This dissertation has been structured as follows. Chapter 2 begins by introducing the modeling framework including model specification and estimation, and choice set generation. Chapter 3 presents the ORP algorithm LC-CDPI as a building block of choice set generation. Chapter 4 presents a network data processing methodology, which may be used to process real-life network data for the modeling framework. In Chapter 5 these methodologies are implemented in a real-life network, Stockholm, Sweden. In Chapter 6 the conclusions are made and anticipated results and future directions are introduced.

# CHAPTER 2

# MODELING FRAMEWORK AND METHODOLOGIES

## 2.1 Network, Information, Route Choice Behavior

The networks are modeled as stochastic time-dependent (STD), in which link travel times are jointly distributed time-dependent random variables. The STD network is denoted as $G = (N, A, T, P)$, where $N$ is the set of nodes, $A$ the set of links with $|A|$ = $m$, $T$ the set of time periods $\{0, 1, ..., K - 1\}$, and $P$ the probabilistic representation of link travel times. Beyond time period $K - 1$, travel times are static and deterministic. $T(i, j, k, t)$ is the deterministic turning movement penalty from link $(i, j)$ to link $(j, k)$ at time $t$.

Belonging to the link travel time representation, a "support point" is defined as a distinctive value that a discrete random variable can take, or a distinctive vector of values that a discrete random vector can take depending on the context. Thus a probability mass function (PMF) of a random variable (or vector) is a combination of support points and the associated probabilities. A joint probability distribution of all link travel time random variables is used: $P$ = $\{v_1, v_2, ..., v_R\}$, where $v_i$ is a vector with a dimension $K \times m$, $i = 1, 2, ..., R$, and $R$ is the number of support points. The $r^{th}$ support point has a probability $p_r$, and $\sum_{r=1}^{R} p_r$ = 1. When link travel time observations from multiple days are available, a support point can be viewed as a day, $R$ is the number of days, and $p_r$ = $1/R$, $\forall r$.

The example in Figure 2.1 and Table 2.1 illustrates the concepts of support points and event collections. This STD network has three links and three time periods: 8:00 AM, 8:20

AM, and 8:40 AM. Therefore there are nine random travel time variables, each for a link at a time period. Each support point, $r^1$, $r^2$ and $r^3$, has a probability of 1/3.

**Table 2.1.** Example of support points and event collections

| Time | Link | $r^1$ | $r^2$ | $r^3$ |
|---|---|---|---|---|
| 8:00 AM | (0,1) | 20 | 20 | 20 |
| | (1,2) | 30 | 30 | 30 |
| | (0,2) | 40 | 40 | 40 |
| 8:20 AM | (0,1) | 20 | 20 | 20 |
| | (1,2) | 30 | 30 | 20 |
| | (0,2) | 40 | 40 | 30 |
| 8:40 AM | (0,1) | 20 | 20 | 30 |
| | (1,2) | 20 | 30 | 20 |
| | (0,2) | 40 | 30 | 30 |

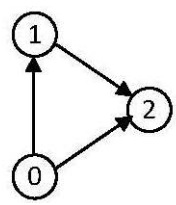


**Figure 2.1.** Network for the example of support points and event collections

With the help of online information, the travelers become more certain about the future or the network becomes less stochastic. To model this effect of information in reducing uncertainty, the concept of the event collection is introduced as a set of support points that are compatible with the revealed information at a given node and time. For instance, a traveler starts a trip at 8:00 AM. At the departure time, all the support points are the same, so the traveler is not sure which support point the network is in. Therefore event collection at 8:00 AM contains the set of all three support points. By the time 8:20 AM, there are two possibilities in the network - if the link travel times are 20, 30, 40, the traveler knows the network is in support point 1 or 2; if they are 20, 20 and 30, then the traveler knows the network is for sure in support point 3. Thus there are two event collections at 8:20 AM with one event collection containing the first two support points and the other containing the third support point. By the time 8:40 AM, if the traveler is in the first event collection at 8:20 AM and if they learn additional information that travel time is now 20, 20, and 40, they then know we are in support point 1, or otherwise we are in support point 2; if the traveler is in the second event collection at 8:20 AM, since it has already been a singleton they do not need any additional information. Therefore, there are three event collections at 8:40 AM each containing one support point and the network has become deterministic.

Real-time information is assumed to include realized travel times of certain links at certain time periods. For example, perfect online information (POI) includes realized travel times on all links up to the current time, while global pre-trip information includes realized travel times of all links up to the departure time. See [Gao & Huang, 2012] for discussions on a number of real-time information access. The passive GPS traces of taxi drivers used in this study cannot tell us what real-time information the drivers have. POI is assumed, since taxi drivers are in general highly sensitive to traffic conditions and stay informed at all times. The discussion in the remainder of the dissertation is therefore specific to POI.

At a given time period $t$, the available real-time information is represented by a joint realization of travel times on all links at time periods $0, 1, \ldots, t$. The joint realization corresponds to a unique subset of compatible support points, defined as an event collection, $EV$, which represents the conditional distribution of link travel times given the realization of link travel times. As more information becomes available, the size of an event collection decreases or remains the same. When an event collection becomes a singleton, the network becomes deterministic.

When a traveler is at the start of link $(i, j)$ at time $t$ with event collection $EV$, he/she makes a decision to take the next link $(j, k)$. Upon arrival at node $j$ (end of link $(i, j)$), he/she will be in a different time period due to the traversal time on link $(i, j)$ and the turning penalty $T(i, j, k, t)$. He/she will also have a potentially different event collection $EV'$, which accounts for realized link travel times between $t$ and the arrival time at node $j$. He/she continues the routing decision process based on dynamically involved event collections. Define $x$ as a state with three elements: link $(i, j)$, time $t$ and event collection $EV$. A routing policy $\mu$ is therefore defined as a mapping from all possible states to the decision of the link to take next, $\mu : x \mapsto (j, k)$.

A routing policy can capture traveler's looking-ahead capability in that the decision at state $x$ depends on the evaluation of all possible future states throughout the remainder of the trip by following each outgoing link. Specifically, the fact that more information will be available in the future is represented by the series of $EV'$ that could be encountered. A routing policy is realized as a path on a given support point (day), and the realized path topologies potentially vary from day to day due to the randomness of travel times and information.

## 2.2 An Illustrative Example of Routing Policy Choice

A network example is designed in Figure 2.2 and Table 2.2 to illustrate the concept of routing policy choice in a STD network. The network consists of two time periods, six nodes including a dummy node $d'$, and six links including a dummy link 6 going out of the destination $d$ with a zero travel time. There are two support points, each with a probability of 0.5, for the joint distribution of ten travel time random variables (links 1, 2, 3, 4, and 5 at time periods 0 and 1). Travel time beyond time period 1 are the same as those in time period 1 in either of the two support points. All turning penalties are assumed to be zero. Two paths are available: link 1 - link 2 - link 4 (path 1) and link 1 - link 3 - link 5 (path 2). At time 0, there is only one possible event collection ($v_1$, $v_2$), as travel times on all links are the same across the two support points. At time 1, there are two possible event collections, $v_1$ and $v_2$.

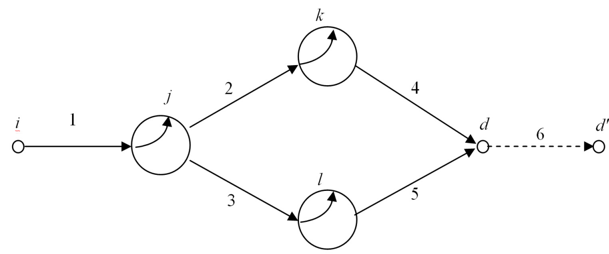


**Figure 2.2.** A stochastic time-dependent network example

**Table 2.2.** Support points and event collections for the network example

| Time | Link | $v_1$ | $v_2$ |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| | 2 | 2 | 2 |
| | 3 | 3 | 3 |
| | 4 | 1 | 1 |
| | 5 | 1 | 1 |
| 1 | 1 | 1 | 2 |
| | 2 | 3 | 2 |
| | 3 | 2 | 2 |
| | 4 | 1 | 1 |
| | 5 | 1 | 1 |

In Figure 2.2, consider the following routing policy: the traveler starts with an initial state $\{1, 0, (v_1, v_2)\}$ and takes link 2; at the beginning of link 2 (node $j$) two states $2, 1, v_1$ or $2, 1, v_2$ are possible. At $2, 1, v_1$ the traveler first takes link 4 and arrives at node $k$ with a state of $4, 4, v_1$, and then takes link 6 and arrived at the destination with a final state of $6, 5, v_1$. At $2, 1, v_2$ the traveler first still takes link 4 and arrives at node $k$ with a state of $4, 3, v_2$, and then takes link 6 and arrives destination with a final state of $6, 4, v_2$. This is represented intuitively in Figure 2.3 as a "state tree" for routing policy. Figure 2.3 also includes the other three routing policies, where $\mu_4$ is not adaptive to states and simply fixed paths. It is also illustrated that multiple routing policies can be realized as the same observed states, e.g., $\mu_1$ and $\mu_2$ are realized as the same states for support point 1; $\mu_1$ and $\mu_3$ are realized as the same states for support point 2.

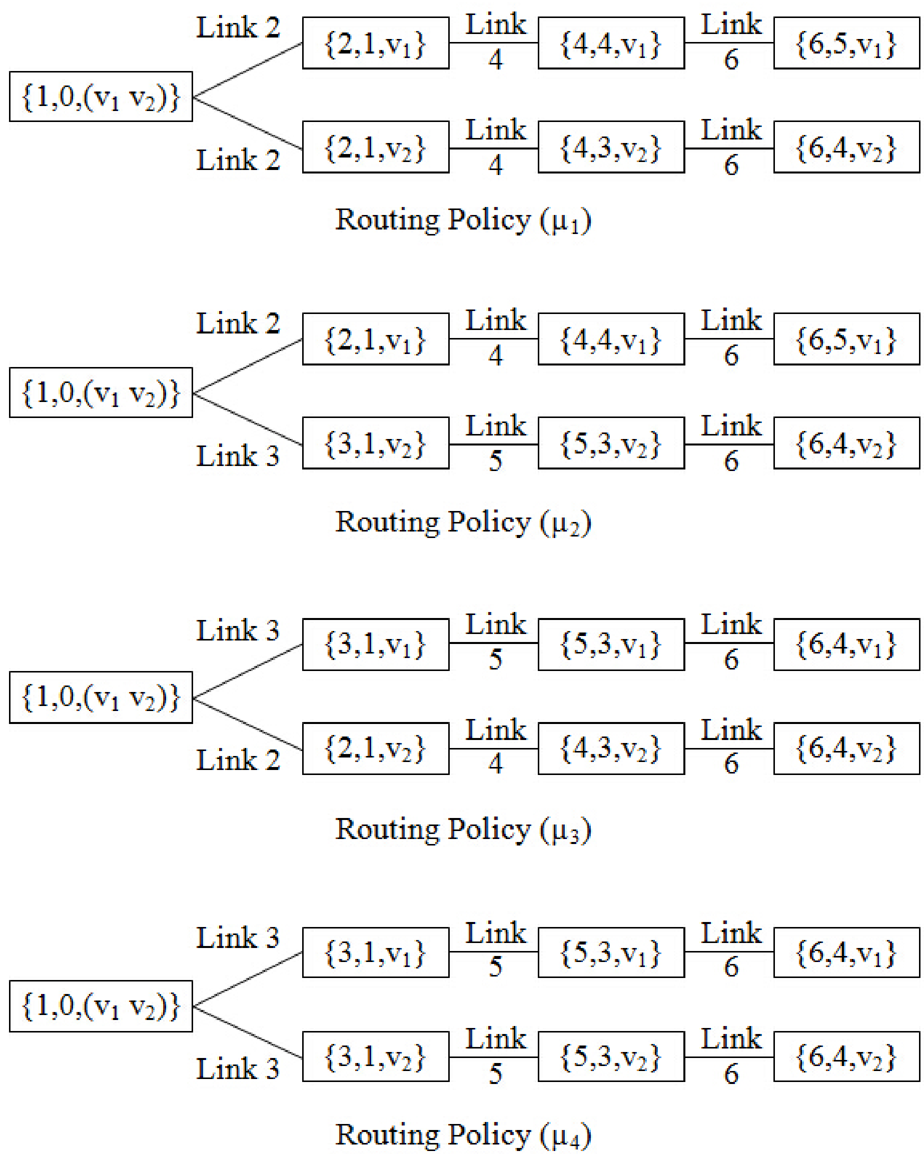


**Figure 2.3.** State trees of all routing policies in the network example

## 2.3 Model Specification and Estimation

Route choice models, which are an integral part of a traffic forecasting model for evaluating ATIS, predict the route a traveler would take when going from the origin to the destination. The model specified in this dissertation is a latent-class, latent-choice, latent-path logit model as described in the following sections.

### 2.3.1 Latent-Choice Factor

For the routing policy choice model, the choice of a routing policy is latent - it can be viewed as a plan in the traveler's mind, and only the result of the plan execution is observed, which is the realized path. The model must be estimated based on path observations and thus a latent-choice specification is needed. A routing policy is realized as a path for a given support point, which can be fully defined by the observed travel times on all random links. Figure 2.4 shows an example illustrating the latent choice concept. Policy A is realized as three paths on three days, and Policy B is also realized as another three paths on three days. Three GPS traces are observed for a trip on day one, which is a series of map-matched network links. Even though the two policies are different, they are realized as the same path on day one, which includes all the observed GPS traces on the same day. Therefore, the traveler could have chosen either policy A or B and thus both policies should be considered.

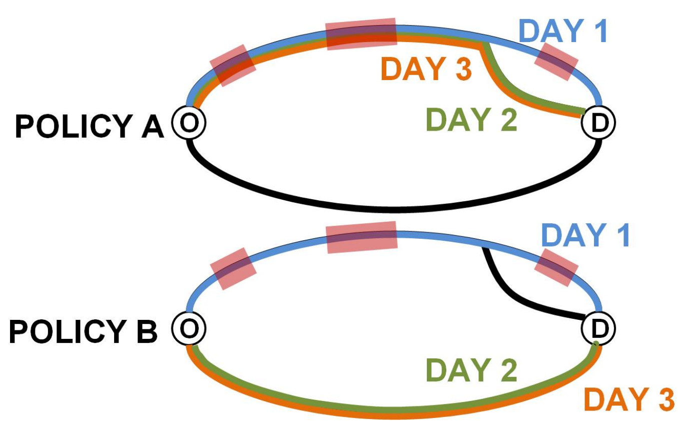


**Figure 2.4.** Latent choice example

### 2.3.2 Latent-Path Factor

Furthermore, GPS traces usually have relatively long gaps and thus the realized path is latent since it typically cannot be uniquely identified. Figure 2.5 shows an example illustrating the latent-path concept. Policy C is realized as three paths on three days, and policy D is also realized as another three paths on three days. Two GPS traces are observed for a trip on day one. Even though policy C and D are realized as different paths on day one, their overlapping part includes all the observed GPS traces on day one and thus the actual path could be either of the two paths. Therefore, the traveler could have chosen either policy C or D and both policies should be considered.

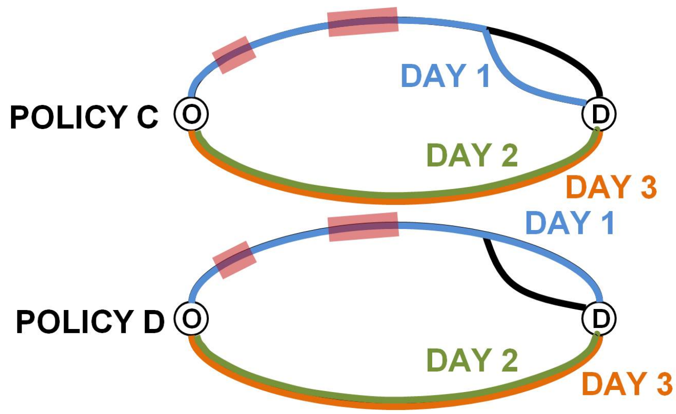


**Figure 2.5.** Latent-path example

### 2.3.3 Latent-Class Factor

It is assumed that there are two classes of individuals, policy users who follow routing policies, and path users who follow fixed paths. $\lambda$ is defined as the probability of a traveler belonging to the policy user class, and thus (1 - $\lambda$) is the probability of a traveler belonging to the path user class. The major difference between the two classes is the choice sets, where the routing policy choice set $\tilde{C}_n$ is a superset of the path choice set $C_n$, as a path is a special routing policy where routing decisions are independent of real-time information. A routing policy is realized as a path on a given day (support point), and potentially as different paths on different days. In general its attribute (e.g., travel time, # of intersections) is calculated as the expected value of the attribute for the realized paths.

### 2.3.4 A Latent-Class Routing Policy Choice Model

Equation 2.1 and 2.2 show that the choice of an alternative (path $i$ or policy $\mu$) for individual $n$ from either class is described by a Logit model with deterministic utility $V$ and parameter set ($\beta$ or $\beta'$). $V$ is a function of explanatory variables and the parameters of the variables are estimated. $PS$ (Path Size) is a deterministic correction for overlapping of paths, and $PoS$ (Policy Size) is its counterpart for routing policies, calculated as the expected path size. The utility functions and parameter sets could differ by class because different class users may have different conceptions on attributes. For example, parameters for policy users and path users differ by a scale and scale is to be estimated along with other parameters, i.e. $\beta$ = Scale * $\beta'$. $PS_i$ and $PoS_\mu$ can be calculated by Equation 2.3 and 2.4. In the literature, some studies fixed the parameters $\beta_{PS}$ and $\beta_{PoS}$ at 1 based on discrete choice theory to correct for the correlation between alternatives due to overlapping paths. Some other studies estimated them along with other parameters to capture behavioral perceptions regarding overlapping paths. Refer to [Frejinger, 2008] for a detailed discussion about Path Size.

$$P(i|C_n;\beta) = \frac{exp(V_i(\beta) + \beta_{PS} lnPS_i)}{\sum_{j \in C_n} exp(V_j(\beta) + \beta_{PS} lnPS_j)} \tag{2.1}$$

$$P(\mu|\tilde{C_n};\beta') = \frac{exp(V_\mu(\beta') + \beta_{PoS} lnPoS_\mu)}{\sum_{\theta \in \tilde{C_n}} exp(V_\theta(\beta') + \beta_{PoS} lnPoS_\theta)} \tag{2.2}$$

$$PS_i = \sum_{l \in I_i} \left(\frac{T_l}{T_i}\right) \frac{1}{M_{l,n}} \tag{2.3}$$

Where:

$I_i$ = set of links of path $i$,

$T_l$ = static and deterministic travel time of link $l$,

$T_i$ = static and deterministic travel time for path $i$,

$M_{l,n}$ = number of paths in choice set $F_n$ using link $l$.

$$PoS_\mu = \sum_{r \in R} \left( \sum_{l \in I_\mu^r} \left( \frac{T_l^r}{T_\mu^r} \right) \frac{1}{M_{l,n}^r} \right) P(r) \tag{2.4}$$

Where:

$R$ = set of support points of link travel time distribution,

$I_\mu^r$ = set of links of realized path of routing policy $\mu$ for support point $r$,

$T_l^r$ = travel time of link $l$ for support point $r$,

$T_\mu^r$ = realized travel time for routing policy $\mu$ for support point $r$,

$M_{l,n}^r$ = number of routing policies in choice set $C_n$ using link $l$ for support point $r$, and

$P(r)$ = probability of support point $r$.

Route choice observations are obtained from individual level passive GPS readings. In some applications these readings are sparse with large gaps (e.g., longer than 1 minute), and thus an individual's chosen route cannot be uniquely identified. The estimation problem is thus based on maximizing the likelihood of observing GPS traces, where a trace is an ordered set of map-matched links between an origin-destination (OD) pair where the links are generally not consecutive.

Equation 2.5 describes the likelihood of observing trace $g$ for a path user $n$ on day $r$. The first equality shows that day $r$ is irrelevant, since the individual does not adapt her choice to realized traffic conditions on any given day. The likelihood is the sum of the likelihood of observing paths from the choice set $C_n$ that contain trace $g$. $P(g|i)$ reflects the latent-path factor as it is a binary indicator which is equal to 1 if path $i$ contains trace $g$ and 0 otherwise.

$$P_{n,r}^{path}(g|\beta) = P_n^{path}(g|\beta) = \sum_{i \in C_n} P(i|C_n;\beta)P(g|i) \tag{2.5}$$

Equation 2.6 describes the likelihood of observing trace $g$ for a policy user $n$ on day $r$ as the sum of the likelihood of choosing policies from the choice set $tildeC_n$ that contain GPS trace $g$. A routing policy $\mu$ is not observable and it is viewed as chosen if the realized path $i$ on day $r$ contains trace $g$. $P(g|M(\mu, r))$ reflects both the latent-path and latent-choice factors. $M(\mu, r)$ reflects the latent-choice factor as it is the mapping of policy $\mu$ to a path $i$ on day $r$ by executing the routing policy at each decision node based on realized link travel times. Similar to $P(g|i)$ in Equation 2.5, $P(g|M(\mu, r))$ reflects the latent-path factor as it is a binary indicator which is equal to 1 if path $i$ contains trace $g$ and 0 otherwise.

$$P_{n,r}^{policy}(g|\beta) = \sum_{\mu \in \tilde{C}_n} P(\mu|\tilde{C}_n;\beta)P(g|M(\mu, r)) \tag{2.6}$$

Equation 2.7 then describes the likelihood of observing a GPS trace $g$ on day $r$ for individual $n$ as the convex combination of the likelihood from the two classes. $\lambda$ is represented by a membership function in Equation 2.8. The membership function is a logit form function in which the utility of path user probability is 0 and the utility of policy user probability, $V'$ is a linear function of an alternative specific constant ($ASC$) and the explainable variables $X'$ in Equation 2.9. The explainable variables could include, for instance, the dummy variable for long trips as travelers tend to be more strategic in long trips.Â

$$P_{n,r}(g|\beta) = \lambda P_{n,r}^{Policy}(g|\beta) + (1-\lambda)P_{n,r}^{Path}(g|\beta) \tag{2.7}$$

$$\lambda = \frac{e^(V')}{e^(V')+1} \tag{2.8}$$

$$V' = ASC + \gamma * X' \tag{2.9}$$

## 2.4 Choice Set Generation

There could be numerous alternative paths/routing policies in a general transportation network for an OD pair, but many of them may be unrealistic by being too circuitous or otherwise unsuitable. Therefore, the objective of the choice set generation is to provide a subset of realistic alternatives considered by a traveler.

Three types of choice sets are defined, path choice set, adaptive routing policy choice set, and combined routing policy choice set. Path users choose from path choice set, which is $C_n$ in Equation 2.1. The alternatives are fixed paths, and the choice set generation is the same as that in conventional path choice context.

Adaptive routing policy choice sets are generated based on the ORP Algorithm and are likely adaptive. This is the intermediate step for generating the choice sets used by policy users. The routing policy choice set generation is different from that in conventional path choice context. Most path choice set generation methods could potentially be generalized for the routing policy choice set generation. Similar to the shortest path algorithm as needed for a path choice set generation, an ORP algorithm is needed as an elementary component of any routing policy choice set generation method. In an STD network, the ORP is a routing policy that moves a traveler on a network from the origin to the destination at the least cost, which can be the least expected travel time in a simple case. Algorithm LC-CDPI(Label Correcting - Complete Dependency Perfect Information) designed in [Ding et al., 2014] is adopted, which efficiently calculates the ORP in STD large networks. The common aspect of choice set generation methods is the repeated calculations of the shortest path/ORP in slightly modified versions of the original network.

Policy users choose from combined routing policy choice set, which is $\tilde{C}_n$ in Equation 2.2. The alternatives are the combination of adaptive routing policies and the fixed paths since fixed paths are special routing policies and policy users consider both paths and policies.

### 2.4.1 Fixed Path Choice Set Generation

In path choice set generation, the static and deterministic shortest paths are first calculated based on the transformed deterministic and static network. Various methods are then utilized to generate the choice sets. Conventional choice set generation methods includes Link Elimination and Simulation. In link elimination and simulation methods, the algorithm uses only the travel time to calculate optimal path. However, when choosing a route, travelers consider a lot of other important factors such as percentage of highway, number of intersections, and number of ramps. Thus a new method, Changing Parameters, is introduced to take into account these factors. Finally, the generated fixed paths are transformed to routing policies by duplicating the same path topology over support points and calculating the new path travel times based on the specific support point.

#### 2.4.1.1 Link Elimination Method

In link elimination method, the static and deterministic shortest path is first calculated for an OD pair. Links on the shortest path are then removed from the network one at a time, and a new shortest path is generated and added to the choice set if not already included.

#### 2.4.1.2 Simulation Method

In simulation method, an independent distribution is adopted to generate the cost of every link for each time period and support point. For each independent sample of link travel times, a static and deterministic shortest path is generated and added to the choice set if different from any existing alternative. The number of samples is pre-determined based on network parameters and can be adjusted empirically. A common distribution used for generating simulation travel times is normal distribution with the original travel time as the mean. The standard deviation is usually a positive constant times the original travel time.

#### 2.4.1.3 Generalized Cost Method: Highway Bias, Intersection Delay, and Changing Functional Class Penalty

In link elimination and simulation methods, the algorithm uses only the expected travel time in the calculation of ORP. However, when choosing a route, travelers consider a lot of other important factors such as travel time standard deviation, number of left turns, number of intersections, scenery, highway bias, tolls and congestion pricing. To capture these factors, three ways are described below to generate the choice sets.

First, to capture travelers' varying attitudes toward the highway, a highway bias is introduced in the route choice. The highway bias is implemented by multiplying a certain constant to the highway link travel times. The constant is set to be positive and smaller than 1, so that generated paths include more highways as if travelers prefer highways, which is observed in real-life. In the algorithm, if a link is a highway link, then its travel time is reduced to a certain proportion. Highway bias constant could be adjusted empirically based on network parameters to reflect travelers' levels of attitudes.

Second, to capture travelers' varying attitudes toward switching on/off highway, a changing functional class penalty is introduced in the route choice. The changing functional class penalty is implemented by adding a certain constant time when going on and off highways. The constant is set to be positive, so that generated paths avoid routes frequently going on and off highways as travelers prefer to stay on the highway. Also realistically switching between local arterials and highways constantly to avoid traffic is usually more time consuming due to the delay on the ramps. In the algorithm, if the functional class of two consecutive links change from/to highways to/from local arterials, its travel time is added a constant penalty. Similar to highway bias constant, changing functional class penalty constant could be adjusted empirically.

Third, to capture travelers' varying attitudes toward the intersections, an intersection delay is introduced in the route choice. Only 4-way intersections are considered since they usually cause the most delay. The intersection delay is implemented by adding a constant to the travel times of the intersection links, which are links ending with a 4-way intersection. The constant is set to be positive, so that paths with a larger number of intersections are penalized, since travelers generally prefer routes with fewer intersections. In the algorithm, if a link is an intersection link, which has 3 outgoing links not including the opposite direction link, its travel time is added a constant penalty. Link number penalty constant could also be adjusted empirically.

### 2.4.2 Adaptive Routing Policy Choice Set Generation

Adaptive routing policy set generation uses a subset of the path choice set generation methods, except that the ORP is used in place of static and deterministic shortest path algorithm and a STD network is used instead of a static and deterministic network. The methods include the Link Elimination and the Changing Parameters method. The Simulation method, however, is not suitable for adaptive routing policy choice set generation. A routing policy generated by Simulation maps simulated network conditions, or event collections, to decisions. However, such simulated conditions may not be found in the original network. Links grouped for an event collection in a simulated network may be very different from those grouped together in the original network. Therefore, simulated routing policies may be unrealistically defined or mapped to the wrong network conditions. One extreme case is that a routing policy is attractive in simulation network but is not attractive at all in the original network.

### 2.4.3 Combined Routing Policy Choice Set Generation

Policy users consider not only routing policies but also fixed paths, so combined routing policy choice sets are the union of the adaptive routing polices and the fixed paths.

## 2.5 Choice Set Evaluation

The generated choice sets are then evaluated to verify that they contain no duplicate routing policies, and have sufficient coverage and adaptiveness. In general, they should not contain routes that no traveler would consider or exclude routes that travelers may choose.

### 2.5.1 Coverage

In an ideal condition, a generated route in the choice set for a particular OD pair should match the observed path link by link so that the choice set has included the observed path. In this case, coverage is the percentage of path observations contained in the generated choice set. To be practical, the matching criterion is relaxed because not all the observed routes will necessarily be generated link by link. To quantify this criterion, overlap is introduced as a percentage of the observed route's deterministic and static travel time shared by the generated route and the observed route. Coverage is in turn redefined as the percentage of path observations for which the algorithm has generated route which meets a particular threshold for overlap. If the matching criterion is broadened, e.g., 80% overlap threshold, then other link parameters such as link lengths, or link deterministic and static link travel times are needed to calculate the coverage.

Various choice set generation methods should be explored to keep improving the coverage of the choice sets until no more trips can be matched or no more alternatives can be generated. Matching trips' choice sets generated from different methods will be combined together to be used in the model estimation.

When checking coverage in the case study, only GPS traces are used because the actual chosen paths are not observable. Routing policies are also not observable, therefore realized paths on the observation day are used. For a given OD pair, the policies in the choice set are realized as particular paths on the day that the GPS observation is recorded. If any one of those paths sequentially contains all the GPS traces, then this OD pair is matching for 100% overlap.

In the network example in Figure 2.6, the routing policy is realized as three paths on three days and there is a set of three observed GPS traces on day one. It is shown that the realized path of the policy on day one include all the observed traces, and thus this trip is matched for 100% overlap. If the set of three GPS traces is different as shown in Figure 2.7, and the realized path only includes two observed GPS traces, then it is not matching for 100% overlap. The overlap is then calculated as as the length of matching GPS traces over the length of all GPS traces. The trip is matched if the overlap is bigger than a threshold, e.g., 90%.

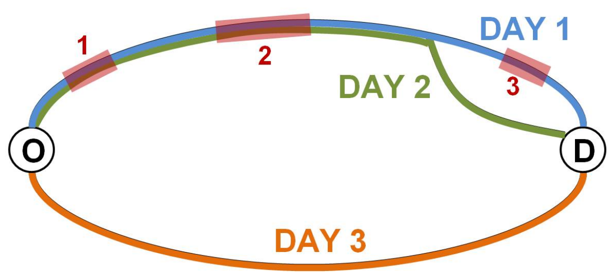


**Figure 2.6.** Example of coverage for 100% overlap

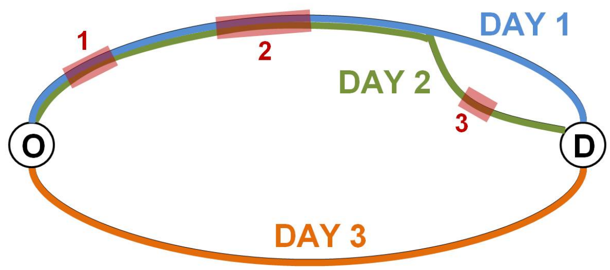


**Figure 2.7.** Example of coverage for overlap smaller than 100%

### 2.5.2 Choice Set Outlier Analyses

Outlier analyses are carried out to investigate the reasons for the unmatching trips by visualization, and coverage is improved utilizing various methods based on the analyses. Common reasons for unmatching trips include GPS errors, intermediate destinations, and downtown grid network. GPS errors are manually corrected and a trip with an intermediate destination is manually divided into two trips with the major trip being kept. Unmatching trips due to the numerous route combinations in downtown grid network are difficult to cover for 100% overlap threshold and thus overlap threshold are relaxed for such trips. In addition, routing policy choice set generation usually provides better coverage than path choice set generation, and thus coverage can be increased by including the additional trips matching in routing policy choice set generation.

### 2.5.3 Adaptiveness

Adaptiveness is defined as the fraction of days when a routing policy is realized as different paths. Adaptiveness for a given OD pair is then averaged over all routing policies in the choice set. Adaptiveness is a new criterion for evaluating routing policy choice sets, which is not used in evaluating path choice sets as paths are fixed over days. This criterion is introduced to show the advantages of routing policies over fixed paths in capturing travelers adaptive routing behavior by requiring a routing policy to be realized as different paths on different days.

In the network example in Figure 2.8, the routing policy is realized as two paths on three days. The adaptiveness is then 0.667 (2 divided by 3).

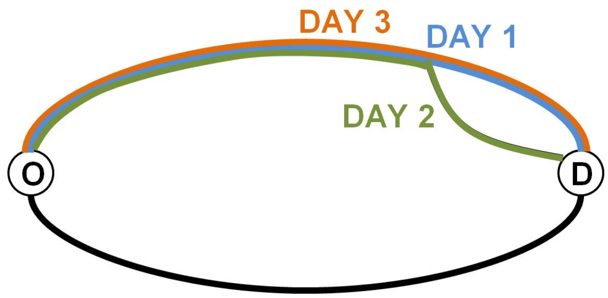


**Figure 2.8.** Example of adaptiveness

# CHAPTER 3

# OPTIMAL ROUTING POLICY AND ALGORITHM LC-CDPI

## 3.1 The Optimality Condition

An optimal routing policy (ORP) algorithm is a building block of any routing policy choice set generation method. In an STD network, the ORP is a routing policy that moves a traveler on a network from the origin to the destination at the least cost, which can be the least expected travel time in a simple case. Define $e_\mu(x)$ as the expected travel time to the destination node $d$ with the routing policy $\mu$ and the initial state $x$. Let $\tau_{jk,t}^{EV}$ be the travel time on link $(j,k)$ at time $t$ given event collection $EV$, which is a single value due to the assumption of POI (all link travel times at time $t$ are known). $A(j)$ is the set of downstream nodes of node $j$, and $B(j)$ is the set of upstream nodes of node $j$. $Pr(EV'|EV)$ is the probability of $EV'$ conditional on $EV$. $EV(t)$ is the set of all possible event collections at time $t$.

$e_{\mu_\star}(x)$ and $\mu_\star$ are optimal solutions if and only if they satisfy the following equations:

$$e_{\mu_\star}(i,j,t,EV) = min_{k\in A(j)}\{\sum_{EV'\in EV(t+\tau_{ij,t}^{EV}+T(i,j,k,t))} e_{\mu_\star}(j,k,t+\tau_{ij,t}^{EV}+T(i,j,k,t),EV') \times Pr(EV'|EV) + T(i,j,k,t) + \tau_{ij,t}^{EV}\} \tag{3.1}$$

$$\mu_{\star}(i,j,t,EV) = argmin_{k\in A(j)}\{ \sum_{EV'\in EV(t+\tau^{EV}_{ij,t}+T(i,j,k,t))} e_{\mu_{\star}}(j,k,t+\tau^{EV}_{ij,t}+T(i,j,k,t),EV') \times Pr(EV'|EV) + T(i,j,k,t) + \tau^{EV}_{ij,t}\} \tag{3.2}$$

with boundary conditions: $e_{\mu_{\star}}(d',d,t,EV) = \tau^{EV}_{d'd,t}$, $\forall t \in T$, $\forall d' \in B(d)$, $\forall EV$, and $e_{\mu_{\star}}(i,j,t,EV) = e_{\mu_{\star}}(i,j,K-1,EV), \forall (i,j) \in A, \forall t > K-1, \forall EV$.

## 3.2 Algorithm LC-CDPI

Algorithm LC-CDPI(Label Correcting - Complete Dependency Perfect Information) is designed based on Algorithm DOT-SPI in [Ding & Gao, 2012], which finds the least expected travel times from all nodes at all departure times with all possible current-information to a certain destination node $d$. A queue, $SE$ (Scan-Eligible), is maintained of all links that might violate their optimality conditions. If $SE$ is empty, clearly an optimal solution is obtained. Otherwise, this queue is examined to select a link, say $(j,k)$, checking if its incoming links violate the optimality condition. $(j,k)$ is removed from $SE$, and if the incoming link $(i,j)$ violates its optimality condition, it is used to update the cost label of $(i,j)$. Increasing cost label of $(i,j)$ maintains the optimality condition for all incoming links at node $i$. But a decrease in the cost label of $(i,j)$ decreases the costs of all links incoming at node $i$ and some of these links might violate the optimality condition. Therefore, if the cost label of $(i,j)$ decreases, $(i,j)$ must be added to the set $SE$. Whenever a link is added to $SE$, all incoming links of the added link are checked for optimality. POI is assumed so that Bellman's Principle holds. Proof can be found in [Gao & Huang, 2012]. The algorithm finds the optimal routing policies when it terminates.

The statement is as follows.

Step 1: Initialization

1.1: Construct EV(t), $\forall$ t $\in$ T

Call $GenerateEventCollection$ (see the statement below)

1.2: $e_{\mu_\star}$(i, j, t, EV) = $\infty$, $\mu_\star$(i,j,t,EV) = - 1

$\forall$ (i,j) $\in$ A - {d', d}, $\forall$ t $\in$ T, $\forall$ EV $\in$ EV(t)

$e_{\mu_\star}$(d', d, t, EV) = $\tau^{EV}_{d'd,t}$, $\mu_\star$(d', d, t, EV) = d

$\forall$ d' $\in$ B(d)

1.4: Create the scan-eligible list SE, and insert all destination links (d', d)

Step 2: Choose Link from SE

2.1: If the SE list is empty, stop. Otherwise, select the first link from the SE list, link (j,k).

Step 3: Update the Link Labels

3.1: For each i $\in$ B(j) (incoming links of node j)

If i = k, continue. (forbid U-Turn)

For each t $\in$ T

For each EV $\in$ EV(t)

$$temp = \tau^{EV}_{ij,t} + \sum_{EV' \in EV(t+\tau^{EV}_{ij,t}+T(i,j,k,t))} e_{\mu_\star}(j, k, t + \tau^{EV}_{ij,t} + T(i,j,k,t), EV') \times \Pr(EV' \mid EV) + T(i,j,k,t)$$

If $temp$ < $e_{\mu_\star}$(i, j, t, EV) then

$e_{\mu_\star}$(i, j, t, EV) = $temp$

$\mu_\star$(i, j, t, EV) = k

flag = 1

End if

End for

End t

If flag = 1 and (i,j) not in SE, insert (i,j) in SE

$GenerateEventCollection$

D = $\{\{\upsilon_1, \ldots, \upsilon_R\}\}$

For t = 0 to K - 1

For each arc (j,k) $\in$ A

For each disjoint set S $\in$ D

w = number of distinct values among $\tau^r_{jk,t}$, $\forall$ r | $\upsilon_r \in$ S;

Divide S into disjoint sets $S'_1$, $S'_2$, ..., $S'_w$,

such that $\tau^r_{jk,t}$ is constant $\forall$ r | $\upsilon_r \in S'_i$, i = 1,. . . ,w and $\bigcup_i S'_i$ = S;

D' $\longleftarrow$ D' $\backslash$\{S\} $\bigcup$ \{$S'_1$,$S'_2$,...,$S'_w$\};

Next S

D' $\longleftarrow$ D'

Next (j,k)

EV(t) $\longleftarrow$ $D$;

Next t

Algorithm LC-CDPI implements the following three major changes to the original label-setting Algorithm DOT-SPI [Gao & Chabini, 2006b] to make it applicable to large networks with improvements in realistic modeling features, memory, and running time.

### 3.2.1 Piece-wise Linear Travel Time Representation

The dynamic link travel times are represented by piece-wise linear functions instead of discrete values. In a typical application of the ORP problem, the study period (e.g., 6-9 am) is divided into smaller time periods each having a length of 5 to 15 minutes. The discrete joint link travel time distribution in Algorithm DOT-SPI is represented by a three-dimensional matrix with the dimension of "number of time periods $\times$ number of support points $\times$ number of links." For real networks with hundreds of thousands of links, it is nearly impossible to store the distribution in the computer memory for a reasonably long study period. A piece-wise linear function, on the other hand, can store only the breaking points. Between the two breaking points, the travel time is derived by interpolation. Since the breaking points are generally sparsely distributed over time, the number of breaking points is much smaller than the number of time periods. For example, a study period of 3 hours with 5-minute time periods has 36 time periods, and if the piece-wise linear function is defined by only 9 points, which in general is adequate to describe the profile of travel time over the 3-hour horizon, then the memory required is reduced by 4 times.

### 3.2.2 Label-correcting

Algorithm DOT-SPI is a label-setting algorithm that goes through all time periods in decreasing order of time, as the name DOT (decreasing order of time) suggests. It requires link travel times to be positive integers in units of time period length such that the time-expanded network is acyclic in the time dimension and label setting can be applied. As a result, the time period length needs to be smaller than the shortest link travel time in the network. This constraint usually results in a time period length in seconds, and therefore a very large number of time periods. This large number of periods leads to very long running time (in addition to the excessive memory usage, which could be resolved by the previously introduced piece-wise linear function). A label-correcting algorithm is designed that only requires the travel times to be non-negative, and thus the time period length can be greater than link travel times. As such, the number of time periods can be significantly reduced, resulting in the reduction of running time.

### 3.2.3 Turn-based

The behaviors of waiting at nodes, making U-turns, and looping are infrequent in a real-life network since travelers want to avoid loss in travel time, but such problems are observed when applying Algorithm DOT-SPI in several real-life networks. Algorithm DOT-SPI cannot avoid such problems because it is node-based and turning penalties cannot be added. The improved Algorithm LC-CDPI is a link-based algorithm so turning penalties can be applied to make waiting, U-turns, and looping less desirable.

## 3.3 Complexity Analysis

Incorporating the changes stated above, Algorithm LC-CDPI consists of two major steps. The first step is to construct event collections, and it takes running time $O(mKR\ln R)$ and $\Omega(mKR)$. The second step is the main loop of the label correcting procedure, which updates labels at all links, all time periods and for all event collections until the optimality conditions are satisfied, and it takes running time $O(m^3K^2R)$. Notably, the worst case complexity of the algorithm is thus $O(mKR\ln R + m^3K^2R)$.

Practically, however, the computational tests in [Ding & Gao, 2012] show that the running time is linear relative to most of the size factors, e.g. number of time periods, number of support points, number of links, while the worst case complexity indicates faster growth. This result suggests the adverse worst case complexity rarely occurs in practice and the algorithm yields good average case performance.

## 3.4 Computational Tests: Pioneer Valley Network

### 3.4.1 Data

#### 3.4.1.1 Network

A case study to test the algorithm is conducted in a real-life large network, Pioneer Valley, which encompasses 43 cities and towns in the Connecticut River Valley in western Massachusetts. Framed on the west by the Berkshires and on the east by the central uplands, this area has an estimated 608,000 people living in the nearly 1,200-square-mile region. The fourth largest metropolitan area in New England is also located in this area.

Underlying network data is obtained from a GIS database provided by Pioneer Valley Planning Committee (PVPC). The network includes all major arterials and highways in Hampshire and Hampden Counties, consisted of 14,417 nodes, 28,731 links, and 512 Traffic Analysis Zones (TAZs). The 2000 U.S. Census data is used for the base year analysis and the 2010 demand forecasts are used due to its proximity to current time.

#### 3.4.1.2 Link Travel Time Distribution

Using the calibrated data from PVPC, 10 runs of DTA are applied to generate 10 day travel time distributions (10 support points) in the original network (base case) and nine (9) slightly modified networks (incident cases). The study duration is from 3pm to 9pm simulating evening traffic. The incident duration is from 4pm to 7pm.

Origin-Destination (OD) demand time file is generated and used for the normal scenario and 9 other different scenarios (capacity reduction or road closure at different locations). The base case is created by running DTA in the original network, which is the normal scenario and serves as a control for incident cases. In incident cases, it is assumed that travel demand is unchanged from the base case and that travelers follow their normal patterns before the incidents since unexpected events come with little or no advanced warning. Incident cases are created by running DTA in networks with disabled links during the incident duration, which simulates road closures from incidents, eg. a high volume due to daily commuter traffic, a vehicle crash, a work zone. In each run of DTA, links at different locations are disabled and thus the network is modified accordingly. The locations of the incidents are chosen because they represent vital links in the network, which would severely impede traffic in Massachusetts and surrounding states if closed. DTA accounts for time-dependent influence of the incidents, and the DTA time period is set as 10 minutes. DTA generally requires OD matrices for each period as input, and the outputs are dynamic link flows, travel times, and other relative variables for every time period. The DTA results present network conditions before, during, and after the incidents. The study duration is six (6) hours and thus there are 36 data points for each link. Piece-wise linear link travel time function for each stochastic link/support point is generated using DTA results, where the breaking points are travel times at each DTA time period. The number of stochastic links are reduced to save memory (RAM). If the variance of link travel times over support points and time periods for a link is less than 0.05, then the link is made deterministic.

Figure 3.1 and Figure 3.2 show Pioneer Valley Network and the locations of the incidents by colored lines.

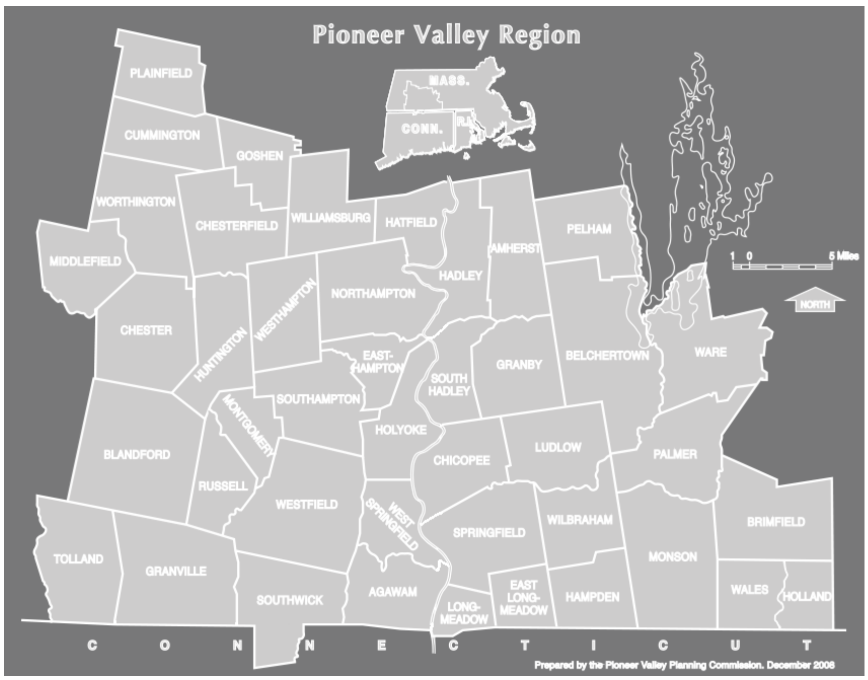


**Figure 3.1.** Map of Pioneer Valley towns

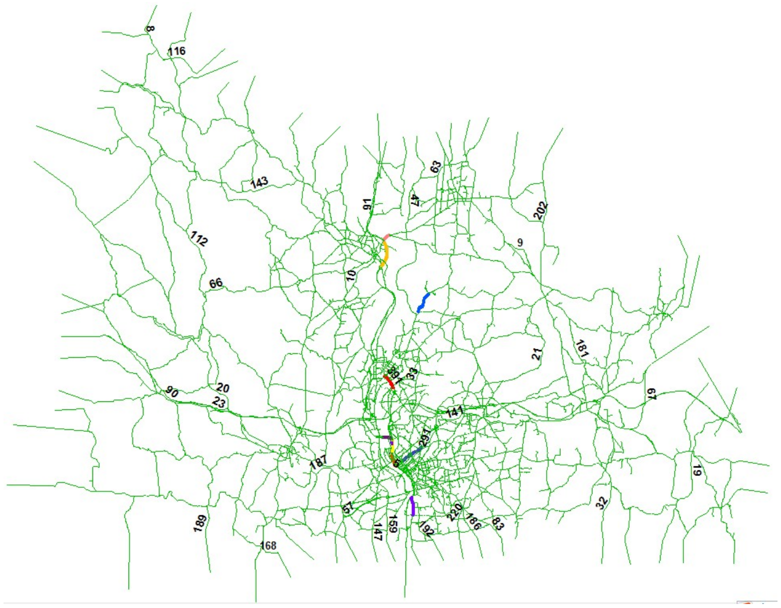


**Figure 3.2.** Road network of Pioneer Valley (designed incident locations shown in colored links)

#### 3.4.1.3 OD Pairs

Around 1000 OD pairs are sampled to test the algorithm. OD pairs are chosen from all-to-all shortest path matrix to get a wide spread of OD pairs. Also they are chosen because they represent real life origins and destinations. All-to-all shortest path matrix is generated by the shortest path algorithm in the simulation software. Figure 3.3 shows fewer number of the chosen OD pairs for demonstration.

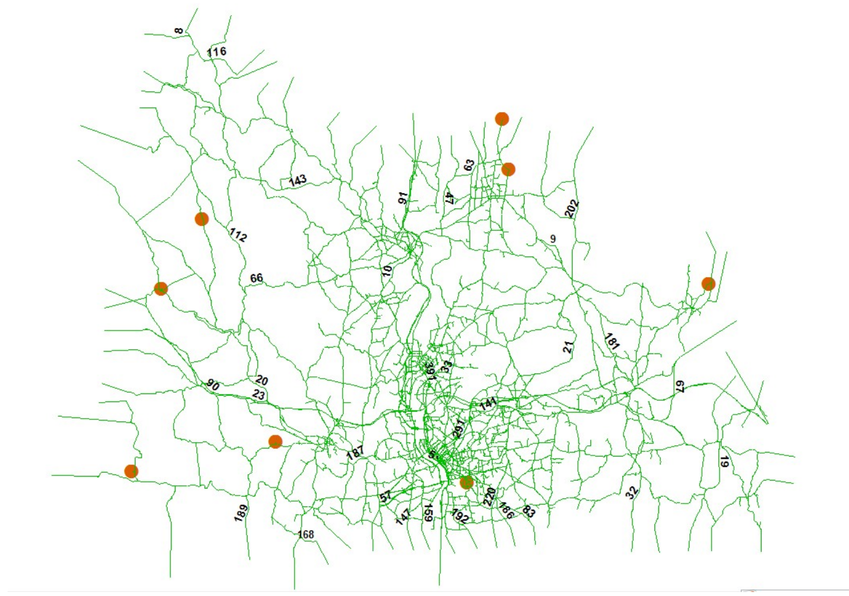


**Figure 3.3.** Example of the OD pair locations

### 3.4.2 Computational Tests

As mentioned in Section 3.4.1.3, 1000 OD pairs are tested to compare Paths and Policies, and to analyze the running time. As is proved theoretically, policies are better than paths. Illustrative examples of trips comparing policies and paths are demonstrated and all results of the running time are in seconds (s). Test results also show the running time is linear in a lot of the factors while the worst case complexity indicates faster growth, as demonstrated in Section 3.3. This means the adverse worst case complexity barely occurs in practice and the algorithm has good average case performance.

#### 3.4.2.1 Paths vs. Policies

Important to this dissertation is the comparison of Paths and Policies. It has been proven in hypothetical simplified networks that Policies are better than Paths, which are fixed over days. Because the existing Optimal Path algorithm is very time consuming, a heuristic is designed to calculate Optimal Paths. In order to perform this comparison, five (5) OD Pairs have been manually created and locations of a high traffic delay or an incident have been manually input to evaluate the difference in results between a Path and a Policy. Departure time from the origin is set to be 4:10PM, 10 minutes after the incident. Three (3) different cases with an incident occurrence have been chosen to present the difference between Path and Policy. The results of this analysis are shown below.

*Case 1: Interstate 91 Southbound, Northampton* Case 1 examines an OD pair traveling from Ludlow Service Plaza located on Interstate 90 Eastbound northerly to Exit 21 in Northampton. In this case an incident has occurred on Interstate 91 Northbound from the Exit 18 to Exit 19 causing heavy delays. This incident results in a change in normal path taking Exit 18 and utilizing Route 5 as shown in Figure 3.4. The issue with this change in Path is the result in a permanent change regardless if there is an incident. The end result is not an optimal route. When performing a Policy to this same case the route choice will differ based on up-to-date information provided to the driver. When an incident occurs, the driver will utilize Exit 18 via Route 5 to bypass the traffic, however when an incident is not present the driver will maintain the optimal route of Interstate 91 as shown in Figure 3.5.

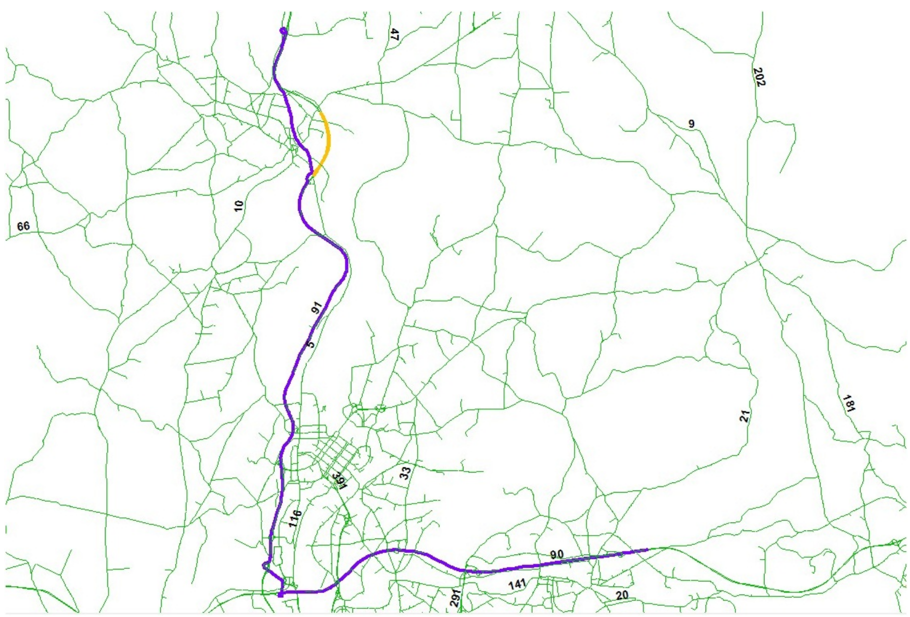


**Figure 3.4.** Case 1: Path

**Figure 3.5.** Case 1: Policy

Case 1 illustrates the advantage of a Policy versus a Path where the introduction of real-time data allows the driver to vary the route choice providing a more optimal route as opposed to a fixed route which are only optimal during an incident occurrence.

*Case 2: Calvin Coolidge Bridge, Hadley* Case 2 evaluates a very well documented traffic location being the Calvin Coolidge Bridge Spanning between Hadley and Northampton across the Connecticut River. This bridge being a main crossing of the river frequently experiences delays. The OD Pair in this case will be traveling from Conway Road in Ashfield, Ma southerly to the Center of Ware, Ma. Due to the incident at the Coolidge Bridge the Path has been redirected to continue Southbound on Interstate 91 and travel East on Interstate 90. This change in route results in an extremely long detour as opposed to the normal Optimal Path of utilizing Route 9 through the towns Amherst and Belchertown. Due to this fixed change in Path the driver will consistently detour a longer route. The resultant Path is shown in Figure 3.6. When exercising a Policy, the driver may once again decide whether to take the longer indirect route or utilize the shorter Route 9 which will optimize travel times based on given information about the incident location. The optimal routes based on Policy are illustrated in Figure 3.7.


**Figure 3.6.** Case 2: Path


**Figure 3.7.** Case 2: Policy

This Case further demonstrates the greater efficiency of Policy versus Path as previously shown in Case 1 to provide a more consistent optimal route.

*Case 3: Interstate 91 Southbound, West Springfield* As in the first two cases, this case studies an OD Pair that is redirected based on a manual input incident. In this case an incident is presented on Interstate 91 Southbound at the "Bow tie" ramps in West Springfield, Ma. In this case The OD Pair travels from Route 143 in Williamsburg, Ma to Island Pond Road in Springfield, Ma. Due to the location of the incident the projected Path is required to completely bypass a section of Interstate 91 permanently and instead travel Route 5 southbound to the Memorial Bridge. As shown in Figure 3.8 this new path will direct drivers through a congested rotary at the Memorial Bridge and additional signalized intersections along Route 5. This new Path will only be optimal during the incident condition. As illustrated in Figure 3.9, a Policy will once again allow a driver to make a route choice decision based on real-time information and thus fully optimizing the route. This case reinforces the position that Policy will result in a more optimal path on a daily basis.

**Figure 3.8.** Case 3: Path

**Figure 3.9.** Case 3: Policy

As these results have been presented, the large difference between a Path and a Policy become readily apparent. A Path fails to optimize route choice on a more daily basis due to the limitation of strictly picking a Path choice that is fixed; however, as this analysis has shown, a Policy will vary the route choice to continuously optimize the travel time. This comparison helps to reinforce the study of Policy for this dissertation.

#### 3.4.2.2 Number of Time Periods vs. Running Time

In the analysis, 15 different number of time periods are tested against the running time. All other parameters remain the same, ie. 10 support points and 36 breaking points. The results show that the running time increases linearly with the increase of number of time periods. Figure 3.10 illustrates the results for four (4) OD pairs.

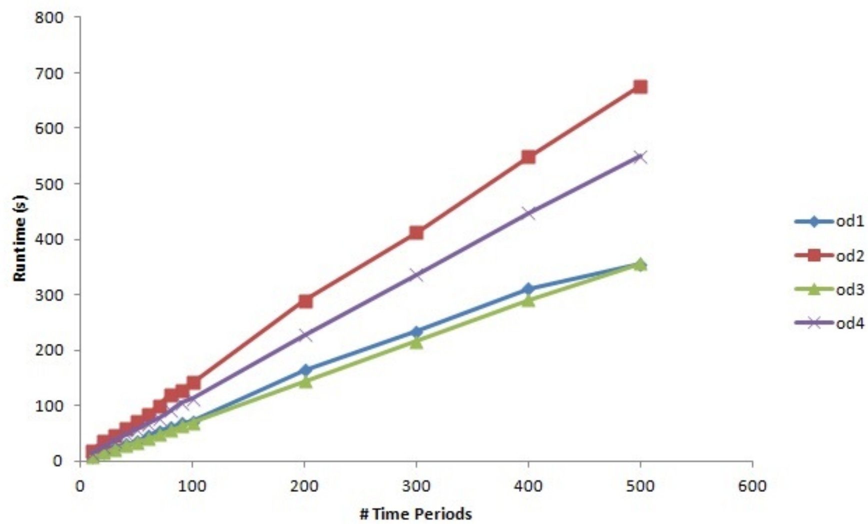


**Figure 3.10.** Number of Time Periods vs. Running Time in Pioneer Valley Network

#### 3.4.2.3 Number of Support Points vs. Running Time

The effect of the number of support points on the running time is important to understand for efficiency. In this analysis the trend of the running time change is tested against 10 different number of support points, from 1 to 10 with an increment of 1. All other parameters remain the same, ie. 12 time periods and 36 breaking points. The results show that the running time increases linearly with the increase of number of support points. Figure 3.11 illustrates the results for four (4) OD pairs.

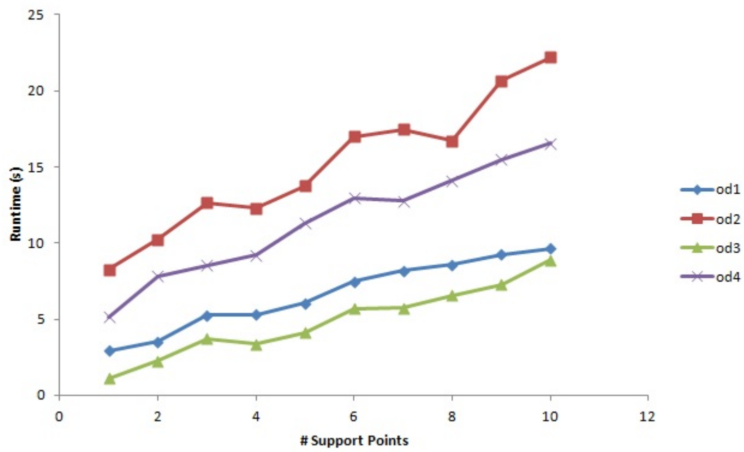


**Figure 3.11.** Number of support points vs. running time in Pioneer Valley network

#### 3.4.2.4 Number of Piece-wise Breaking Points vs. Running Time

Another important comparison is the effect of number of piece-wise breaking points on running time. The relationship is more intriguing than other parameters because number of time periods is a covariate of number of breaking points affecting the running time. In the tests, the number of breaking points is increased from 6 to 36 in periods of 6. Each set of breaking points are tested for every time period from 1 through 42. From these tests, a chart is plotted to determine the overall trend of running time with number of breaking points. The results of the analysis indicate two separate trends with number of time periods. When the number of time periods is small (less than 14 in this case) the run time steadily increases with increasing number of breaking points and remains constant after 12 breaking points or more. Figure 3.12 below illustrates the results.

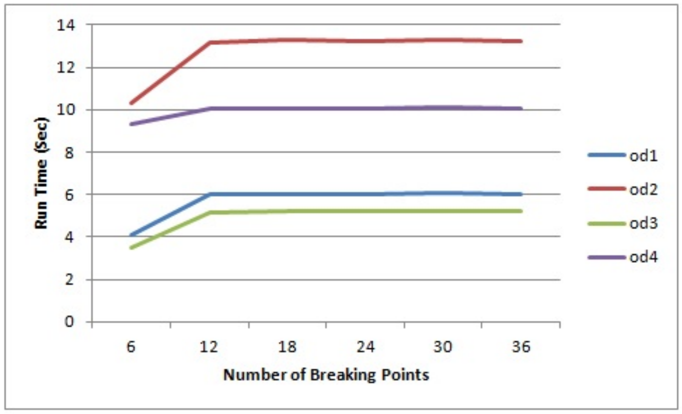


**Figure 3.12.** Number of breaking points vs. running rime in Pioneer Valley network (time periods<14)

The second trend which is a little less intuitive which occurs under large number of time periods (greater than 14 in this case). This trend indicates that, similar to the previous trend, the running time steadily increases with increased number of breaking points and remains constant after 12 breaking points. However for greater than 24 breaking points the running time steadily declines and finally remains constant at 30 breaking points. The results are shown in Figure 3.13

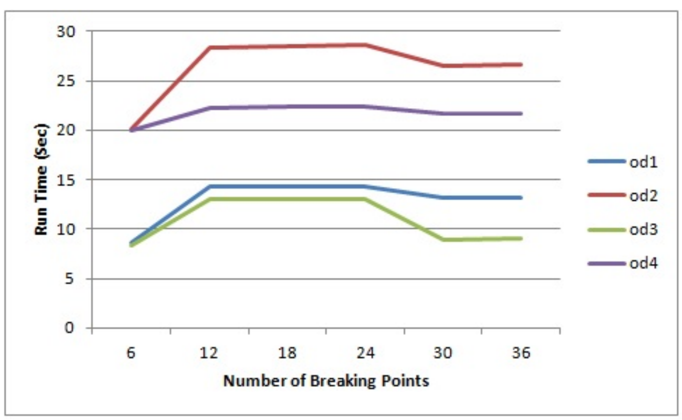


**Figure 3.13.** Number of breaking points vs. running time in Pioneer Valley network (time periods>14)

The reasoning for this trend in the running time is there are two contradictory forces determining the running time when increasing the number of breaking points. First, to search for the corresponding breaking point for a certain time stamp, the algorithm linearly goes through each breaking point starting from first one until the corresponding breaking point is found. So the increase of number breaking points increases the searching range and in turn increases the running time. Second, the time periods after the last breaking point will request the algorithm to go through all the breaking points. So the more the time periods are after the last breaking point, the higher the running time is. When increasing the breaking points, there will be more time periods becoming smaller than the last breaking point and hence it decreases the searching range and saves the running time.

## 3.5 Computational Tests: Random Networks

### 3.5.1 Data

The algorithm is developed to meet the growing demand of solving adaptive routing problems in large-scale STD real-life networks. Meanwhile, it can also be applied to solve adaptive routing problems efficiently in networks of any size. In order to fully understand the applicability and the running time, the algorithm is also tested in random networks of varying sizes. A random network is a network with randomly generated topology. For the random networks used in the tests, the number of links is set as three (3) times the number of nodes. The maximum number of incoming links for each node and the maximum number of outgoing links for each node are set as five (5). All the link travel times (on all links and for all time periods) are random, i.e., sampled from the truncated multivariate normal distribution. The expected value of the truncated multivariate normal distribution for link travel time is set as five (5) minutes. The standard deviation is set as two (2) minutes. The correlation coefficient is set as 0.5. The probabilities are set as equal for every support point. The destination is set as node $N-1$, where $N$ is the number of nodes.

### 3.5.2 Computational Tests

Similar tests mentioned in Section 3.4.2 are conducted on random networks to determine if the trends shown previously will remain consistent. The running time is tested on 66 random network specifications of different parameter combinations. Each specification includes ten (10) different instances and the running time results for each specification are the average of the ten (10) instances. Similar to Pioneer Valley network, the parameters examined include number of time periods, number of support points, number of breaking points. In addition, the parameter of number of nodes is also examined, which cannot be tested in the fixed size network, Pioneer Valley. One (1) OD pair is shown here in each network for demonstration and all results of the running time are in seconds (s).

The algorithm works efficiently in random networks. Comparisons between Pioneer Valley network and random networks show that network size is a dominant factor of the running time. The running time increases with network size. Test results also further reinforce the conclusions from Pioneer Valley network that the running time is linear in a lot of the factors as demonstrated in the following sections.

#### 3.5.2.1 Number of Time Periods vs. Running Time

Four (4) random networks specifications of ten (10) random instances each are tested at periods of number of time periods from 2 to 20 with an increment of 2. The networks chosen are of varying sizes and all results show similar trends, in which the running time increases quite linearly with the number of time periods. The test results tests are illustrated in Figure 3.14.

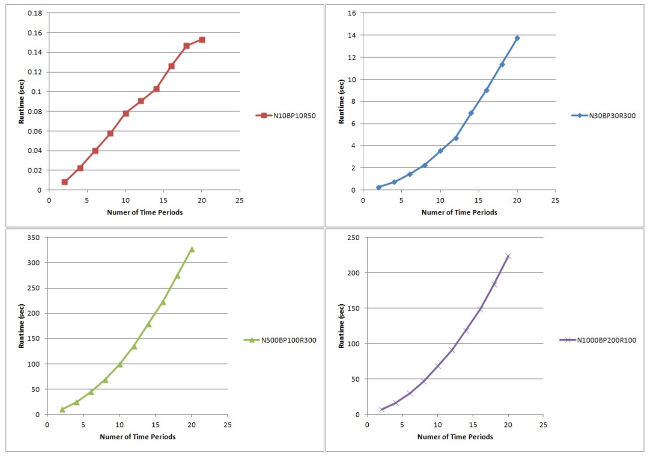


**Figure 3.14.** Number of time periods vs. running time in random networks

#### 3.5.2.2 Number of Support Points vs. Running Time

Under small networks the tests were conducted on running time against number of support points at periods of 50, 100, and 300. Each of these periods were tested against various combinations of number of nodes (10, 20, 30) and number of breaking points (10, 20, 30). For all the tests below, the number of time periods is maintained at 12, which is the same as that used in Pioneer Valley network corresponding tests. The results of the tests are illustrated below in Figure 3.15.

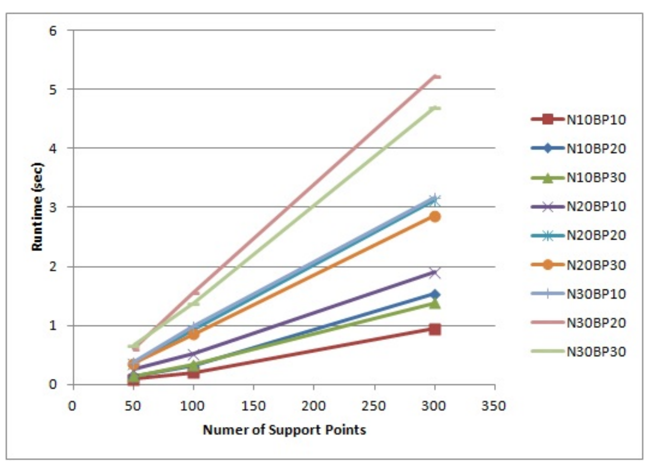


**Figure 3.15.** Number of support points vs. running time in small random network

The results indicate similar to the pioneer valley network that the run time increases linearly with number of support points. To further verify this trend, a further analysis on large random networks are performed. Tests include periods of number of support points at 50, 100, and 300. Each period is tested against various combinations of number of nodes (50, 100, 250, 500, 1000) and number of breaking points (25, 50, 100). The results of this analysis are shown in Figure 3.16.

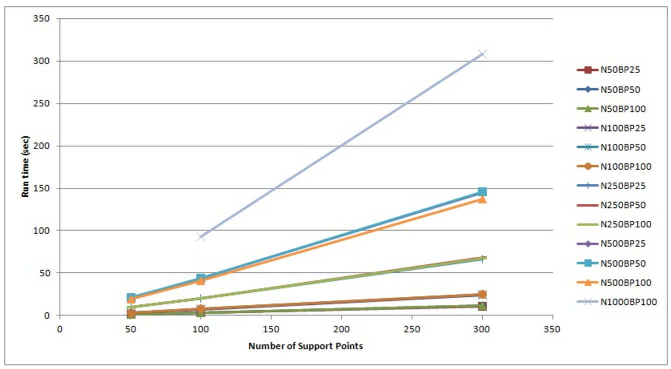


**Figure 3.16.** Number of support points vs. running time in large random network

Once again the results further support the findings that the run time will increase linearly with number of support points regardless of the size of the network.

#### 3.5.2.3 Number of Piece-wise Breaking Points vs. Running Time

Under a small network the tests where conducted on running time against number of breaking points at periods of 10, 20, and 30. Each of these periods were tested against various combinations of number of nodes (10, 20, 30) and number of support points(50, 100, 300). The results of the tests are illustrated below in Figure 3.17.

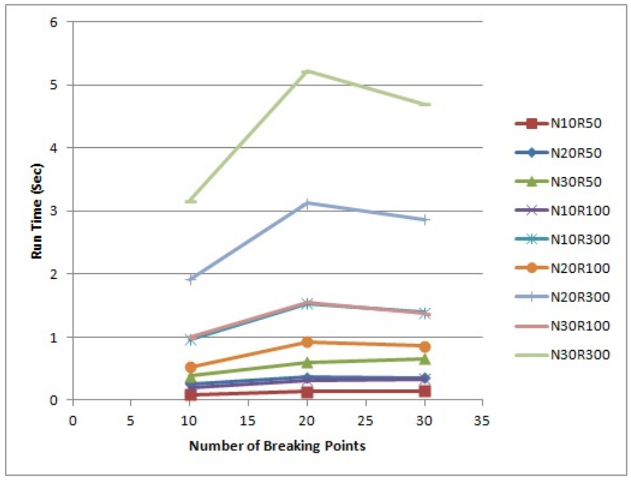


**Figure 3.17.** Number of breaking points vs. running time in small random network

The results indicate under a small network for all combinations of number of nodes and support points the Run Time will increase from 10 through 20 and will level off or slightly decrease beyond 20 breaking points. This type of trend is consistent with the previous findings in the Pioneer Valley network.

Similar testing was then conducted on the large random networks to further substantiate these findings. Testing included varying breaking points in periods of 25, 50, and 100. Each set of breaking points is tested similar to the small network against various combinations of number of nodes (50, 100, 250) and support points (50, 100, 300). The results in this testing are shown in Figure 3.18

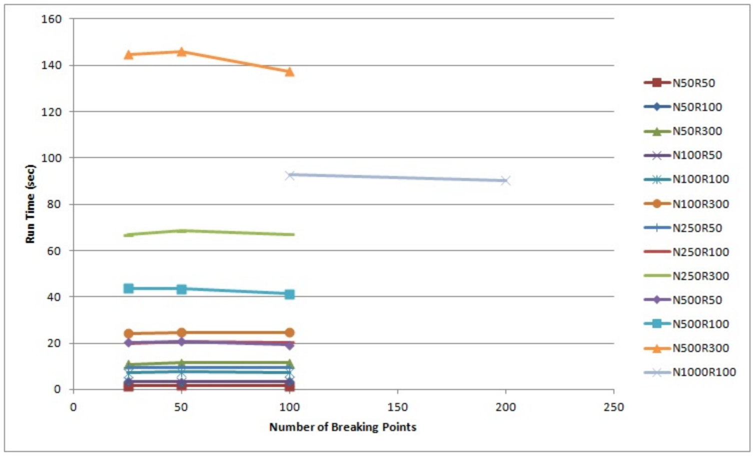


**Figure 3.18.** Number of breaking points vs. running time in large random network

The results as illustrated further corroborate the the previous findings. The Run Time slightly increases from 25 through 50 breaking points and then levels or slightly decreases beyond 50 breaking points.

The overall findings of this testing indicate that as the number of breaking points increases preceding the time periods range, the running time will continue to increase. As the number of breaking points overlaps the time periods, the Run Time begins to level out.

#### 3.5.2.4 Number of Nodes vs. Running Time

Small random networks are tested on running time against number of nodes (or network size) at periods of 10, 20, and 30 at varying combinations of breaking points (10, 20, 30) and support points (50, 100, 300). Results of this testing is shown in Figure 3.19.

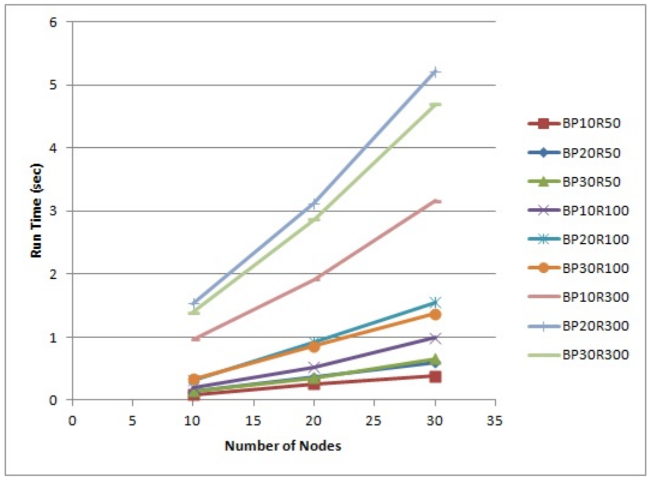


**Figure 3.19.** Number of nodes vs. running time in small random network

These results indicate that run time increases linearly with increasing number of nodes. Similar testing is performed on large random networks to determine if the same trend will hold. The number of nodes are in periods of 50, 100, 250 and 500 at combinations of breaking points (25, 50, 100) and support points (50, 100, 300). The results of this analysis are illustrated in Figure 3.20.

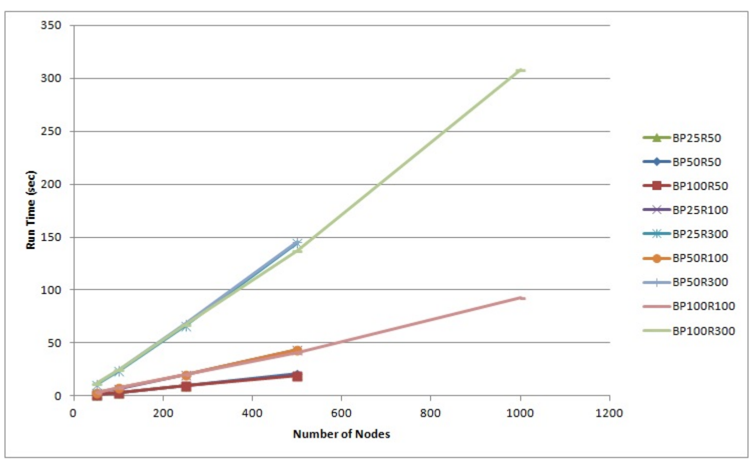


**Figure 3.20.** Number of nodes vs. running time in large random network

The results are consistent with the previous findings that run time increases linearly with number of nodes in the network. Therefore, the number of nodes will be a factor in run time regardless of the network.

## 3.6 Improvement on Optimal Routing Policy Efficiency

### 3.6.1 Base Case

Although the running time of the original algorithm LC-CDPI is manageable, running time increases with decreasing time period length or increasing number of time periods. Study duration is usually long in large-scale real-life networks with large number of time periods. Therefore, extensive efforts are made in improving the optimal routing policy efficiency and the original algorithm LC-CDPI is referred to as the base case.

### 3.6.2 Large File Support

Algorithm LC-CDPI is an all-to-one algorithm that finds solutions from all links to the destination. Thus only one calculation is needed for trips with a same destination. Such trips are combined together after sorting the destinations in the input file. With all trips in one file, the output file size can be very large. Therefore, Software with the feature of large file support is utilized so large output files can be processed. Unnecessary reading and writing from/to hard disk is also avoided.

### 3.6.3 Two-queue Data Structure

The data structure of the original algorithm LC-CDPI is a first in first out (FIFO) queue, which is proved to be inefficient in the literature for a label correcting algorithm. Thus new data structures including Deque ( [Pape, 1974]) and two-queue ( [Pallottino, 1984]) are explored and it is found that two-queue provides smaller running time.

Two-queue data structure meets the interest of updating as quickly as possible by inserting unlabeled links (link cost is infinity) in one queue and those which have been labeled and scanned in another queue. two-queue can be seen as a queue $Q'$ connects to another queue $Q$” if link $(i, j) \notin Q$, insert $(i, j)$ at the end of $Q$ if it is unlabeled or at the connection point if it has been previously labeled and scanned. Figure 3.21 illustrates the two-queue data structure for the label correcting algorithm.

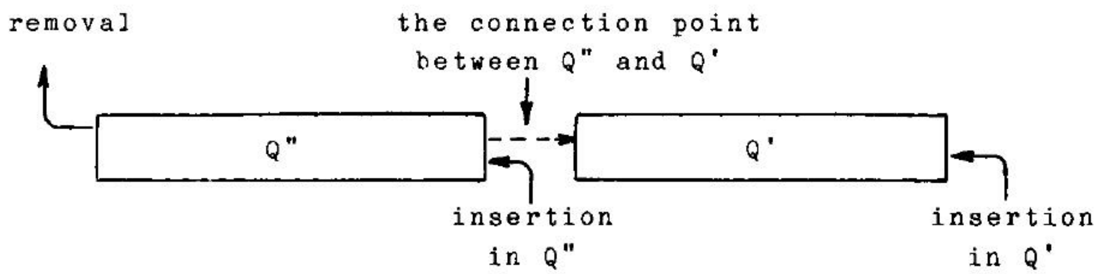


**Figure 3.21.** Two-queue data structure for the label correcting algorithm

### 3.6.4 Appropriate Time Period Length

As mentioned in 3.2.2, the new algorithm allows flexible time period length so that less time periods are stored and efficiency is improved. However, if the time period length is too long as in the base case, then the network is equivalent to a static one for each trip as no trip can cross two time periods. An appropriate time period length should be decided based on link travel time histogram so that it is smaller than most link travel times, e.g. 10th percentile the link travel times). Computer memory and running time should also be considered when choosing time period length. Therefore, there is a trade-off between computational efficiency and solution quality and the accuracy should be acceptable after making approximations.

### 3.6.5 Customized Study Duration

In the base case, a long study duration which can cover all the trips is applied to every single trip. However, the length of each trip is much shorter then the study duration and so it is not necessary to calculate or store solution of a long study duration for each trip. A customized study duration for each trip is thus used which is calculated as: observed trip duration + buffer time before the trip + buffer time after the trip. The number of time periods for each trip is then calculated as: (customized study duration)/(time period length).

A buffer time before the trip is added because information before departure time affects the decision at the departure time. A buffer time after the trip is added to make sure attractive alternatives with slightly longer travel time are included.

### 3.6.6 Enqueue Affected States

In the original algorithm LC-CDPI, if any support point and time period of a link is updated, this link is then enqueued to check all support points and time periods of its upstream links. However, this is not necessary since not all support points and time periods of its upstream links are affected. One solution is to enqueue states (link, time, support point), instead of links. Then only affected time periods and affected support points of an upstream link are checked given downstream link states.

Affected time periods are time periods equal or smaller than the current link time period because the arrival time at a downstream link is always later than the arrival time at its upstream link. Affected support points are support points in the event collection that is the superset of the downstream link event collection. As time goes on, an event collection includes less and less support points because of the additional information collected with time. A given support point is included in only one event collection (and its subsets as time goes by). Therefore, event collection at an earlier time is a superset of that of a later time.

In addition, only affected destination states are enqueued in initialization. Affected destination states are states with time periods bigger than the trip departure time because the arrival time at any link between the origin and destination is later than the departure time.

### 3.6.7 Latest State Time and Earliest Link Arrival Time

Because all states with time periods smaller than the current time period are going to be checked, only the maximum time period of a given link and support point need to be enqueued. Therefore, a new state is first compared with all the states with the same link and support point that have been previously checked, and only enqueued when its time is the biggest among the times of those states. The states are also enqueued by decreasing order of time so that states with bigger times with be checked first to reduce the possibility of checking states with smaller times.

Furthermore, for any link, only states with time bigger than earliest possible time arriving at the link need to be enqueued, which is calculated as: departure time + static shortest path time from origin to the current link among all support points. An one-to-all shortest path algorithm is applied to calculate static shortest path time.

# CHAPTER 4

# NETWORK DATA PROCESSING

*This section is a collaborative effort with KTH Royal Institute of Technology, Stockholm, Sweden.*

Three data sets are needed for the modeling framework: network topology, vehicle trajectories, and link travel time distributions. GPS traces are used to generate the latter two data sets. Each trace consists of latitude, longitude, timestamp, status (free/hired), and the unique vehicle ID. The trace generation process is time-based and the gap between traces varies depending on the location. The GPS data is then processed using map-matching methods to produce two sets of output used in the subsequent choice set generation and evaluation: vehicle trajectories and dynamic link travel time distributions.

Hired taxi data refer to trips with passenger(s) on board, whose duration is between meter on and meter off. It is assumed that when there are passengers on board, taxi drivers have a clear origin and destination, and have routing goals similar to those of general drivers, whereas non-hired taxis roam the network in order to pick up passengers. It is likely that taxi drivers are more experienced, aggressive, and knowledgeable about the area and take a route that some commuting drivers would not take such as back roads or shortcuts to avoid difficult intersections or traffic areas. Therefore, the developed model represents behaviors of a subset of the general drivers who are knowledgeable about the network and sensitive to real-time traffic information. The methodology can be applied in the future to model more general drivers' behaviors when data becomes available.

## 4.1 Network and Map-Matching

The network is represented as a directed graph with links for streets, nodes for intersections, and locations where link attributes change. Each link has a number of attributes including speed limit, functional class and presence of traffic signal. For this application, the network is simplified so that links in series with identical speed limit and functional class attributes are merged, reducing time and memory requirements of subsequent processing. Furthermore, to ensure that there exists a path between every pair of nodes, only the largest strongly connected components of the network are used.

The GPS traces are matched to the road network using a 4-step map-matching method designed for sparse Floating Car Data (FCD), which is data collected from traced vehicles that "float" with the traffic flow [Rahmani & Koutsopoulos, 2013b]. The method first finds candidate links in the vicinity of each trace, then connects the candidate links of each pair of traces. The method then creates a candidate graph between a sequence of traces and, finally, finds the most likely path from the candidate graph.

## 4.2 Vehicle Trajectories

The start and end of each vehicle trajectory are identified through changes in the vehicle status from free to hired and vice versa since only hired taxi trips are used. For each trajectory, the data includes the vehicle ID, origin, destination, date, start time, duration of the trip, number of GPS traces, the sequence of links where the vehicle sent the traces (determined through the map-matching), and the corresponding sequence of link entry times.

## 4.3 Link Travel Time Distribution

A non-parametric method [Rahmani & Koutsopoulos, 2013a] is used to compute the link travel time distribution per time period using the map-matched GPS data. For each road segment between a pair of traces, the observed travel time (i.e., the difference between the time stamps) is decomposed to the traversed links considering their overlapping lengths. The weighted average, where the weight reflects the overlap with both the considered link and other links, over observations from different vehicles within the same time period is the estimated link travel time. The travel time estimation is performed for each time period separately for each day in the data set, producing an empirical travel time distribution with the joint support points as days.

With the available data, there are link-day-period combinations for which the travel time cannot be estimated due to lack of observations. These missing values are filled in through a sequence of inter/extrapolation steps. Furthermore, unreasonably high or low link travel times are removed to produce reliable estimates. Links with many missing values before interpolation are treated as deterministic, and a single mean travel time is estimated across all days.

## 4.4 Data Evaluation

After acquiring and processing the GPS data, an evaluation is conducted regarding factors such as the frequency of GPS traces, the level of congestion, and uncertainties of the network. For instance, GPS traces with a time gap larger than a certain threshold are filtered out to ensure accurate observations of chosen routes. Sub-networks with high levels of congestion and uncertainties are identified and extracted to allow potential adaptive route choices. They are also simplified by merging links with the same attributes and connected by nodes that are not real-life intersections. The running time and memory are thus reduced by the subnetwork extraction and link/node simplification.

# CHAPTER 5

# STOCKHOLM CASE STUDY

A case study is conducted in a subnetwork of Stockholm, Sweden. Three data sets are needed for the modeling framework: network topology, vehicle trajectories, and link travel time distributions. GPS traces generated by hired taxis from a fleet management system are used to generate the latter two data sets through map-matching and non-parametric link travel time estimation. The trace generation process is time-based and the gap between traces varies depending on the location. Only hired taxi data are used because when there are passengers on board, taxi drivers have a clear origin and destination, and have routing goals similar to those of general drivers, whereas non-hired taxis roam the network in order to pick up passengers. The passive GPS traces cannot tell us what real-time information the drivers possess. Consequently, perfect online information (POI) is assumed, which includes realized travel times on all links up to the current time, since taxi drivers are in general highly sensitive to traffic conditions and stay informed at all times.

## 5.1 Stockholm Data

### 5.1.1 Introduction

As the capital and the largest city of Sweden, Stockholm constitutes the most populated urban area in Scandinavia. As for transportation network, Stockholm is at the junction of the European routes E4, E18 and E20, and a half-completed motorway ring road exists on the south and west sides of the City Center. A subset of the Stockholm network is studied, which includes the Arlanda airport area, E4 motorway between the airport and the city, and northeast part of the inner city. In this sub-network, according to the observations of local residents, taxi drivers adapt to traffic conditions when making route choices going into and out of the city center. In particular, between point A and point B shown in Figure 5.1, there is a choice among two common routes, either the western route along E4 or the eastern route along E18 and LV276. The GPS traces utilized are generated by the taxis from a fleet management system in Stockholm. The trace generation process is time-based with data from November 1, 2012 through January 18, 2013, covering the time periods of Mondays through Fridays, resulting in 56 days (support points). The network statistics are listed in Table 5.1.


**Figure 5.1.** Road network in the Stockholm case study

### 5.1.2 Trip Sampling

For model estimation, 500 out of 4,520 trips are sampled. To make the observations diverse, wide spread locations of OD pairs are selected based on their coordinates. An inventory of coordinates of all trip ends are first obtained. The geographical ranges of

**Table 5.1.** Network statistics for the Stockholm network

| Statistic | Stockholm |
|---|---|
| # of Nodes | 2,872 |
| # of Links | 5,447 |
| # of Stochastic Links | 619 |
| # of Taxis | 1,500 |
| # of Support Points | 56 |
| Traces Time Gap | 1-2 min |
| # of Vehicle Trajectories | 4,520 |
| # of OD Pairs Evaluated | 500 |
| # of Breaking Points | 30 |
| Breaking Point Period | 5 min |
| Total Study Duration | 7:30 AM - 11:30 AM |
| Departure Time Duration | 7:30 AM - 9:00 AM |

coordinates in the airport area and downtown area are then found. The airport area contains the collection of trip ends with higher latitudes; the downtown area contains the collection of trip ends with lower latitudes. The airport area is divided into three zones and the downtown area is divided into nine zones. A total of 500 trips are then evenly sampled from each combinations of airport/downtown zones.

Two quality control criteria are also used to sample the trips, in which the GPS traces of the trip are compared with its inferred path obtained in the map-matching process. First, the travel time of the GPS traces (last GPS time stamp - first GPS time stamp) is compared to the travel time of the inferred path (entry time of corresponding first link - entry time of corresponding last link). Then normalized time difference is calculated, and sampled trips should have small normalized time difference. Normalized time difference is the travel time difference between the GPS traces and the inferred path normalized against the travel time of the GPS traces. Second, the gaps of link entry times in GPS traces are compared with the corresponding gaps of link entry times in the inferred paths. Normalized Rooted Mean Squared Error (NRMSE) for gap difference is then calculated, and the sampled trips should have small NRMSE. NRMSE is the standard deviation of the differences between gaps in

GPS traces and inferred path gaps normalized against the average gaps in GPS traces. The normalized time difference was found to be a looser criterion than the gap difference.

## 5.2 Computational Requirements

The case study is conducted on ordinary desktop configurations. The methodologies are computationally efficient and can be applied in real-life large networks.

### 5.2.1 Computer Specifications

The computer specifications are Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz, 3193 Mhz, 2 Core(s), 4 Logical Processor(2), 8G memory.

### 5.2.2 Memory

For data processing, the required memory depends on the network size. For instance, the map-matching and path inference module requires approximately 1 GB in the case of using the entire Stockholm network. For the Stockholm case study, however, 256 MB is enough since the sub-network size is relatively small. The map-matching module reads the input data (GPS traces) as a stream, and writes the output, which are trajectories, as another stream. This shows that memory consumption does not depend on the size of the input data.

For choice set generation, the major memory usage is from the storage of link travel times, which requires space calculated as "Number Of Links $\times$ Number Of Support Points $\times$ Number Of Breaking Points $\times$ Storage Unit." For example, if the travel times are stored as float numbers, the memory used for this storage would be approximately 4.6 MB for the Stockholm case study. Therefore, an ordinary desktop configuration would satisfy the requirements.

For Model Estimation, the major memory usage is from the storage of observation data, which requires space calculated as "Number Of Observations $\times$ Number Of Maximum

Alternatives in all choice sets $\times$ Number Of Factors (Number of Attributes + Availability Condition + Latent Factor Condition) associated with each alternative $\times$ Storage Unit."

### 5.2.3 Running Time

On an ordinary desktop configuration, the map-matching and path inference module processes about 50 traces per second. The running time required for the travel time estimation increases with the number of trace pairs, days, time-of-day periods and network links. For the Stockholm case study, there are 501,555 trace pairs in the data set and the computations take around 10 minutes.

In choice set generation, for fixed network parameters (number of time periods, number of support points, number of links, number of stochastic links), the running time of choice set generation is linear in link elimination period, number of simulations, and number of OD pairs. After extracting the network, choice set generation takes around 4.7 minutes per OD pair.

Model estimation takes relatively shorter time compared to data processing and choice set generation. For around 100 observations and 10 parameters, model converges within seconds. Model estimation can also be executed faster using multi threading by writing a statement in the model file indicating the maximum number of processors that will be used for the estimation.

### 5.2.4 Software

Various computer programs and software have been utilized to perform this dissertation. The map-matching and path inferring programs are coded in Java, and the travel time estimation program is coded in Matlab. The data is visualized in Google Earth and is stored in PostgreSQL database with PostGIS plugin.

Computer programs are also coded in C++ with Microsoft Visual Studio for the ORP algorithm, and choice set generation and evaluation. In the simulation method, the link travel time samples are generated by R utilizing the normal distribution function. Excel Macros are also utilized throughout the experiment to prepare the inputs and analyze the results.

Various software are able to perform model estimation. For this dissertation, Python Biogeme is chosen, which is a version of Biogeme based on the Python language. Unlike most other commercial software, it allows the user to write explicitly the model and the likelihood function, which makes the estimations of discrete choice models with latent classes and latent variables possible. In principle, Python Biogeme is able to estimate any model that uses maximum likelihood estimation.

## 5.3 Choice Set Generation and Evaluation

Path choice sets consist of routing policies transformed by duplicating paths by the number of support points, which are generated based on static and deterministic link travel times using the methods of Link Elimination, Simulation, and Changing Parameters. The travel times are changed to deterministic and static by taking average of the original link travel times over time periods and support points. In each of the 30 simulations, travel time samples are generated by independent normal distribution, with mean as the estimated travel time and standard deviation as 10 times of the mean. In Changing Parameters, the highway bias ranges from 0.5 to 0.9 with an increment of 0.1, the intersection delay ranges from 30 secs to 90 secs with an increment of 30 secs, and the changing function class penalty ranges from 30 secs to 150 secs with an increment of 30 secs. Adaptive routing policy choice sets are generated by Link Elimination and Changing Parameter methods only. Combined routing policy choice sets include both fixed paths and adaptive routing policies.

### 5.3.1 Coverage

Table 5.2 illustrates the coverage for different choice set types. For path choice sets, 456 out of 500 trips are matched resulting in a coverage is 91.2%; for adaptive routing policy choice sets, 452 trips are matched resulting in a coverage is 90.4%; for combined routing policy choice sets, in which paths are added to routing policies, the coverage is 90.4%. The goal for satisfactory coverage is considered achieved for model estimation for this study.

| **Choice Set Type** | **Overlap (%)** | **Coverage (%)** |
| --- | --- | --- |
| Path Choice Sets | 100 | 91.2 |
| Adaptive Routing Policy Choice Sets | 100 | 90.4 |
| Combined Routing Policy Choice Sets | 100 | 91.2 |

**Table 5.2.** Converge for different choice set types

### 5.3.2 Choice Set Outlier Analyses

Outlier analyses are carried out to investigate the reasons for the unmatching trips by visualization and categorization. Examples are given below for each category in which the red links represent observed GPS links and the blue path represents a generated path in the choice set. Various methods are carried out based on different categories to improve the coverage to 100%.

There are GPS errors in the observed trips. One example is shown in Figure 5.2 in which one GPS link is out of the vicinity of all other GPS links and it is likely to be a GPS error. The GPS errors are manually corrected to improve the coverage.


**Figure 5.2.** Example of an unmatched trip due to an GPS error

In addition, there are some trips containing unusually long detours, which could be caused by the traveler having a intermediate destination. One example is shown in Figure 5.3 in which the traveler takes a long detour south before traveling north to the final destination. There are 18 trips involving such intermediate destinations. A trip with an intermediate destination are can be manually divided into two trips and the major trip is kept for model estimation. Coverage is further improved by breaking up such trips.

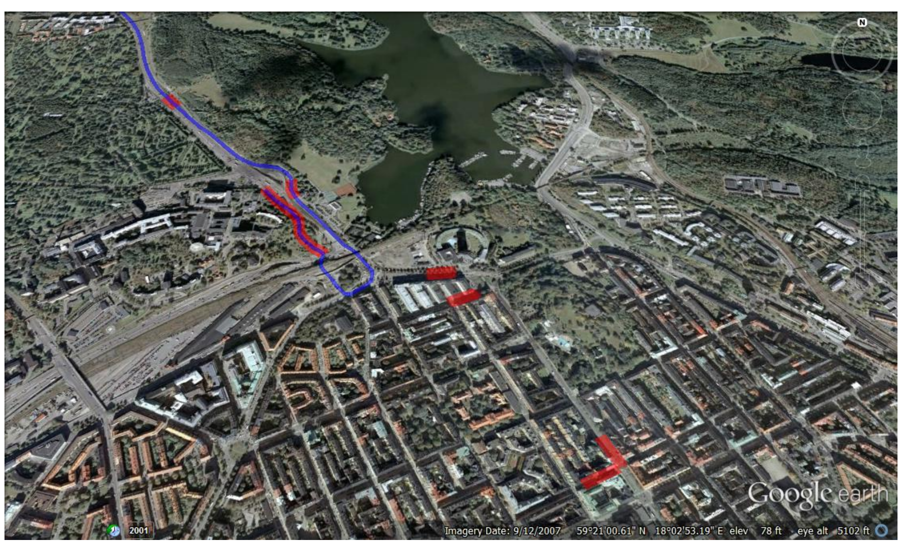


**Figure 5.3.** Example of an unmatched trip due to due to a second destination

Furthermore, some other trips are not covered around downtown area because there are too many combinations of different routes in the downtown grid network. One example is shown in Figure 5.4 in which one GPS link is not covered by any generated paths. One solution to improve the coverage is to reduce the overlap threshold, while another solution is to generate more alternatives in choice set generation. For this dissertation, the former solution is adopted and it is shown that the coverage can be increased to 100% for 90% overlap calculated based on distance.


**Figure 5.4.** Example of an unmatched trip due to downtown grid network

Table 5.3 shows a summary of different methods that improve the coverage.

**Table 5.3.** A summary of different methods that improve the coverage

| **Method** | **# of new matching trips** | **Overlap** | **Improved Coverage** |
|---|---|---|---|
| Path choice set generation | 456 | 100% | 91.2% |
| Correct GPS errors | 12 | 100% | 93.6% |
| Breakup trips with intermediate destinations | 7 | 100% | 95% |
| Reduce overlap threshold downtown grid network trips | 26 | 90% | 100% |

### 5.3.3 Adaptiveness

Based on the concept of adaptiveness, there are two extreme cases: if a routing policy realizes as the same path on all different days, the routing policy adaptiveness should be 1 divided by 56, which equals 0.0179; if a routing policy realizes as a different path on all different days, the the routing policy adaptiveness should be 56 divided by 56, which equals 1. For an observed trip, the adaptiveness of an adaptive routing policy choice set is averaged over all routing policies in the choice set.

Results show that 440 out of the 456 matched adaptive routing policy choice sets have adaptiveness bigger than 0.0179, indicating that most routing policies are realized as different paths over days. The average adaptiveness is 0.113, and the median is 0.103. Those routing policies that are realized as the same paths over days are eliminated from the adaptive routing policy choice sets and moved to the corresponding path choice sets if not already included. The adaptiveness histogram is shown in Figure 5.5.

The adaptiveness of the adaptive routing policy choice sets increases with Trip expected travel time, as shown in Figure 5.6. Trip expected travel time is averaged over all routing policies in the choice set for a given trip. This trend is intuitive, as longer trips generally motivate travelers to consider more carefully of their travel plans as well as allow for more diversion opportunities.

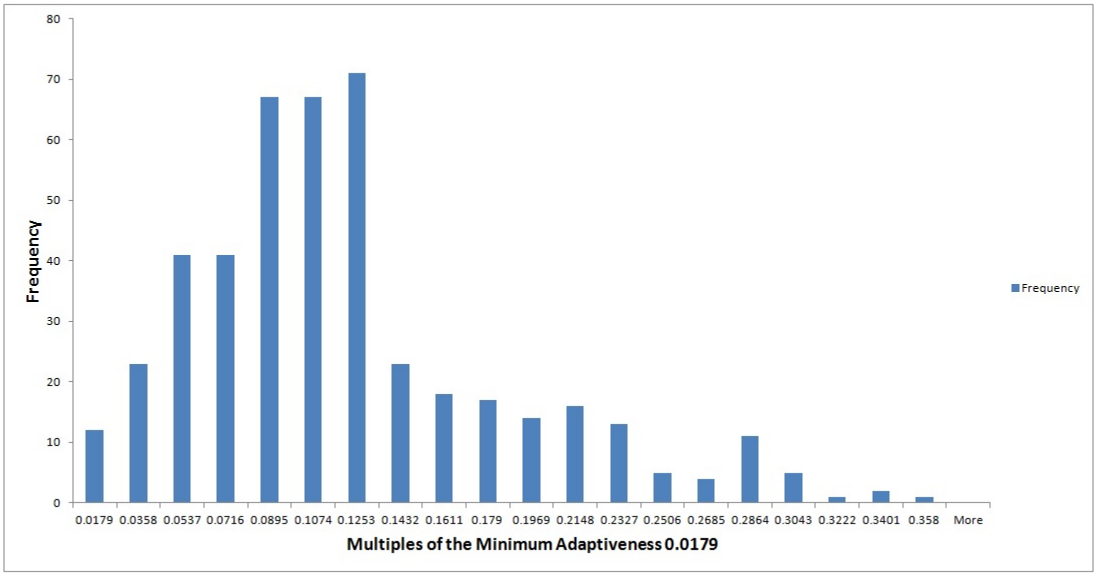


**Figure 5.5.** Adaptiveness histogram for the adaptive routing policy choice sets of matching trips

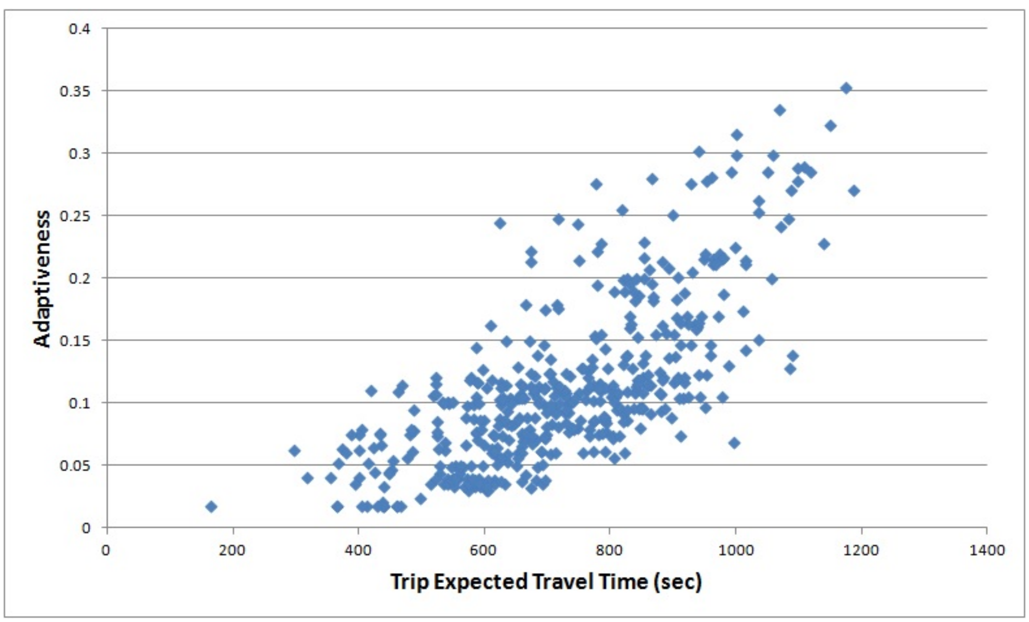


**Figure 5.6.** Adaptiveness vs. expected travel time for the adaptive routing policy choice sets of matching trips

### 5.3.4 Path Choice Set Benchmark

Based on a different data set of 997 trips, the adaptive routing policy choice set is compared to a benchmark of fixed path choice set as conventionally used for route choice in the literature utilizing the same methods, link elimination and simulation methods. Table 5.4 illustrates the comparison of coverage between adaptive routing policy choice sets and fixed path choice sets. This comparison is an indication that the adaptive routing policy choice sets could provide better coverage than the fixed path choice sets. However, since not all the possible means to improve either type of choice sets have been exhausted, future work is needed to provide a more conclusive comparison.

**Table 5.4.** Comparison of coverage on varying choice set types and methods for the Stockholm case study

| Choice Set Type | Choice Set Generation Method | OD Pairs | Overlap Threshold | Matched OD Pairs | Coverage |
|---|---|---|---|---|---|
| Adaptive Routing Policy Choice Sets | Link Elimination | 997 | 1 | 633 | 0.63 |
| | | 997 | 0.8 | 788 | 0.79 |
| | Link Elimination and Simulation | 997 | 1 | 803 | 0.81 |
| | | 997 | 0.8 | 917 | 0.92 |
| Fixed Path Choice Sets | Link Elimination | 997 | 1 | 563 | 0.56 |
| | | 997 | 0.8 | 718 | 0.72 |
| | Link Elimination and Simulation | 997 | 1 | 583 | 0.58 |
| | | 997 | 0.8 | 737 | 0.74 |

### 5.3.5 Improvements on Optimal Routing Policy Efficiency

When Algorithm LC-CDPI is applied in the Stockholm case study, the running time for choice set generation is still found to be high when a reasonable time period length is used. Therefore, efforts are made to further reduce running time based on the methodologies introduced in Section 3.6. In the base case, a total study duration of 4 hours is applied to every single trip. However, the trip length is only around 10 to 30 minutes and thus a customized study duration for each trip is calculated. A uniform buffer time of 1 minute is used for both before and after the trip. The 900 seconds time period length in the base case is also too long to render a realistic solution and thus an appropriate time period length of 12 seconds is decided based on link travel time histogram, computer memory, and running time.

Table 5.5 presents a summary of methods in improving the running time in the case study.

**Table 5.5.** A summary of methods that improve the optimal routing policy efficiency in the case study

| **Method** | **Time Period Length** | **# Time Periods** | **Running Time per Destination** |
|---|---|---|---|
| Base Case | 900s | 16 | 87s |
| Large file support in Microsoft Visual Studio 2013 C++ and newer | NA | NA | NA |
| Two-queue data structure for the label correcting algorithm | 900s | 16 | 22s |
| Appropriate time period length | 12s | NA | NA |
| Customized study duration for each trip | 12s | Customized | 286s |
| Enqueue affected states | 12s | Customized | 270s |
| Latest state time & Earliest link arrival time | 12s | Customized | 254s |

## 5.4 Model Estimation

### 5.4.1 Systematic Utility Functions

The attributes of Long Trip Dummy and Alternative Specific Constant (ASC) are in the membership function for policy user probability. Long Trip Dummy is a dummy variable that equals 1 if shortest path travel time is bigger than the threshold of 900 secs, and 0 otherwise.

The deterministic utility functions for path and policy choice are linear in parameter with attributes of Expected Travel Time (min), Travel Time Range (min), interaction term between Travel Time Range and Airport Bound, # of Signals, # of Left Turns, # of Functional Class Changes, Average Speed (m/s), as well as dummy variables for Min Expected Travel Time, Max Expected % of Highway Distance, and Min # of Functional Class Changes. For routing policies, the attributes are averaged over all support points. The parameters of Policy Size and Path Size are fixed at 1 to correct for the correlation between alternatives due to overlapping paths. The attribute of Travel Time Range (the difference of the maximum and minimum travel time) is shown as an example of travel time reliability measure, while other different measurements are also estimated. Such measurements include travel time standard deviation, variance, percentile (difference between 95 percentile and median travel time), and coefficient of variation (the ratio of the travel time standard deviation and the mean travel time). The interaction term between Travel Time Range and Airport Bound equals Travel Time Range if traveling airport bound and 0 otherwise. Average Speed is calculated as distance divided by Expected Travel Time. The parameter sets for the two classes travelers differ by a scale (Path Parameters = Scale $\times$ Policy Parameters), as introduced in Section 2.3.

### 5.4.2 Latent-Class Routing Policy Model Estimation

All model estimation was performed using BIOGEME Python 2.0 ( [Bierlaire, 2003] and [Bierlaire, 2008]). Table 5.6 presents the model estimation results of the latent-class routing policy choice model based on the 456 matching trips in choice set generation.

With a logit form membership function, it cannot be directly checked whether the policy user probability is significant at a certain level based on the parameters statistics. Therefore, to prove that the policy user probability is not zero or one, the corresponding path model (restricting path user class probability = 1) and routing policy model (restricting policy user probability = 1) are also estimated. The attributes in the restricted models are similar to those in the unrestricted model except there are no Scale or membership function related parameters. The loglikelihood ratio test is then performed on the unrestricted latent-class routing policy model over the two restricted models. Since -2(-243.573 + 235.178) = 16.790 and -2(-252.034 + 235.178) = 33.712, and $\chi_(0.90, 3)$ = 6.251 at 90% level of confidence, it can be concluded that the unrestricted model has a significant improvement in fit and the null hypothesis of homogeneous traveler behavior can be rejected, i.e., travelers are heterogeneous in terms of their ability and willingness to plan ahead and utilize real-time information. Therefore, there could be potential biases when simplified assumptions are applied that travelers follow fixed path choice under real-time information. An appropriate route choice model for uncertain networks should take into account the underlying stochastic travel times and traveler heterogeneity in terms of real-time information utilization.

The signs and magnitude of the estimates seem reasonable. Long Trip Dummy estimate is positive and significant at 0.1 level indicating travelers tend to be more strategic in longer trips. As most considered attributes, Expected Travel Time and Min Expected Travel Time Dummy parameters are significant at 0.1 level and have negative effect on travelers route choice. # of Signals and # of Left Turns estimate show that alternatives with less signals and left turns are preferred. # of Functional Class Changes and Min # of Functional Class Changes Dummy estimates suggest that travelers also prefer not to switch on/off highways frequently. Speed is also an important factor that affects travelers' route choice. For instance, given two alternatives of same travel times, many travelers choose the one with faster speed even if it has longer distance. This phenomenon is greatly related to travelers' preference on highways (highway bias). Thus highways also play a very important role, which is further substantiated by Max Expected % of Highway Distance Dummy estimate. The estimate indicates that travelers favor the alternative with longer highway distance on highways $E4$ and $E18$. The ratios of the estimates also seem reasonable. For example, the ratios suggest that 1 minute in travel time is equivalent to about 3 signals, 1.8 left turns, 6 intersections, and half of a functional class change.

While Policy Size and Path Size are fixed at 1, the parameter sets for the two classes travelers differ by a scale. Scale being significant at 0.1 level indicates that policy users and path users have different behavioral perceptions on route choice.

The measures of travel time reliability are also estimated, including travel time range, standard deviation, variance, coefficient of variance, and percentile. However, they are not significantly different from 0 at 0.1 level, indicating travelers are risk neutral in this case study. An interaction term between travel time reliability and airport bound is also estimated and found to be significantly different from 0 at 0.1 level, indicating airport bound travelers are more risk averse than downtown bound travelers probably because they are trying to catch a flight. Travel time reliability can be included to have a potentially better model as it might provide more explanatory power in prediction, or models with a larger sample size. Moreover, there could be other ways to capture travel time reliability to improve the significance of this reliability and in turn the overall model fit. Such measures include more advanced model form for risk attitude, e.g., expected utility theory (using non-linear value functions), prospect theory (using non-linear value functions and probability weighting functions).

Table 5.7 presents the estimation results for the latent-class routing policy model based on different sample size of mathcing policies. As shown in Table 5.3, after applying different coverage improvement methods, the number of matching trips is first increased to 475 without relaxing overlap threshold. The number of matching trips is further increased to 500, which is full sample size, by relaxing overlap threshold to 90% only for unmatching downtown grid network trips. Two additional models are estimated based on increased sample sizes. As expected, the estimates appear consistent across different sample sizes and the final loglikelihood decreases when the sample size increases.

**Table 5.6.** Estimation results for latent-class routing policy model and restricted models

| | **Latent-class policy model** | | **Policy User Probability=1** | | **Path User probability=1** | |
|---|---|---|---|---|---|---|
| **Parameters** | **Estimate** | **t-stat** | **Estimate** | **t-stat** | **Estimate** | **t-stat** |
| ASC | -2.00 | -1.57 | NA | | NA | |
| Long Trip Dummy (SPT>900 s) | 3.88 | 1.71 | NA | | NA | |
| Expected Travel Time (min) | -1.28 | -3.19 | -0.685 | -5.17 | -0.638 | -4.87 |
| Travel Time Range (min) | 0.0710 | 0.39 | 0.110 | 0.94 | 0.0433 | 0.46 |
| Travel Time Range * Airport Bound | -0.835 | -1.71 | -0.506 | -2.05 | -0.445 | -1.82 |
| # of Signals | -0.428 | -2.26 | -0.195 | -2.06 | -0.217 | -2.71 |
| # of Left Turns | -0.712 | -2.07 | -0.332 | -1.58 | -0.444 | -2.21 |
| # of Functional Class Changes | -2.77 | -3.33 | -1.40 | -5.86 | -1.23 | -6.43 |
| Average Speed (m/s) | 2.27 | 2.73 | 1.01 | 4.81 | 1.15 | 4.94 |
| Min Expected Travel Time | 3.04 | 5.29 | 1.33 | 2.26 | 1.21 | 3.63 |
| Max Expected % of Highway Distance | 2.28 | 2.55 | 1.50 | 2.85 | 1.06 | 3.14 |
| Min # of Functional Class Changes | 2.49 | 2.96 | 2.57 | 7.50 | 1.03 | 2.32 |
| Scale for Two Class Parameters | 0.435 | 3.98 | NA | | NA | |
| Sample Size | 456 | | 456 | | 456 | |
| # of Parameters | 13 | | 10 | | 10 | |
| Adjusted Rho Square | 0.637 | | 0.618 | | 0.639 | |
| Initial Loglikelihood | -679.577 | | -680.029 | | -695.987 | |
| Final Loglikelihood | -233.542 | | -249.786 | | -241.480 | |

NA indicates that the parameter is not included in a model

Policy Size and Path Size are fixed at 1

**Table 5.7.** Estimation results for latent-class routing policy model with different sample size of matching trips

| | **Coverage=91.2%** | | **Coverage=95.0%** | | **Coverage=100%** | |
|---|---|---|---|---|---|---|
| **Parameters** | **Estimate** | **t-stat** | **Estimate** | **t-stat** | **Estimate** | **t-stat** |
| ASC | -2.00 | -1.57 | -2.23 | -1.60 | -2.17 | -1.65 |
| Long Trip Dummy (SPT>900 s) | 3.88 | 1.71 | 3.60 | 1.75 | 4.40 | 1.20 |
| Expected Travel Time (min) | -1.28 | -3.19 | -1.15 | -2.91 | -1.06 | -3.36 |
| Travel Time Range (min) | 0.0710 | 0.39 | 0.162 | 0.94 | 0.116 | 0.79 |
| Travel Time Range * Airport Bound | -0.835 | -1.71 | -0.894 | -1.96 | -0.756 | -1.94 |
| # of Signals | -0.428 | -2.26 | -0.266 | -1.86 | -0.227 | -1.96 |
| # of Left Turns | -0.712 | -2.07 | -0.992 | -2.83 | -0.843 | -2.72 |
| # of Functional Class Changes | -2.77 | -3.33 | -2.30 | -3.49 | -2.07 | -4.12 |
| Average Speed (m/s) | 2.27 | 2.73 | 1.82 | 2.63 | 1.60 | 2.82 |
| Min Expected Travel Time | 3.04 | 5.29 | 2.78 | 5.07 | 2.45 | 5.09 |
| Max Expected % of Highway Distance | 2.28 | 2.55 | 1.78 | 2.23 | 1.57 | 2.51 |
| Min # of Functional Class Changes | 2.49 | 2.96 | 2.19 | 2.40 | 2.14 | 3.03 |
| Scale for Two Class Parameters | 0.435 | 3.98 | 0.491 | 3.86 | 0.549 | 4.62 |
| Sample Size | 456 | | 475 | | 500 | |
| # of Parameters | 13 | | 13 | | 13 | |
| Adjusted Rho Square | 0.637 | | 0.611 | | 0.612 | |
| Initial Loglikelihood | -679.577 | | -715.786 | | -740.793 | |
| Final Loglikelihood | -233.542 | | -265.223 | | -274.532 | |

Policy Size and Path Size are fixed at 1

# CHAPTER 6

# CONCLUSIONS AND FUTURE DIRECTIONS

## 6.1 Conclusions

This dissertation studies the first adaptive route choice model based on individual-level GPS observations in real-life large-scale networks, where travelers can revise their route choices based on real-time information. This model can be incorporated into a traffic prediction model to enhance its capability to evaluate transportation network management strategies and policy measures, especially those pertaining to ATIS. A case study is carried out in a real-life network, Stockholm, Sweden, based on GPS data from hired taxis. First, efficient computer algorithms are designed and implemented for choice set generation in real-life large STD networks, where link travel times are stochastically dependent random variables. Next, a latent-class, latent-choice, latent-path Policy Size Logit model is specified. Combined routing policy choice sets are then generated, where the routing policies represent alternatives that allow re-routing. The choice sets are then evaluated based on coverage and adaptiveness. It is shown that using a combination of different methods yields a satisfactory coverage of 91.2%. Outlier analyses are then carried out for unmatching trips in choice set generation. It is shown that the coverage can be increased to 95% after correcting the errors in the GPS observations. It is also shown that 100% coverage can be reached if relaxing the overlap threshold to 90% based on distance to capture those trips not matching due to the downtown grid network. Benchmark analyses comparing policy choice sets and path choice sets based on link elimination and simulation methods are also carried out on a different data set and the results indicates that a policy choice set could potentially provide better coverage and capture the adaptive nature of route choice. The adaptiveness evaluation shows that most of the routing polices are adaptive and realized as different paths on different days.

A latent-class routing policy model is first estimated based on the 91.2% coverage (456 trips). Two additional models are then estimated based on the 95% coverage (475 trips) and 100% coverage (full sample). The estimates in the three models appear consistent across different sample sizes. The Loglikelihood ratio test is performed on the unrestricted model (latent-class routing policy model) over two restricted models (path model and policy model) based on the 456 trips. The results indicate that travelers are heterogeneous in terms of their ability and willingness to plan ahead and utilize real-time information. Thus a fixed path model as commonly used in the literature may lose explanatory power due to simplified assumptions of network stochasticity and travelers' utilization of real-time information. Therefore, an appropriate route choice model for uncertain networks should consider the underlying stochastic travel times and traveler heterogeneity in terms of real-time information utilization.

## 6.2 Future Directions

Since the advances in GPS technology make it possible to track individual route choices of travelers, GPS taxi data is utilized for this route choice study. However, this type of passive monitoring cannot generate observations regarding the information access of travelers, and the fact that the information in a dynamic network changes over time and space makes it more difficult. This study circumvents this difficulty by assuming POI access, but it hardly represents real-life situations. First, travelers' do not consider link travel time but path travel time. Second, travelers do not make rerouting decisions at every node but at important diversion points such as bridges and tunnels. For example, a VMS message "Incident ahead - 30 min to Summer Tunnel" indicates path travel time including multiple links and travelers may only make one rerouting decision during their trip which is at the tunnel. Therefore more realistic information access are outside the scope of this dissertation and are to be explored in future studies. Drivers' information access can be obtained through a number of means, for example, periodic survey questions delivered to smartphones in real time after some major diversion points are traversed, location prompted recall survey at the end of day, and records of in-vehicle GPS navigation systems that provide real-time information.

GPS data also cannot generate socio-economic attributes of travelers, such as driving experience, education, and trip purpose. Such attributes can be obtained through SP surveys and combined with RP data in future studies. These socio-economic attributes could also be included in the membership function for the policy user probability. Currently the probability only depends on the trip attributes such as trip length, but it is hardly transferable to another driver population.

Adaptive path users as introduced in Section 1.2.3.2 is not studied from this dissertation, and thus a future direction is to include such users as a third latent-class and investigate the existence, or lack thereof, adaptive path choice behavior. An even more complex way of modeling routing policy choice is successively estimating a sequence of routing policy choice models at each decision node in a similar fashion that adaptive path choice is modeled, whereas this dissertation assumes that the travelers choose a routing policy only in the beginning of the trip.

The accuracy of real-time information is the basis for routing policy choice modeling. However, the information provided to travelers is not necessarily always precise or timely in operational settings since it requires considerable ammounts of data processing. Adapting to poor information may not be helpful or may even be harmful to travelers. Furthermore, after repeatedly making bad route choice decisions due to poor information, travelers could lose trust and eventually ignore real-time information. Therefore, a potential future direction is to investigate the impacts of poor information on adaptive route choice.

While this research mainly contributes to planning models, the ORP algorithm has the potential to be applied in operation settings as well. The running time of the ORP algorithm is around 4 minutes, which is not timely for providing real-time route guidance to travelers in real-time. However, if its running time can be further reduced or if an effective heuristic can be proposed, it may potentially provide travelers with better route guidance in uncertain networks with real-time information.

In addition, as introduced in Section 1.2.3.4, a new realm of route choice study is to estimate route choice models without choice set generation, or link-based dynamic discrete choice models. So far such studies have only been carried out on path choice models in static and deterministic networks. Therefore, a potential future direction is to explore link-based dynamic discrete choice models for routing policy choice.

# BIBLIOGRAPHY


(1996). *A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks*, Proceedings of the 13th International Symposium on Transportation and Traffic Theory, Lyon, France.

Abdel-Aty, M. & Abdalla, M. F. (2004). Modeling drivers' diversion from normal routes under ATIS using generalized estimating equations and binomial probit link function. *Transportation*, *31*, 327–348.

Abdel-Aty, M. & Abdalla, M. F. (2006). Examination of multiple mode/route-choice paradigms under ATIS. *IEEE Transactions on Intelligent Transportation Systems*, *7*(3), 332–348.

Azevedo, J. Costa, M. S. M. J. S. & Martins, E. V. (1993). An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, *69*, 97–106.

Balakrishna, R., Ben-Akiva, M., Bottom, J., & Gao, S. (2013). Information impacts on traveler behavior and network performance: State of knowledge and future directions. In S. V. Ukkusuri & K. Ozbay (Eds.), *Advances in Dynamic Network Modeling in Complex Transportation Systems*, volume 2 of *Complex Networks and Dynamic Systems* (pp. 193–224). Springer New York.

Batley, R. & Ibanez, N. *International Choice Modeling Conference*.

Bekhor, S. Ben-Akiva, M. E. & Ramming, S. (2006). Evaluation of choice set generation algorithms. *Annals of Operations Research*, *144*(1).

Ben-Akiva, M. & Bierlaire, M. (1999). Discrete choice methods and their applications to short-term travel decisions. In R. Hall (Ed.), *Handbook of Transportation Science* (pp. 5–34). Kluwer.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. N., & Mishalani, R. (2011). A dynamic discrete choice approach for consistent route choice model estimationn. *Presented at the Swiss Transport Research Conference (STRC)*.

Ben-Akiva, M., Gao, S., Wei, Z., & Wen, Y. (2012). A dynamic traffic assignment model for highly congested urban networks. *Transportation Research Part C*, *24*, 62–68.

Ben-Akiva, M. E. Bergman, M. D. A. & Ramaswamy, R. (1984). Modeling inter urban route choice behaviour. *Proceeding of the 9th International Symposium on Transportation and Traffic Theory*.

Bierlaire, M. (2003). Biogeme: a free package for the estimation of discrete choice models. *Proceedings of the 3rd Swiss Transport Research Conference, Ascona, Switzerland*.

Bierlaire, M. (2008). An introduction to biogeme version 1.6. *http://biogeme.epfl.ch*.

Bogers, E., Viti, F., & Hoogendoorn, S. (2005). Joint modeling of advanced travel information service, habit, and learning impacts on route choice by laboratory simulator experiments. *Transportation Research Record*, (1926), 189–197.

Bolduc, D. & Ben-Akiva, M. (1991). A multinomial probit formulation for large choice sets. Proceedings of the 6th International Conference on Travel Behaviour.

Bovy, P. H. L. & Fiorenzo-Catalano, S. (2006). Stochastic route choice set generation: behavioral and probabilistic foundations. *Proceedings of the 11th International Conference on Travel Behaviour Research, Kyoto, Japan*.

Boyles, S. (2009). An exact label-correcting algorithm for the online shortest path problem in cyclic networks. *In review, Networks*.

Boyles, S. (2012). Bush-based sensitivity analysis for approximating subnetwork diversion. *Accepted for publication in Transportation Research Part B*.

Boyles, S. & Waller, S. T. (2010a). A mean-variance model for the minimum cost flow problem with stochastic arc costs. *Networks*, *56*.

Boyles, S. & Waller, S. T. (2010b). Traffic network design and analysis. *Encyclopedia of Operations Research*.

Boyles, S. & Waller, S. T. (2011). Optimal information location for adaptive routing. *Networks and Spatial Economics*, *11*.

Chabini, I. (2000). Minimum expected travel times in stochastic time-dependent networks revisited. *Internal Report. MIT, Cambridge, MA*.

Chatterjee, K. & McDonald, M. (2004). Effectiveness of using variable message signs to disseminate dynamic traffic information: Evidence from field trials in European cities. *Transport Review*, *24*(5), 559–585.

Chorus, C. G., Molin, E. J. E., & van Wee, B. (2006). Use and effects of advanced traveller information services (atis): A review of the literature. *Transport Reviews*, *26*(2), 127–149.

de la Barra, T. Perez, B. & Anez, J. (1993). Multidimensional path search and assignment. *Proceedings of the 21st PTRC Summer Meeting*, 307–319.

Ding, J. & Gao, S. (2012). An optimal adaptive routing algorithm for large-scale stochastic time-dependent networks. *Presented at TRB 91st Annual Meeting, Washington, D.C.*

Ding, J., Gao, S., Jenelius, E., Rahmani, M., Huang, H., Ma, L., Pereira, F., & Ben-Akiva, M. (2014). Routing policy choice set generation in stochastic time-dependent networks: Case studies for stockholm and singapore. *Transportation Research Record*, (2466), 76–86.

Fosgerau, M., Frejinger, E., & Karlstrom, A. (2011). A dynamic discrete choice approach for consistent route choice model estimationn. *Presented at the Swiss Transport Research Conference (STRC)*.

Fosgerau, M., Frejinger, E., & Karlstrom, A. (2013). A link based network route choice model with unrestricted choice set. *Transportation Research Part B*, *56*, 70–80.

Frejinger, E. (2007). *Route choice analysis: Data, models, algorithms and applications*. PhD thesis, Ecole Polytechnique Federale de Lausanne.

Frejinger, E. (2008). Route choice analysis: Data, models, algorithms and applications. *PhD thesis, Ecole Polytechnique Federale de Lausanne*.

Frejinger, E. & Bierlaire, M. (2007). Capturing correlation with subnetworks in route choice models. *Transportation Research Part B*, *41*(3), 363–378.

Frejinger, E., Bierlaire, M., & Ben-Akiva, M. (2009). Sampling of alternatives for route choice modeling. *Transportation Research Part B*, *43*(10), 984–994.

Friedrich, M. Hofsass, I. & Wekeck, S. (2001). Timetable-based transit assignment using branch and bound. *Transportation Research Record*, *1752*.

Gao, S. (2004). Optimal adaptive routing and traffic assignment in stochastic time-dependent networks. *PhD Dissertation, Massachusetts Institute of Technology, Cambridge, MA*.

Gao, S. (2005). *Optimal Adaptive Routing and Traffic Assignment in Stochastic Time-Dependent Networks*. PhD thesis, Massachusetts Institute of Technology.

Gao, S. & Chabini, I. (2006a). Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B*, *40*(2), 93–122.

Gao, S. & Chabini, I. (2006b). Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B*, *40*(2), 93–122.

Gao, S., Frejinger, E., & Ben-Akiva, M. (2008). Adaptive route choice models in stochastic time-dependent networks. *Transportation Research Record*, *2085*, 136–143.

Gao, S. & Huang, H. (2012). Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks. *Transportation Research Part C*, *21*(1), 196–213.

Ghosh, A. (2001). Valuing time and reliability: commuters' mode choice from a real time congestion pricing experiment. *PhD thesis*.

Hall, R. W. (1986). The fastest path through a network with random time-dependent travel times. *Transportation Science*, *20*(3), 182–188.

Hensher, D. (2010). Hypothetical bias, choice experiments and willingness to pay. *Transportation Research Part B (special issue)*.

Hoogendoorn-Lanser, S. (2005). Modelling travel behaviour in multi-modal networks. *PhD thesis, Delft University of Technology*.

Lam, T. C. & Small, K. A. (2001). The value of time and reliability: measurement from a value pricing experiment. *Transportation Research Part E*, *37*, 231–251.

Li, Z., Hensher, D., & Rose, J. (2010). Willingness to pay for travel time reliability in passenger transport: a review and some new empirical evidence. *Transportation Research Part E*, *46*, 384–403.

Liu, H., He, X., & Recker, W. (2007). Estimation of the time-dependency of values of travel time and its reliability from loop detector data. *Transportation Research Part B*, *41*, 448–461.

Mahmassani, H. & Liu, Y. H. (1999). Dynamics of commuting decision behaviour under advanced traveller information systems. *Transportation Research Part C*, *7*, 91–107.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Network and Spatial Economics*, *1*, 267–292.

Miller-Hooks, E. D. (2001). Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, *37*.

Miller-Hooks, E. D. & Mahmassani, H. S. (2000a). Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, *34*.

Miller-Hooks, E. D. & Mahmassani, H. S. (2000b). Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, *34*(2), 198–215.

Pallottino, S. (1984). Shortest-path methods: Complexity, interrelations and new propositions. *Networks*, *14*, 257–267.

Pape, U. (1974). Implementation of efficiency of moore-algorithms for the shortest root problem. *Mathematical Programming*, *7*, 212–222.

Papinski, D., Scott, D. M., & Doherty, S. T. (2009). Exploring the route choice decision-making process: A comparison of planned and observed routes obtained using person-based GPS. *Transportation Research Part F*, *12*, 347–358.

Peeta, S. & Ramos, J. L. J. (2006). Driver response to variable message signs-based traffic information. In *IEE Proceedings in Intelligent Transportation Systems*, volume 153, (pp. 2–10).

Peeta, S. & Yu, J. W. (2005). A hybrid model for driver route choice incorporating en-route attributes and real-time information effects. *Networks and Spatial Economics*, *5*, 21–40.

Pillat, J. Mandir, E. & Friedrich, M. (2011). Dynamic choice set generation based on global positioning system trajectories and stated preference data. *Transportation Research Record: Journal of the Transportation Research Board*, (2231), 18–26.

Polydoropoulou, A., Ben-Akiva, M., Khattak, A., & Lauprete, G. (1996). Modeling revealed and stated en-route travel response to advanced traveler information systems. *Transportation Research Record*, (1537), 38–45.

Prato, C. (2011). Latent factors and route choice behavior. *LAP Lambert Academic Publishing*, 192.

Prato, C. (2012). Meta-analysis of choice set generation effects on route choice model estimates and predictions. *Transport*, *27*(3), 286–298.

Prato, C., Rasmussen, T., & Nielsen, O. (2014). Estimating value of congestion and value of reliability from the observation of car drivers' route choice behavior. *Proceedings of the 93rd Annual Meeting of the Transportation Research Board, January 12-16*.

Prato, C. G. & Bekhor, S. (2006). Applying branch and bound technique to route choice set generation. *Presented at the 85th Annual Meeting of the Transportation Research Board*.

Rahmani, M. Jenelius, E. & Koutsopoulos, H. N. (2013a). Route travel time estimation using low-frequency floating car data. *2013 16th International IEEE Conference on Intelligent Transportation Systems*.

Rahmani, M. & Koutsopoulos, H. N. (2013b). Path inference from sparse floating car data for urban networks. *Transport Research Part C*, *30*, 41–54.

Ramming, S. (2002). Network knowledge and route choice. *PhD thesis, Massachusetts Institute of Technology*.

Razo, M. & Gao, S. (2010). Strategic thinking and risk attitudes in route choice: A stated preference approach. *Transportation Research Record*, (2085), 136–143.

Razo, M. & Gao, S. (2013). A rank-dependent expected utility model for strategic route choice with stated preference data. *Transportation Research Part C*, *27*, 117–130.

Rieser-Schussler, N. Balmer, M. & Axhausen, K. (2012). Route choice sets for very high resolution data. *Transportmetrica*, *2012*, 1–21.

Small, K., Noland, R., Chu, X., & Lewis, D. (1999). Valuation of travel-time savings and predictability in congested conditions for highway user-cost estimation. *NCHRP Report 431, Transportation Research Board, National Research Council*.

Small, K., Noland, R., & Koskenoja, P. (1995). Socio-economic attributes and impacts of travel reliability: a stated preference approach. *California PATH Research Report, UCB-ITS-PRR-95-36*.

Small, K., Winston, C., & Yan, J. (2005). Uncovering the distribution of motorists preferences for travel time and reliability: implications for road pricing. *Econometrica*, *73 (4)*, 1367–1382.

Spissu, E. Meloni, I. & Sanjust, B. (2011). Behavioral analysis of choice of daily route with data from global positioning system. *Transportation Research Record: Journal of the Transportation Research Board*, (2230), 96–103.

Srinivasan, K. K. & Mahmassani, H. (2003). Analyzing heterogeneity and unobserved structural effects in route-switching behavior under ATIS: A dynamic kernel logit formulation. *Transportation Research Part B*, *37*, 793–814.

Srinivasan, S. & Dhakar, N. (2013). Route-choice modeling using gps-based travel surveys. *Final Report to the CENTER FOR MULTIMODAL SOLUTIONS FOR CONGESTION MITIGATION*.

Tian, H., Gao, S., Fisher, D. L., & Post, B. (2011). Route choice behavior in a driving simulator with real-time information. In *90th TRB Annual Meeting DVD*.

Tsirimpa, A., Polydoropoulou, A., & Antoniou, C. (2007). Development of a mixed multinormal logit model to capture the impact of information systems on travelers' switching behavior. *Journal of Intelligent Transportation Systems*, *11*(2), 79–89.

Van der Zijpp, N. J. & Fiorenzo-Catalano, S. (2005). Path enumeration by finding the constrained k-shortest paths. *Transportation Research Part B: Methodological*, *39*(6), 545–563.

Waller, S. T. & Ziliaskopoulos, A. K. (2002). On the online shortest path problem with limited arc cost dependencies. *Networks*, *40*(4), 216–227.

Yai, T., Iwakura, S., & Morichi, S. (2002). Real-time simulation of traffic demand-supply interactions within dynamit. *Transportation WId Network Analysis: Current Trends. Miscellenea in Honor ofMichael Florian (M. Gendreau and P. Marcotte, eds.), Kluwer, Netherlands*, 19–36.

Yan, J. Heterogeneity in motorists' preferences for travel time and time reliability: empirical finding from multiple survey data sets and its policy implications. *PhD Thesis*.

Ziliaskopoulos, A. & Mahmassani, H. (1993). Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications. *Trans Res Rec*, *1408*.