Optimal Adaptive Routing and Traffic Assignment in Stochastic Time-Dependent Networks

by

Song Gao

B.S., Civil Engineering, Tsinghua University, China (1999)
M.S., Transportation, MIT (2002)

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2005

Author . . . .
Department of Civil and Environmental Engineering
October 27, 2004

Certified by . . . .
Moshe E. Ben-Akiva
Edmund K. Turner Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by . . . .
Andrew J. Whittle
Chairman, Departmental Committee on Graduate Studies

Optimal Adaptive Routing and Traffic Assignment in Stochastic Time-Dependent Networks

by

Song Gao

Submitted to the Department of Civil and Environmental Engineering
on October 27, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Transportation

Abstract

A stochastic time-dependent (STD) network is defined by treating all link travel times at all time periods as random variables, with possible time-wise and link-wise stochastic dependency. A routing policy is a decision rule which specifies what node to take next out of the current node based on the current time and online information. A formal framework is established for optimal routing policy problems in STD networks, including generic optimality conditions, and a comprehensive taxonomy with insights into variants of the problem. A variant pertinent to road traffic networks is studied in detail, where a discrete joint distribution of link travel times is used to accommodate the most general stochastic dependency among link travel times, and the access to perfect online information about link travel times is assumed. Both exact and approximation solution algorithms are designed and tested. The criteria of optimality are then extended to reliability measures, such as travel time variance and expected early/late schedule delays.

The first routing-policy-based stochastic dynamic traffic assignment (DTA) model is established. A general framework is provided and the equilibrium problem is formulated as a fixed point problem with three components: the optimal routing policy generation module, the routing policy choice model and the policy-based dynamic network loader. An MSA (method of successive averages) heuristic is designed. Computational tests are carried out in a hypothetical network, where random incidents are the source of stochasticity. The heuristic converges satisfactorily in the test network under the proposed test settings. The adaptiveness in the routing policy based model leads to travel time savings at equilibrium. As a byproduct, travel time reliability is also enhanced. The value of online information is an increasing function of the incident probability. Travel time savings are high when market penetrations are low. However, the function of travel time saving against market penetration is not monotonic. This suggests that in a travelers' information system or route guidance system, the information penetration needs to be chosen carefully to maximize benefits.

Thesis Supervisor: Moshe E. Ben-Akiva
Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Acknowledgments

Thank Ismail Chabini for his supervision, guidance and support during my first four and half years' stay at MIT. Thank Moshe Ben-Akiva, my thesis supervisor, and Nigel Wilson, Patrick Jaillet, and Jon Bottom, my thesis committee members, for their significant contributions to the thesis.

Thank the National Science Foundation for funding the thesis research.

Thank ACTS, the great research group for me to work in.

Thank my parents for always being there. This work is dedicated to them.

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Road traffic congestion is a significant problem of modern society which has multiple impacts. To an individual traveler, congestion reduces the quality of life by consuming leisure time, increasing anxiety, and wasting personal resources. To firms, congestion reduces the productivity of employees and increases freight transportation costs. To society as a whole, congestion negatively affects environmental quality by causing more tail-pipe emissions and noise, and endangers traffic safety by raising drivers' stress and fatigue.

Congestion can be classified as recurrent or non-recurrent. Recurrent congestion is due to the mismatch between demand and supply under normal conditions. Usually traffic infrastructure is updated in a fairly long cycle. The design capacity of a road segment is calculated based on the projected demand at the time of the infrastructure design, which could be below the actual demand at a later time. Recurrent congestion is usually seen in peak hours, but if the capacity is significantly low compared to average demand, congestion is likely to spread outside peak hours. Non-recurrent congestion is due to disturbances to the traffic network that reduce road capacities, such as incidents, vehicle breakdown, bad weather, work zones, special events, and so on. Some of these disturbances are completely unpredictable, such as incidents and vehicle breakdown. Others are predictable to some extent, such as bad weather, work zones and special events, but usually there are prediction errors. A weather forecast is usually in a probabilistic format, e.g. a precipitation probability of 90%. Work zones and special events are scheduled, so they are less unpredictable in some sense, but the schedules might not be available to the travelers in a timely manner, and thus are actually unpredictable to travelers. Non-recurrent congestion is a significant part of the total congestion, as described in the 2003 Urban Mobility Report by the Texas Institute of Transportation [45, p. V]: "Crashes, vehicle breakdown, weather, special events, construction and maintenance activities greatly affect the reliability of transportation systems; these delays account for about 50 percent of all delay on the roads."

It is commonly recognized that building more infrastructure, which is usually politically, financially, and environmentally constrained, is not the only remedy to congestion. Furthermore, new infrastructure will induce more demand, which could diminish the benefit from the increased capacity or even make the congestion worse. Nowadays, traffic measures to relieve congestion are generally based on the concept of making best use of current infrastructure with the advances in information technology, which is the underlying idea of Intelligent Transportation Systems (ITS). Among the various sub-systems of ITS, advanced traveler information systems (ATIS) aim to provide travelers with updated and useful information about network conditions, in hope that a better informed traveler can make a better decision, and collectively better decisions by many travelers would result in a relief from congestion. The value of ATIS is most evident when traffic conditions are stochastic. For example, when an incident happens, a timely notice by ATIS to travelers who plan to take the route with incident would be quite beneficial. Otherwise, in a network where traffic quantities are almost certain, travelers are already quite well-informed and ATIS has little to provide.

An outstanding feature of congested traffic networks is the stochasticity in traffic quantities, such as travel time, link volume, queue length, and so on, on a day-to-day base. The travel time from home to work on a Monday morning could be different from that on a Tuesday morning, or another Monday morning of the next week. The randomness can come from multiple sources. One of the most significant sources is the disturbances that cause non-recurrent congestion, as described in the previous paragraph. Traffic conditions with recurrent congestion, on the other hand, are also usually different from day to day, largely because of fluctuations in origin-destination (OD) trips. The fluctuations can be in both the total number of OD trips and the spread of OD trips over departure times. Travelers with non-commuting trip purposes might decide not to take a trip at a particular day, due to other personal business, and the no-travel decisions collectively result in a random number of OD trips. Travelers might also respond to congestion by shifting departure times from day to day, and thus there exists a random pattern in OD trips' spread. Fluctuations in OD trips combined with random disturbances in supply can bring a lot of stochasticity to a traffic network.

Travelers make decisions (destination, mode, departure time, route) based on their information about the traffic network. The information can be obtained through a wide range of means: travelers' own experience, word of mouth, radio broadcast, variable message signs (VMS), in-vehicle communication system, and so on. The information can be classified as *a priori* or online. *A priori* information is about the general picture of the day-to-day fluctuations of traffic quantities, e.g. the travel time on a bridge is 30 seconds on average, but roughly once in a month, the travel time is unusually high, due to various reasons. Online information is about the traffic condition on a specific day, e.g. an incident just occurred on this bridge, and it will probably last for 20 to 30 minutes. This classification is meaningful only when there is stochasticity in the network, such that online information is different from *a priori* information. Destination, departure time and mode decisions are usually made at origins only and can hardly be changed en route, while route decisions can be changed en route more easily and thus benefit more from online information. ATIS can provide both *a priori* and online information. Travelers only have personal experience on their selected routes. In order to obtain *a priori* information about the whole network, they need to go beyond their personal experience, and one of the good sources is ATIS. ATIS can provide travelers with reports of traffic conditions in the past and possibly predictions about the future, for the temporal and spatial ranges and in formats specified by travelers. Combining all sources of *a priori* information, travelers can form their own general pictures about the network. Nevertheless, the benefit of ATIS is primarily embodied through the provision of online information, especially in a network disturbed randomly by incidents, vehicle breakdowns, bad weather, work zones, special events, and so on.

There are various mechanisms for providing online information, i.e. various implementations of ATIS, and they differ in the spatial and temporal availability, the quality, and the format of information provided. A VMS is usually fixed in location and thus only travelers passing it can obtain the information. It is also limited in the amount of information it can provide, due to the limitation of the display panel. Usually it simply tells traveler that an incident happened somewhere, and sometimes with estimated delay on an affected major route. Radio-based or telephone-based systems can provide information to travelers anywhere in the radio coverage or when a telephone is available (with the popularity of cell phones, this actually can be anywhere). More detailed information is available compared to VMS, such as that on a number of alternative routes, or on some specific locations of interest to the traveler. There is still a limitation in the amount of information provided, as travelers who listen to the radio or call an information provider usually process information in their mind, whose processing ability is quite limited, given that driving already poses a large workload. More advanced in-vehicle systems usually contain a database of road map, travel times under normal conditions, records of past incidents, and so on, and can communicate with information centers to obtain very detailed and updated information. They usually have significant processing power, and can process a large amount of information, and display processing results as requested by travelers.

Travelers' routing decisions in a stochastic network with online information is conceivably different from those in a deterministic network. It is generally believed that adaptive routing will save travel time and enhance travel time reliability. For example, in a network with random incidents, if one does not adapt to an incident scenario, he/she could be stuck in the incident link for a very long time. However, if adequate online information is available about the incident and the traveler adapts to it by taking an alternative route, he/she can save travel time compared to the non-adaptive case. The adaptiveness also ensures that the travel time is not prohibitively high in incident scenarios, and thus provides a more reliable travel time.

It is therefore a very interesting research question how an individual traveler makes adaptive routing decisions in a stochastic and time-dependent network. The network is time-dependent, since in an operational context, within-day dynamics should be considered. Preferably the research on adaptive routing should address the following two issues:

- Traffic variables are not only random, they are also usually correlated link-wise and time-wise. We use link travel times as an example. If the randomness comes from weather, then link travel times of the whole network over a certain time period are correlated. If the randomness comes from incidents, then link travel times around the incident location and around the incident duration are correlated. It is desirable to capture these correlations.

- As discussed before, there are a large variety of information provision mechanism and thus a large variety of information accessibility situations. It is preferably to formulate the problem so that simplified assumptions on information accessibility are avoided. Common simplified assumptions include no online information, and full information (knowing the future for sure).

After understanding how an individual traveler makes adaptive routing decisions, another research question would be: what will be the network-level impact if many travelers make adaptive routing decisions? In a congested network, the stochastic nature of traffic variables affects travelers' routing decisions, which in turn affect traffic conditions. The interaction between supply and demand in a stochastic dynamic network needs to be captured to evaluate the network-level impact. This interaction in a deterministic network (with possible perception errors from the demand side) is captured by a conventional dynamic traffic assignment (DTA) model. The question is then how to establish a DTA model in a stochastic time-dependent network where travelers make adaptive routing decisions. An interesting application of such a model would be the investment assessment of ATIS. ATIS requires a large investment in sensors, communication equipment, computing equipment and other supporting systems. In order to perform a cost-benefit analysis of such a system, a model is needed to evaluate the benefit of ATIS. This model allows travelers to make adaptive decisions with the help of online formation provided by ATIS, and can derive network-level measures of effectiveness, such as total travel time or travel time reliability. This model is also able to derive network-level measures of effectiveness when no ATIS are available, since no information is one possible information accessibility situation and can be handled by the general adaptive routing method as described before. The difference in travel time or reliability between the ATIS and no-ATIS scenarios can be computed and translated into monetary values, and thus a cost-benefit analysis is possible.

1.2 Scope of the Thesis

This thesis is built around a core concept: routing policy. As discussed in the previous section, adaptive routing decisions are preferred in a stochastic time-dependent network equipped with ATIS. The definition of an adaptive routing problem is itself a research question. The questions involved include:

- How to describe a stochastic time-dependent network? In the literature, each link travel time at each time period is defined as a random variable, but they are usually assumed implicitly or explicitly as independent, which is not realistic in traffic networks.

- How to define traffic information? In reality, information can take many forms. The subject of descriptive information can be: link travel time, path travel time, queue length, delay compared to normal travel time, incident occurrence, incident severity, incident duration, weather forecast, surges in OD trips, and so on. If the information is prescriptive, the subject would be: what alternative route to take? The temporal and geographical characteristics of information also vary a lot. For example, in what frequency is information provided? Is the information updated to the current time or lagged? Does the information contain what have happened in the past or predictions about future network conditions, or both? Where is the information available? What locations are reported by the information?

- How to define a routing decision? Since the routing is carried out in a stochastic network, should the decision itself be stochastic in some sense? Suppose decisions are made only at nodes. One of the possibilities is: for any specific time and information, choose what next link to take. Another possibility is: for any specific time and information, assign probabilities to a set of outgoing links and randomly take a next link according to its probability.

Ideally, all the complications involved in an actual adaptive routing problem are considered. However, this will inevitably make a problem intractable, and a trade-off between reality and tractability must be made. This thesis defines the adaptive routing problem with reasonable assumptions on the above three issues, while still addressing a fairly general version of the problem.

A stochastic time-dependent (STD) network is defined as a network where link travel times are random variables with time-dependent distributions. A joint distribution of all random variables is used such that both link-wise and time-wise stochastic dependencies of link travel times are modeled. Travelers are assumed to have this joint distribution available *a priori*, which can be obtained through personal experience, word of mouth, ATIS or a combination of these means. Sometimes information might not be in the form of link travel times, however it is assumed that travelers can process the information *a priori* and transform it into in the form of link travel times. The joint distribution is a rather general description and can handle a wide range of *a priori* network knowledge situations. For example, sometimes a traveler only has knowledge about routes actually traveled, and little knowledge about alternative routes. In this case, the *a priori* distribution of link travel times on traveled routes is used based on his/her experience, while those on other routes are assumed to be free flow travel times or plus a mark-up. This is a reasonable assumption, as a traveler can easily obtain this knowledge on untraveled routes from a map. The underlying topology of the network is assumed to be deterministic. In fact, a stochastic topology can be represented by assigning positive probabilities to prohibitively high travel times on involved links.

Online information is assumed to be in the form of travel times of a set of links at a set of time periods. This is a reasonable assumption, given that technologies to detect real-time link travel times are available, either indirectly through loop detectors, or directly through probe vehicles or cameras. The specific set of links and set of time periods are dependent on the information provision mechanism, and are functions of the location of a traveler and the current time. For example, information provided by a VMS is about links downstream from the VMS location and is only available when a traveler is passing the VMS location. Therefore, at any time and node, the online information could contain realized travel times on links that were reported by VMSs along the route the traveler has traveled so far. If the traveler has not passed any VMS, his/her online information is empty. An advanced in-vehicle communication system, on the other hand, could provide online information about all monitored links in the network. The time lag in information provision can also be represented, by having the set of time periods contain only time periods up to a certain time in the past. If the online information updated in a fixed interval, say 15 minutes, at 7:00, 7:15, 7:30 and etc, then the set of time periods at times 7:15 through 7:29 contain time periods up to 7:15 only. In most of the cases, the information is about realized link travel times, and this thesis focuses on such cases. However, it can include predictions about future link travel times, such that the set of time periods contain those beyond the current time. Theoretically, the set of links and set of time periods can be any combination of all possible links and time periods. The set is a function of location and time. As will be discussed later, routing decisions are only made at nodes, it is thus sufficient to represent locations in terms of nodes only.

Only descriptive information is considered in this thesis. The study of prescriptive information itself can be a deep research question and is not in the scope of this thesis.

Routing decisions are made at nodes only and every node is viewed as a decision node. At a given node and a given time, with given online information, a decision is made on what node to take next. In another word, a traveler does not flip a coin to decide. Research on random selection of outgoing links is an interesting topic itself, and is included in the future directions. The realization of the decision is also deterministic, i.e. if a traveler chooses to take a next node, she/he will end up arriving at that node. The action of U-turn is not modeled.

In a stochastic time-dependent (STD) network with online information, the mapping from any possible combination of node, time and online information to a next node defines a decision rule and is formally defined as a routing policy in this thesis. An optimal routing policy (ORP) is a routing policy that moves a traveler on a network from one node to another in least expected travel time, least travel time variance, or least expected schedule delay. A framework and taxonomy for optimal routing policy problems in STD networks are presented that cover variants in the literature and suggests new variants. One new variant that is particularly pertinent in traffic networks is studied in depth. A summary of the research and findings on optimal routing policy (ORP) problems will be presented in the next section and discussed in detail in Chapters 2, 3 and 4. In fact, ORP problems are of fundamental research significance and have a wide domain of applications beyond transportation.

The definition of routing policy enables a systematic approach to modeling traffic networks with the presence of ATIS, as online information is embedded in the definition. Furthermore, online information as defined above can handle a wide range of information accessibility situations, and thus avoids the usual simplified assumptions such as no information or full information. After online information is encapsulated in travelers' routing choices, it is rather natural to generalize conventional methods to be applicable in a traffic system with ATIS by replacing paths (non-adaptive routing decisions) with routing policies when necessary. This is not to say that the generalization is trivial. In fact, there are a lot of issues to be dealt with in the generalization and the details are in Chapters 5 and 6.

A policy-based stochastic dynamic traffic assignment (DTA) model is developed in this thesis to study the network-level impact of online information provision and adaptive routing. Key assumptions of this model are:

- Travelers take routing policies rather than paths.
- The traffic network is stochastic and time-dependent.

Input to the model is the joint distribution of stochastic time-dependent OD trips and supply, depending on the actual stochastic factors in the supply side, e.g. random incident locations and start times. Output from the model is the joint distribution of link travel times at equilibrium and equilibrium routing policy splits. A generalization of Waldrop's First Principle is used as the equilibrium condition: each user follows a routing policy with minimum perceived disutility at his/her departure time and no user can unilaterally change routing policies to improve his/her perceived disutility. A disutility function is just the opposite of a utility function in a conventional random utility discrete choice model. The systematic component of a utility function can be a linear combination of various routing criteria, such as expected travel time, travel time standard deviation, expected early/late schedule delay. Random errors are associated with a systematic utility function and thus the word "perceived" is used in the equilibrium condition.

A fixed-point formulation of the model can help understand the equilibrium condition. Assume travelers have the equilibrium distribution of link travel times as their *a priori* distribution based which optimal routing policies are chosen by travelers with some criteria and possible perception errors. Travelers then are loaded into the network with stochastic supply, and note that the number of travelers could also be stochastic. This loading will result in a distribution of link travel times. If this newly derived distribution is consistent with the one before, it is believed that an equilibrium is reached.

Because of the flexibility in defining online information in a routing policy, the policy-based stochastic DTA model can handle a wide range of online information accessibility situations. For example, some travelers might have advanced in-vehicle systems, some have radio broadcast, while others only have access to VMS. The heterogeneity in travelers' online information access can be handled by classifying travelers by their information access, i.e. definitions of routing policies.

Another noticeable feature of the problems studied in this thesis is the discretization of time. The distribution of link travel times is discrete, which is the support of a routing policy definition. Since all later algorithms and models are based on the notion of routing policy, they also require a discretization of time. This is not a restriction, since an arbitrarily fine resolution can be taken in the discretization to meet the requirements of applications. However, a continuous representation might lead to more efficient method, and is certainly of interest for future research.

1.3 Research Summary

There are two major parts in this thesis. First, optimal routing policy problems in stochastic time-dependent networks are studied, and then a policy-based stochastic dynamic traffic assignment model is developed, where users are assumed to take adaptive routing decisions.

1.3.1 Optimal Routing Policy Problems in Stochastic Time-Dependent Networks

A distinctive feature of a traffic network is the link-wise and time-wise stochastic dependency of link travel times. However, a comprehensive literature review on optimal routing policy (ORP) problems in stochastic time-dependent (STD) networks reveals that no research has considered this important feature of a traffic network. Nevertheless, research on ORP problems in simpler STD networks in the literature is helpful to this thesis. On the other hand, ORP problems in networks which are not explicitly time-dependent have been studied in more detail. These networks are either static or Markovian. Link-wise stochastic dependency of link travel times in static or Markovian networks is considered in quite a number of papers, among which the paper by Polychronopoulos and Tsitsiklis (1996) [41] gives inspiration for ORP problems in STD networks studied in this thesis.

In recognition that quite a few variants of ORP problems exist in the literature with various assumptions on network probabilistic properties and information access, differences and similarities among them are identified and a framework is established. Under the framework, an STD network is described by a joint distribution of link travel times and is known *a priori* by travelers. A traveler can be in many possible states during a trip, and he/she needs to make a routing decision for each state he/she is in. A state is defined as a triple (node, time, current information) where "current information" is a specific term in this thesis (different from information in general) and is determined jointly by online information access and whether part or all of the online information is useful in making inferences about the future. A detailed discussion of online information accessibility is in the previous section. The issue of whether some online information is useful is a more involved problem and is discussed in detail in the taxonomy (Section 2.2.2). A dynamic-programming type generic optimality condition for ORP problems in STD networks is presented, which will be specialized later to design an algorithm for the new variant defined in this thesis. Two key components have been identified in characterizing a variant: information access and network stochastic dependency, and thus a taxonomy is established based on them. Discussions on variants are available to provide examples on how the two components interact and how one can dominate over the other and vice versa.

A "support point" is defined as a distinct value that a discrete random variable can take or a distinct vector of values that a discrete random vector can take, depending on the context. Thus a probability mass function (PMF) of a random variable(vector) is a combination of support points and the associated probabilities.

Next two variants are studied in detail. The first is the one that has been studied in the literature, and is presented with computational tests to give an easy pass to understanding more complicated variants. Next a new variant that is particularly pertinent to traffic networks is defined and studied. It considers the link-wise and time-wise stochastic dependency of traffic link travel times, and the impact of online information on adaptive decision routing. The generic optimality condition is operationalized and an algorithm is designed. An algorithm worst case complexity analysis shows that the running time is polynomial in the number of support points of the joint distribution of link travel times, which could be exponential in the number of links and time periods. Thus there is need for good approximations. Several approximations are proposed, and their effectiveness against the exact algorithm studied, both theoretically and computationally.

When faced with uncertainty, travelers are also concerned about the reliability of their travel times. Travel time variance and expected schedule delay are used to represent travel time reliability. A routing policy with less travel time variance or less expected schedule delay is viewed as more reliable. Schedule delay is defined as the difference between the actual arrival time and the desired arrival time. For commuters, the desired arrival time in the morning might be some time around the work starting time. For a traveler to catch a plane, the desired arrival time might be roughly one hour before the plane departure. It is generally believed and verified by some empirical studies that both early and late arrivals cause disutility to the user.

Since expected travel time is the primary criterion in routing optimization, and reliability measures (variance or expected schedule delay) are generally secondary, it is not necessary to design algorithms that minimize variance or expected schedule delay only. Instead, an algorithm that minimizes a linear combination of expected travel time and expected early/late schedule delays is designed. It is found that Bellman's principle of optimality does not hold for the minimum variance policy problem, and thus no exact optimality conditions are available. Instead, an approximation is presented.

1.3.2 Policy-Based Stochastic Dynamic Traffic Assignment Model

A policy-based stochastic dynamic traffic assignment (DTA) model is the topic of the second part of this thesis. The major feature of the DTA model is that users are assumed to choose routing policies rather than paths, which takes into account the fact that users make adaptive decisions based on revealed network conditions. There is no DTA model in the literature with this feature.

A general framework is presented and the equilibrium problem is formulated as a fixed point problem. An intuitive interpretation of the fixed-point problem is available in the previous section.

A policy-size Logit model is developed for users' choice of policies, which is a counterpart of path-size Logit in path-based choice model. A Logit model assumes that choice utilities are independent, while in a route choice situation, paths or policies are usually overlapping and their travel times cannot be independent. The path-size or policy-size model accounts for the overlapping by multiplying the exponential of the utility with a factor, which is the path-size or policy-size.

The policy-based network loader is implemented as an iterative process over a traditional path-based dynamic network loader, which is a convenient approach, making use of the already developed path-based dynamic loaders readily available within the transportation research community.

An MSA heuristic is designed to solve the overall policy-based stochastic DTA model which is an integration of three components: the optimal routing policy generation module, the routing policy choice model and the policy-based dynamic network loader. There is no proof of convergence, and computational tests show that the MSA heuristic with reset works reasonably with the test network under the specific test settings. However, caution should be taken when we change the test network and settings.

The tests are designed with random incidents. Four equilibrium assignment models in STD networks – policy model, online path model, path model, and base model

– are implemented to be compared in the tests. They are distinguished by three features: the knowledge of the incident PMF, adaptiveness to online information, and optimal adaptive decision. A policy model has them all. An online path model is also an adaptive model, and the difference from a policy model is in how the adaptive routing decision is made. In a policy model, a user takes the next node computed by the dynamic-programming-type optimality condition, with the underlying assumption that the network conditions could change probabilistically and that he/she will have further diversion possibilities in the future. While in an online path model, the decision is more myopic. The next node is computed like this: a path with minimum expected travel time from the current node to the destination is computed based on the current information, and then the first link along this path is followed. When the user arrives at the next node, a new minimum expected travel time path is computed and the first link followed. Note that the new path is not necessarily a subpath of the previous one. This routing method is adaptive as a new path is computed each time a new decision node is reached, but it is myopic in the sense that it assumes no future changes in network conditions when computing the next node to take. The path model is not adaptive. It considers the random incident, but users follow simple paths instead of being adaptive. The base model is even simpler. It does not have the PMF of the incident. It is actually a conventional deterministic DTA working in a network with no incident.

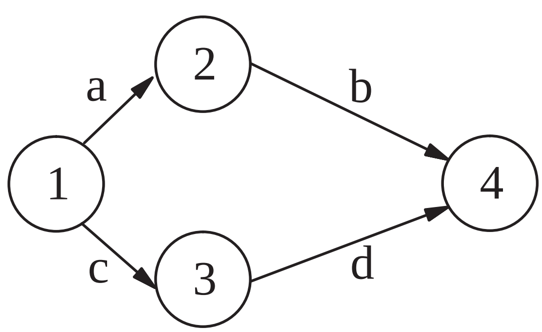
By comparing the policy model and online path model, the value of optimal usage of online information is obtained. By comparing the two adaptive models with the path model, the value of online information in obtained. By comparing the path model and the base model, the value of *a priori* knowledge on incident distribution is obtained.

Throughout a number of tests, the adaptiveness leads to travel time savings. As a byproduct, travel time reliability is also increased, as the travel time peak caused by a random incident is cut.

Sensitivity analysis shows that the value of online information is an increasing function of incident probability for specific demand scale and network topology. This result should be generalized with caution. It is also shown that online information penetration plays an important role in travel time savings. The savings are high when market penetration is low. The function of travel time savings against penetration is not monotonic. This suggests that when implementing a traveler information system, one needs to consider the penetration of information to maximize benefits.

1.4 Path vs. Routing Policy

The focus of our discussion has been on routing policy. Loosely speaking, routing in an STD network can take one of the two forms: a path and a routing policy. A path is a pre-specified set of successive links between a pair of nodes. Travelers who follow a path make decisions *a priori* and take a fixed set of links, regardless of the network conditions. In contrast, travelers who follow a routing policy make decisions online and therefore can end up taking different sets of links, depending on the network conditions that have been revealed during their trip. Mathematical definitions of these terms can be found in Chapter 2. In this section, we use two illustrative examples (Figure 1-1 and Figure 1-2) to show the difference between a path and a routing policy.



$$(t_a, t_b, t_c, t_d) = \begin{cases} (5, M, 1, 9) & w.p.\ 0.5 \\ (1, 6, 4, M) & w.p.\ 0.5 \end{cases}, \text{ where } M \text{ is a very large positive number}$$

Figure 1-1: Paths vs. Routing Policies in a Non-Time-Dependent and Statistically Dependent Network

In Figure 1-1, we seek a route from node 1 to node 4. Two paths, *a*-*b* and *c*-*d*, are available from node 1 to node 4. The network is stochastic but not time-dependent. The data shown beneath the network shows the distribution of travel times for different parts of the network. From this data, we can see that one of the paths will be blocked at a given time. For instance, link b can have a very large travel time M with probability 0.5. Assume that one can learn the actual realization of travel times on links a and c when one arrives at node 1.

We can compute the expected travel times of path $a - b$ and path $c - d$. The expected travel time of path *a*-*b* is $(5 + M) \times 0.5 + (1 + 6) \times 0.5 = 6 + M/2$, and that of path *c*-*d* is $(1 + 9) \times 0.5 + (4 + M) \times 0.5 = 7 + M/2$. If the routing model from node 1 to node 4 identifies least expected travel time paths, one will always choose path *a*-*b*. However, one can do better if the information available at node 1 is effectively used. If the routing model identified least expected travel time routing policies, we will do the following: when the travel time of link a is 5, we choose path *c*-*d*; otherwise we choose path *a*-*b*. The expected O-D travel time of the policy is $(1+9) \times 0.5 + (1+6) \times 0.5 = 8.5$. An optimal routing policy defers the decision until some useful information is collected. In this example, the decision is delayed until one knows which path is blocked.

This example shows the usefulness of information in a stochastic routing problem. The value of information in the example is due to the stochastic dependency of link travel times. When we learn the realizations of some of the links, we can make inferences about travel times on other links so that better decisions can be made. There are cases where the link travel times are not statistically dependent, but their time-dependency makes information valuable. The example in Figure 1-2 shows such a case.

Data next to each link shows the probability mass function (PMF) of the link travel time for that link at each departure time of interest. Let us assume in this example that the link travel time random variables are statistically independent. We also assume that the arrival time at the current node is the only online information available to the traveler at that node. This implies that the traveler does not know

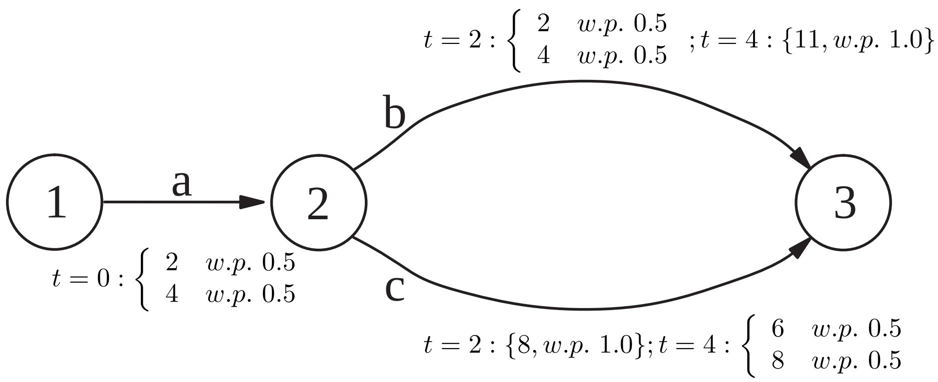


Figure 1-2: Paths vs. Routing Policies in a Time-Dependent and Statistically Independent Network

the actual realizations of link b or link c at node 2. This assumption is made in order to show that the knowledge of arrival times can also benefit the routing decision-making in an STD network. We seek a route from node 1 to node 3 for departure time 0. The least expected travel time path is path a-b, with an expected travel time of $(2+2) \times 0.25 + (2+4) \times 0.25 + (4+11) \times 0.5 = 10$, while the expected travel time of path a-c is $(2+8) \times 0.5 + (4+6) \times 0.25 + (4+8) \times 0.25 = 10.5$.

Let us consider the following routing policy: if the arrival time at node 2 is 2, take link b; if the arrival time at node 2 is 4, take link c. The expected travel time of the routing policy is $(2+2) \times 0.25 + (2+4) \times 0.25 + (4+6) \times 0.25 + (4+48) \times 0.25 = 8$. The decision in this routing policy is delayed until the arrival time at a decision node is known.

An intuitive representation of the routing policy, denoted as a "state network", is shown in Figure 1-3. We use the pair (j, t) to denote the network state, based on which the routing decision is made. j is a node and t is a time point. The traveler starts from $(1, 0)$, and the decision is to go to node 2. At node 2, two situations $(2, 2)$ or $(2, 4)$ are possible. With $(2, 2)$, the traveler chooses link a, and could end up at either of the following two situations: $(3, 4)$ or $(3, 6)$. Similarly with $(2, 4)$, the traveler chooses link b, and could end up at two situations: $(3, 10)$ and $(3, 12)$.

From the above two examples, we can see that a routing policy generally involves

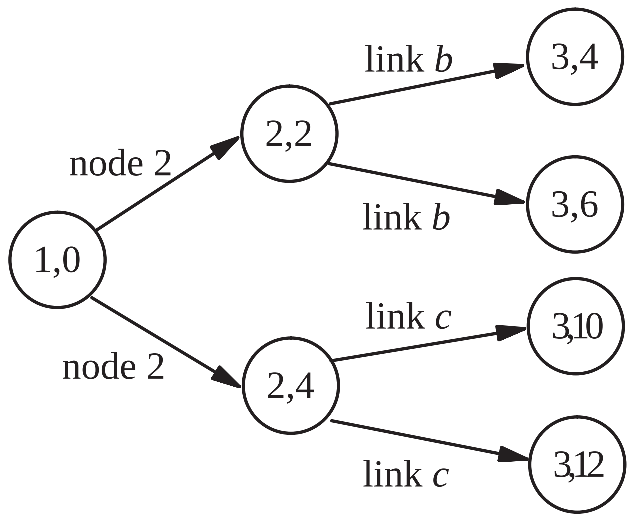


Figure 1-3: The Optimal Routing Policy for the Example in Figure 1-2

more than one path. Which path is taken depends on the network conditions, i.e. link travel time realizations and/or the arrival times. When one is at a given node, the least expected travel time path does not exploit the possible information collected during the trip, and thus is generally less effective than an optimal routing policy. The value of information is either due to the stochastic dependency or due to the time dependency of stochastic link travel times. We can also construct examples where the value of information is due to both the stochastic dependency and the time dependency of the link travel times. Various assumptions can be made about the stochastic dependency of link travel times and the degree of knowledge one would have about available link travel times. These factors lead to numerous variants of the ORP problem in an STD network. A limited number of these variants have been studied in the literature, and some will be explored in this thesis for the first time.

1.5 Thesis Contributions

The contributions of the thesis to the knowledge base of routing problems in stochastic time-dependent networks are summarized as follows:

1. Establishes the first framework for ORP problems in STD networks. Various assumptions have been made in the literature to define routing problems in stochastic networks. This thesis identifies the similarities among these variants and establishes a framework for a unifying understanding of the problem. A comprehensive taxonomy is provided, based on information access and network stochastic dependency. It contains all variants in the literature and can lead to new variants suitable to various applications.

2. Extends the concept of routing policies. The study of routing policies in STD networks in the literature has been restricted to policies based on arrival times at decision nodes. This thesis recognizes the role of information in routing decision making, and includes information as an integral part of a routing policy.

3. Models stochastic dependency of link travel times and designs a solution algorithm. Traffic networks are generally statistically dependent both link-wise and time-wise, yet no papers in the literature have addressed this problem in STD networks.

4. Identifies the importance of designing good approximation algorithms for ORP problems. This thesis provides four possible approximation algorithms and studies their effectiveness both theoretically and computationally. These studies are the first step to designing time-efficient routing policy algorithms for real-time traffic applications.

The contributions of this thesis to the knowledge base of dynamic traffic assignment are summarized as follows:

1. Establishes the first dynamic user equilibrium traffic assignment model where users' choices are routing policies. Since the definition of the routing policy is very general, as defined in the first part, the corresponding DTA model is also general, in terms of the various assumptions one can make about the network stochastic dependency and information access.

2. Considers full stochasticity in the DTA model, including the random perception errors in the users' policy choice model, the random origin-destination trip rates, and the random supply, such as incident, bad weather, etc. The output from the model is the distribution of link travel times and other traffic quantities in interest.

1.6 Thesis Organization

The thesis is organized as follows. We study the stochastic routing problem first. We propose exact algorithms and approximations. We then design a policy-based stochastic dynamic traffic assignment model.

In Chapter 2, we study the optimal routing policy problem in a stochastic time-dependent network. We first survey the literature on this topic, including deterministic routing problems, routing problems in stochastic static networks and routing in stochastic time-dependent networks. This survey reveals that there are a number of variants of the ORP problem in an STD network, and it motivates the establishment of a formal framework for this problem. We then proceed to describe the framework which includes a general description of an STD network, the decision process, the problem statement and the generic optimality conditions. We then present a comprehensive taxonomy based on assumptions on the network stochastic dependency and information access. A discussion of most of the variants is given, and two variants are studied in detail. The first is the no-online-information (NOI) variant which is easy to understand and can be solved in polynomial time. We give the formulation, an algorithm and computational tests for this variant. The second variant studied in detail is the perfect-on-line (POI) information variant, which is particularly pertinent to transportation applications. This variant has never been studied before in the literature. We give a formulation, an algorithm, and results from computational tests.

The complexity analysis shows that the algorithm for the POI variant can be prohibitively time-consuming. Therefore approximations are also developed in Chapter 3. Four approximations are presented with an analysis of their efficiency and effectiveness. This analysis is done both theoretically and computationally. The computational tests are not comprehensive, but they provide insight into the performance of approximations. Other approximations are suggested, however without computational tests.

Chapter 4 extends the framework and algorithms developed in Chapter 2 such that the optimization criteria are minimum variance and minimum expected schedule delay respectively. This extension is motivated by the serious concern about travel time reliability in a stochastic network. Problem formulations and solution algorithms are presented. For the minimum expected schedule delay policy problem, exact solution algorithms exist. However for the minimum variance policy problem, it is shown through counterexample that Bellman's principle of optimality does not hold. Therefore an approximate algorithm for the problem is proposed instead.

In Chapter 5, we develop a policy-based stochastic dynamic traffic assignment model. The model differs fundamentally from traditional ones, in the sense that it is based on routing policies, rather than paths, to explicitly model the adaptive choices in a stochastic network. An illustrative example is used to motivate the study and to show the characteristics of a policy-based traffic assignment model. We then present a users' policy choice model and a dynamic traffic network loading model; all are based on routing policies. A stochastic DTA heuristic is then proposed using the method of successive averages (MSA).

In Chapter 6, we presented the computational tests of routing policy based DTA models in a hypothetical network, where random incidents are the source of the stochasticity. We study the convergence property of the MSA heuristic for the policy-based stochastic DTA model. We then study four different DTA models in stochastic networks in terms of the expected travel time, travel time variance, and travel time distribution. These models differ in the information available and routing choices assumed for the users: 1) no information about incidents and deterministic shortest paths; 2) *a priori* information about probabilistic incident parameters and *a priori* minimum expected travel time paths; 3) online information about link travel time realizations and online minimum expected travel time paths; and 4) online information about link travel time realizations and optimal routing policies. The value of *a priori* information and that of online information at stochastic equilibrium are studied rigorously by examining results from these models. It is shown that market penetration and incident probability affect the sign and the magnitude of the values of information.

In Chapter 7, we give a summary of the thesis work and findings and discuss future directions of research.

Chapter 2

Optimal Routing Policy Problems in Stochastic Time-Dependent Networks

In this chapter, we study optimal routing policy problems in stochastic time-dependent networks. We first provide a literature review on a broad range of network routing problems. We then establish a framework, in order to provide a unified view of this problem, considering the large variety of variants already in the literature and the numerous possibilities of new variants. This framework includes a general description of a stochastic time-dependent network, the decision process in a stochastic time-dependent network, the minimization problem, and the optimality conditions. Following this somewhat abstract framework, we give a comprehensive taxonomy based on two criteria: network stochastic dependency and information access. We discuss the variants within the taxonomy, and pay special attention to the variants already in the literature to see how they fit into the framework. This discussion may still seem abstract, as no algorithms are given at this point. We suggest that the reader come back to this part after he/she finishes the following algorithmic parts. We study two variants in detail after the framework and taxonomy. The first is the no-information variant which is not very realistic in traffic settings, but has a very good running time and is relatively easy to understand. We study it first to provide some background knowledge for the more complicated variants. Next we study the perfect-online-information variant where the dependency of traffic variables is considered. Both variants are studied in a formal way: the formulation, algorithm, implementation, complexity analysis, and computational tests are presented in sequence. The study of the perfect-online-information variant also suggests the need for good approximations to the exact algorithm, which is the topic of the next chapter.

2.1 Literature Review

Routing problems in networks have been an important and well researched topic for a long time. We first give a brief introduction to the shortest path problem in deterministic networks, including the well developed static shortest path (SSP) problem and the dynamic shortest path problem. This will be useful to the study of routing problems in stochastic networks. We then proceed to stochastic networks. There are various ways of defining a stochastic network, and this results in a variety of variants of the ORP problem. Most of the problem variants studied in the literature assume that the underlying network is static (not dependent on time). Some other variants studied in the literature work with special cases of dynamic stochastic networks. They do not represent time explicitly. A limited number of papers have studied the ORP problem in a STD network with specific assumptions. A comprehensive study of the problem is not available in the literature.

2.1.1 Deterministic Routing Problems

Compared to routing in STD networks, the classical static shortest path (SSP) problem has been extensively studied. Let $G(N, A)$ be a network, where N is the set of nodes and A is the set of links. Each link (i, j) has a cost $c(i, j)$ and we term a path with minimum cost as a shortest path. The SSP is to find a shortest path from a source node s to a destination node d. Dijkstra's algorithm is the most commonly used algorithm to solve the shortest path problem for networks with non-negative arc costs. Various implementations of Dijkstra's algorithm exist. The most straightforward one is based on the array data structure and has a running time of $O(n^2)$, where n is the number of nodes. The implementation using a binary heap can achieve a running time of $O(m \ln n)$, where m is the number of arcs. If the network has negative arc costs, more sophisticated algorithms (such as the label-correcting algorithms) are needed. These algorithms basically check whether the optimality conditions $d(i) + c(i,j) \geq d(j), \forall (i,j) \in A$ are satisfied, where the label $d(i)$ is the cost for the origin to node i. They make necessary changes by changing cost labels until no arc violates this condition. A first-in-first-out (FIFO) queue implementation of the label correcting algorithm has a running time of $O(mn)$.

The dynamic shortest path problem arises when it is required to model a transportation network in which travel times change significantly as a function of time. Let $G(N, A, T)$ be a dynamic network, with T the set of time periods. We define a dynamic shortest path as a path with minimum travel time in a dynamic network. (The algorithms for minimum time paths can be extended to compute minimum cost paths with minor changes.) At each time period t, each link (i,j) has a travel time $c_{ij}^t > 0$. The dynamic shortest path problem is to find a shortest path from a given source node s to a destination node d for a given departure time t at node s. We define a link (i,j) as FIFO, iff $t_1 + c_{ij}^{t_1} \geq t_2 + c_{ij}^{t_2}, \forall t_1, t_2 \in T$ and $t_1 > t_2$. A network is FIFO, iff all links are FIFO. When a dynamic network is FIFO, we can apply a Dijkstra-like algorithm to solve the dynamic shortest path problem. When the network is non-FIFO, generally a label-correcting type of algorithm would be able to solve the problem. Chabini (1999) [17] presents an algorithm called DOT with an optimal running time $\theta(SSP + mK + nK)$ to solve the dynamic shortest path problem with positive travel times from all nodes at all departure times to one destination node, where SSP is the running time of a static shortest path algorithm and K is the number of time periods. Algorithm DOT is optimal in the sense that no algorithm with better theoretical running time exists. Algorithm DOT sets the labels in decreasing order of time, based on the fact that the distance label of a node at a given time can only be updated by labels of later times. This algorithm is the base of the algorithms we develop for the stochastic routing problem.

2.1.2 Routing in Stochastic Static Networks

Loosely speaking, a stochastic network is a network where the link travel times are random variables with some *a priori* distributions. If the underlying network is assumed to be static (non-time-dependent), the link travel times remain unchanged after they are revealed to the travelers. In a time-dependent network, on the other hand, the travel time of every link at every time period is an individual random variable, so travel times revealed at different time periods could be different. The study of ORP problems in static networks is useful to the study of its time-dependent counterpart.

Andreatta and Romeo (1988) [5] study the problem in a static network where the topology is stochastic. A stochastic topology is defined by a deterministic set of nodes N and a random set of links $A \in N \times N$. Each possible topology has a positive probability. A random link can be either active or not. When it is active, it is included in the network; when it is not active, it is removed from the network. The decision maker (DM) can learn whether a link is active or not once he/she reaches the node from which the link emanates. The DM can reroute once he/she finds out the next link is inactive. The notion "stochastic shortest path" is used, yet actually a routing policy problem is studied. The path without recourse actions in a routing policy (i.e. the path composed solely of nodes representing "active" scenarios in the state network of a routing policy) is used to denote that policy, so a stochastic shortest path in this paper is actually a least expected cost routing policy. Andreatta and Romeo (1988) [5] proves four facts about a stochastic shortest path that are different from those about a deterministic shortest path. A stochastic dynamic programming formulation of the problem is provided, with the definition of "information state" which gives the active/inactive links of the network revealed to the decision maker so far and based which the recourse decision is made. It is pointed out that the complexity of the algorithm can grow exponentially with the number of links. Therefore a restricted version of the problem is studied and it is shown that polynomial algorithms exist for this particular case.

Polychronopoulos and Tsitsiklis (1996) [41] extend the work of Andreatta and Romeo (1988) [5]. They study the problem both in networks with link travel times that are correlated and in networks with independent link travel times. For the dependent case, a joint distribution of link travel times is used to represent the stochastic network. We can see that the stochastic topology in Andreatta and Romeo (1988) [5] is actually one special form of joint distribution of link travel times. It is assumed that the travel time realizations of outgoing links of a given node are known and remembered by the traveler once he/she arrives at this node, and the realizations remain unchanged afterwards. As the traveler moves on the network from the origin to the destination, more link travel time realizations are learned, and the network becomes closer to a deterministic one. The concept of information set is introduced to represent the traveler's knowledge about the network. An information set is composed of support points that are consistent with the link travel times revealed so far. When the information set becomes a singleton, the network becomes deterministic. A dynamic programming approach is presented where the stage of dynamic programming is labeled by the cardinality of the information set, starting from the smallest. Some of the main concepts in this thesis originate from Polychronopoulos and Tsitsiklis (1996) [41]. A similar approach is designed for the independent case, with changes in the manner in which the information set is defined. The algorithms, however, have exponential running times: the algorithm for the dependent case has running time exponential in the number of support points, and the algorithm for the independent case exponential in the number of links. It is proved that the problem with dependent link travel times is NP-complete, and that with independent link travel times is NP-hard. Some heuristics are given and the relationships between results from heuristics and exact algorithms are studied.

Cheung (1998) [19] studies the problem with the same independent network assumptions as those in Polychronopoulos and Tsitsiklis (1996) [41], except the assumption that two visits to the same node result in two independent realizations of outgoing link travel times. This assumption (which is termed as "reset" later by Provan (2003) [43]) actually makes ambiguous the statement that the network is static, as the same link can take different travel times at different times, although the distribution is the same. On the other hand, the reset assumption makes possible a simple recursive equation for the expected minimum travel times. An approach that mimics the classical label-correcting algorithm is presented. Computational tests are carried out to compare different implementations of the label-correcting approach. Provan (2003) [43] studies the same problem as defined by Cheung (1998) [19] with the extension that the link travel times can be dependent. However, this relaxation from independent to dependent networks does not make the problem harder. In fact, the reset assumption makes the term "dependent" less clear, as one can never make inferences about travel times on links other than those going out of the current node. The same recursive equation is presented, but a polynomial-time algorithm is designed and its complexity analyzed. The author also gives a good classification of ORP problems in time-independent networks, and shows the relationship between them.

2.1.3 Routing in Stochastic Time-Dependent Networks

Hall (1986) [29] studies for the first time the time-dependent version of the optimal routing policy problem. The problem is studied within the context of transit networks. It is shown that in a stochastic time-dependent network, adaptive route choices (routing policies) are more effective than simple paths. A dynamic programming approach is provided, in which the stages of the dynamic program are the number of links from the destination node. An upper bound k on the number of stages is specified, and it is stated that when k is sufficiently large, the solution should be very close to the optimum. The recursive equations for the dynamic program are given and it is implicitly assumed in the equations that routing policies are based only on arrival times at decision nodes. We note that with this implicit assumption, k can be set to be the sum of number of time periods and number of nodes to guarantee the optimality of the solution.

The assumption that routing policies only depend on arrival times at decision nodes is also made by Chabini (2000) [18]. A dynamic programming algorithm is developed in which the stage of the program is the time period t, based on the concept of decreasing order of time that is also used in developing Algorithm DOT by Chabini (1999) [17]. This formulation of the problem enables an optimal algorithm DOT-S with a complexity of $\theta(SSP+nK+mKQ)$, where Q is the maximum number of support points for a single link travel time discrete distribution. This algorithm is optimal in the sense that no algorithms with better theoretical complexity exist. The algorithm is extended to solve the minimum expected travel cost routing policy problem with minor changes. Computational tests are carried out to study the running times of Algorithm DOT-S and the label-correcting algorithm developed by Miller-Hooks and Mahmassani (2000) [35]. It is concluded that algorithm DOT-S is computationally efficient both in theory and in practice.

Miller-Hooks and Mahmassani (2000) [35] study the ORP problem assuming time-wise and link-wise statistically independent link travel time random variables. This assumption leads to routing policies based only on arrival times at decision nodes. A label-correcting algorithm is developed to solve the problem. The label-correcting algorithm has a rather high worst-case running time, but its practical performance is good. Miller-Hooks (2001) [36] compares the label-correcting algorithm presented in Miller-Hooks and Mahmassani (2000) [35] and the dynamic programming algorithm working in decreasing order of time by Chabini (2000) [18] in both sparse transportation networks and dense telecommunication data networks. It is shown that the label-correcting algorithm has an empirical running time much better than its worst-case theoretical complexity. It is also concluded that in dense networks, the label-correcting algorithm is more computationally efficient than Algorithm DOT-S. This conclusion is not consistent with that obtained in Chabini (2000) [18], and further research is desired. Recently, Yang and Miller-Hooks (2004) [49] also extend the study of the time-adaptive routing policies to a signalized network.

Bander and White (2002) [8] study the same time-adaptive routing policy problem, i.e. the link travel time random variables are time-dependent, but statistically independent from each other, and thus the decision is dependent on the (time, node) pair. The major contribution of this paper is the design of a heuristic approach with a very promising feature: it will terminate with an optimal solution if one exists, given that the heuristic function underestimates the true cost-to-go. The proposed heuristic has a significant computational advantage compared to dynamic programming, shown through computational tests.

Fu (2001)[25] studied a variant where travel time realizations of outgoing links of the current node are available to the vehicle before it enters the link. Independent link travel time random variables are assumed implicitly, according to the formulation of the problem. An efficient probabilistic approximation is proposed to solve the formulated problem and the advantage of adaptive routing systems is shown. However, this paper does not model time explicitly. The dynamics of the system are represented through the fact that predictions of link times change over time (and are conceivably more accurate when the vehicle is closer to a link). Yet for the current available link time predictions, the problem is indeed non-time-dependent.

We also briefly review literature on the least expected time path problem in a STD network, as it is closely related to the ORP problem. Fu and Rilett (1998) [26] model link travel times as a continuous-time stochastic process. It is assumed that travel times on individual links at a particular point in time are statistically independent, and correlations between link travel times are modeled through the time-dependency of link travel time distributions. Relationships between the mean and variance of the travel time of a given path and the mean and variance of link travel times on that path are identified. A heuristic is designed recognizing the computational intractability of the problem. Miller-Hooks and Mahmassani (2000) [35] study the least expected time path problem under the same assumptions for the ORP problem. They establish a dominance rule for paths in STD networks and design a label-correcting type of algorithm. The worst-case complexity of the algorithm is exponential as a function of the network size, but computational tests on sparse transportation networks show the actual performance is practically linear with respect to the network size.

Some researchers studied the ORP problem variants with stationary Markovian link costs. Polychronopoulos (1992) [40] assumes global information access and defines a combination of travel times of all links as a state. It is further assumed that the transition probability matrix is available, and that the occurrence of a transition in unit time is related to the network conditions. A dynamic programming formulation of the problem is suggested and it is claimed that any standard Markov decision algorithm can solve the problem. Psaraftis and Tsitsiklis (1993) [44] assume travel times of outgoing links of a given node are functions of condition at this node, which evolves as a Markovian chain. Markovian chains at different nodes are assumed to be independent and the network is acyclic. Vehicles can wait at a node (at a cost) in anticipation of a more favorable arc cost. Three different types of algorithm are developed to solve the case of a single arc network: successive approximation (SA), policy iteration (PI), and parametric linear programming. A dynamic programming approach is then developed, making use of the algorithm for a single arc. The algorithm is shown to be polynomial, due to the assumptions of acyclic networks and stationary Markovian costs independent across nodes.

2.2 Framework and Taxonomy

2.2.1 Framework

From the literature review, we find that there is no formal definition of the routing problem in a stochastic time-dependent network. Various assumptions are made to define a stochastic network and to define how the realizations of the stochastic network are revealed to the travelers (decision makers). For example, in Andreatta and Romeo (1988) [5], the topology of the network is stochastic; in Polychronopoulos and Tsitsiklis (1996) [41], the whole static network is described by a joint distribution of link travel costs in the dependent case, and by marginal distributions of link travel times in the independent case; in Polychronopoulos (1992) [40] and Psaraftis and Tsitsiklis (1993) [44], the link costs evolve as Markov chains; in Hall (1988) [29], Chabini (2000) [18], Miller-Hooks and Mahamassani (2000) [35], Miller-Hooks (2001) [36], Yang and Miller-Hooks (2004) [49], and Bander and White (2002) [8], time-dependent networks are described by marginal distributions of link travel times. As for the revealing of the stochastic network, some assume that one learns the realization of a link travel cost once he/she arrives at the node from which the link emanates (Andreatta and Romeo (1988) [5], Polychronopoulos and Tsitsiklis (1996) [41], Cheung (1998) [19], Provan (2003) [43]), while others do not state explicitly how travelers learn about the network conditions other than the arrival times at decision nodes (Psaraftis and Tsitsiklis (1993) [44], Chabini (2000) [18], Miller-Hooks and Mahamsssani (2000) [35], Miller-Hooks (2001) [36], Yang and Miller-Hooks (2004) [49], Bander and White (2002) [8], Pretolani (2000) [42]). Therefore the routing policies are time-adaptive, i.e. dependent on the (time, node) pair. However, we can generalize these various descriptions and assumptions. We also realize that the routing process in a stochastic network is a mapping from some knowledge of the network to a decision, and whether what knowledge is available and/or useful depends on specific assumptions about the network and the information access. A general set of optimality conditions is then possible with the formal definitions of the problem.

We establish the framework to provide a unified view of the optimal routing policy problem in a stochastic time-dependent network. We will be able to see the connections among various variants in the literature with the aid of the framework, and generate new variants that are required by specific applications. The generic optimality conditions can provide a general way of designing solution algorithms for variants of the problem.

The Network

Let $G = (N, A, T, P)$ be a **stochastic time-dependent network**. N is the set of nodes and A is the set of links. The number of nodes and links are denoted respectively as $|N| = n$ and $|A| = m$. The network has a single destination node d. T is the set of time periods $\{0, 1, ..., K-1\}$. Travel time on each link (j, k) during each time period t is a random variable $\tilde{C}_{jk,t}$ with finite number of discrete, positive and integral support points. Throughout the thesis, a symbol with a $\sim$ over it is a random variable, while the same symbol without the $\sim$ is one specific value of the random variable. Beyond time period $K-1$, travel times are static and deterministic, i.e. the travel time of link (j, k) at any time $t \geq K - 1$ is equal to $C_{jk,K-1}$.

P is the probabilistic description of link travel times. Throughout the thesis, we assume that travelers know the probabilistic description *a priori*. Different descriptions exist because of different assumptions about network statistics. The most general one is in the form of the joint probability distribution of all the link travel time random variables, which is described next. Let $P = \{v_1, v_2, ..., v_R\}$ be the set of support points of the link travel time distribution. The rth support point has a probability p_r, and $\sum_{r=1}^{R} p_r = 1$. $C^r_{jk,t}$ is the travel time on link (j, k) at time t for the rth support point. Note that we assume the underlying topology of the network is deterministic, as a network with stochastic topology can be transformed into a network with deterministic topology where a blocked (or missing/inactive) link is modeled by setting its travel time to infinity (or computationally, a very large positive number).

We will use an example to show how the joint distribution description works. Figure 2-1 shows a small network with three nodes, three links and three time periods. All possible values of travel times are in Table 2.1. Each of the eight support points has a probability of 0.125. The network is designed to be very small to make understanding the concept easier.

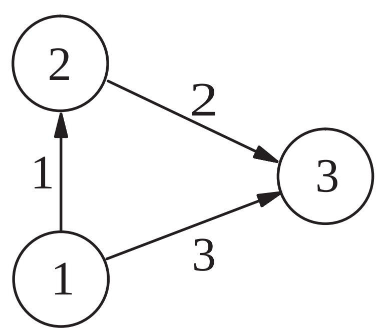


Figure 2-1: A Small Network

The joint distribution description of the link travel times can be specialized to other descriptions, depending on the assumptions about the network stochastic dependency. If all the link travel time random variables are statistically independent, both link-wise and time-wise, we could still use the joint distribution description. However, it would be much more efficient to keep only the marginal distributions of

Time	Link	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
0	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1
	3	1	1	1	4	4	4	3	3
1	1	1	1	1	1	1	1	1	1
	2	2	2	1	2	2	1	2	1
	3	3	3	2	2	2	1	3	2
$t \geq 2$	1	1	1	2	1	1	1	2	2
	2	1	2	1	1	1	1	2	1
	3	3	2	2	3	4	3	5	2
p_i		0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125

Table 2.1: Joint Distribution for the Small Network

link travel times. If we assume all links can be divided into several groups and link travel times in a given group are independent of link travel times outside the group, we need keep only the joint distribution of link travel times in each group. The same grouping can also be done along the time dimension, or along both the time dimension and link dimension.

The Decision Process

Throughout the thesis, we assume the traveler knows *a priori* the probabilistic description P of the network. Assume the traveler can make decisions only at nodes. The decision is what node k to take next, based on the *current state* $x = \{j, t, I\}$, where j is the *current node*, t is the *current time*, and I is the *current information*. current information I is defined as a set of available realized link travel times at the current time and current node that are useful for making inferences about future link travel times. It represents the traveler's knowledge about the network conditions. This knowledge could be dependent on time, location of the traveler, mode of transportation, etc. current information I therefore should be regarded as $I(j, t)$, but we usually use I only since I is always associated with a state where j and t are well defined. More discussion about current information can be found in the next subsection about taxonomy.

An ideal case is that travelers have perfect information about the whole network, but generally the information is local, as shown in the example of Figure 1-1, where one learns the travel time realization of a link when he/she arrives at the node from which the link emanates. In this example, the current information would be the combination of link travel time realizations of links a and c. The decision at node 1 can then be described as: when current state is $\{1, t, (2, 1)\}$, take node 3 next; when current state is $\{1, t, (1, 3)\}$, take node 2 next, for all t. Note that "current information" is one component of a current state and refers to link travel time realizations based on which the current decision is made, while a reference to "information" alone is in the general sense. One can be in many different states traveling in the stochastic time-dependent network, and we have the following definition.

Definition 2.2.1 (Routing Policy). *A routing policy $\mu(x)$ is a mapping from states to decisions (next nodes specifically in networks).*

This definition indicates that the routing decision in a stochastic time-dependent network is far from being set *a priori*. Rather, it is closely related to the network conditions, and this notion is critical in any ITS application.

We look ahead after making the decision at the current state. We do assume that the outcome of the decision is certain, i.e. the traveler will end up arriving at node k if he/she chooses it. The next state $y = \{k, \tilde{t}', \tilde{I}'\}$ the traveler will occupy is uncertain, i.e. $\tilde{t}'$ and $\tilde{I}'$ are random variables. The travel time of link (j, k) at time t conditional on I could be uncertain, resulting in an uncertain arrival time $\tilde{t}'$ at node k. The next current information $\tilde{I}'$ is also uncertain, as $\tilde{t}'$ itself is uncertain. Even if $\tilde{t}'$ is deterministic, link travel time realizations from t to $\tilde{t}'$ could take multiple values, as the network is still stochastic to the traveler at the current state. However, for a given current state and a given decision, probabilities of all possible next states can be evaluated from the network statistics P.

Definition 2.2.2 (State Chain). *Define a state chain $\{x_0, x_1, ..., x_S\}$ as the series of states a traveler experiences during the trip, where x_S is a state with the destination node d as its current node.*

Current nodes of a state chain form a path, and S is the number of links in the path. With a given initial state x_0 and a routing policy μ, one could experience multiple state chains. For example, the routing policies in Figures 1-1 and 1-2 involve more than one path. As shown in the visualization of a routing policy in Figure 1-3, a routing policy with an initial state can be visualized as a **state network**. In this state network, a node is a state and outgoing links of a node are the decisions based on that state. The succeeding nodes stand for the possible next states the traveler will be in.

Definition 2.2.3. *Denote the set of possible state chains for a given initial state x_0 and a given policy μ as $M(x_0, \mu)$, then the state network is a representation of $M(x_0, \mu)$.*

We use the small network example in Figure 2-1 and Table 2.1 to show the decision process. We assume the traveler has knowledge of all the link travel time realizations up to the current time, regardless of his/her current node. This assumption implies the following: at time 0, the traveler knows what travel time values links 1, 2 and 3 take as of time 0; at time 1, the traveler knows what travel time values links 1, 2 and 3 take as of times 0 and 1; at time 2, the traveler knows what travel time values links 1, 2 and 3 take as of times 0, 1 and 2; etc. Therefore the current information I at time t would be one of the joint realizations of $C_{1,0}$, $C_{2,0}$, $C_{3,0}$, ..., $C_{1,t}$, $C_{2,t}$, and $C_{3,t}$. Note that we use a single subscript to denote a link rather than a pair of node numbers, for the sake of simplicity.

Let us now go through the state network step by step. The initial state is $\{1, 0, (1, 1, 4)\}$. We can see from the joint distribution table that the network could be in v_4, v_5, or v_6. As travel time on link 3 at time 0 is greater than 3, the traveler chooses node 2 as the next node. When he/she arrives at node 2, he/she could be in two possible states. One is $y = \{2, t', I'\} = \{2, 1, (1, 1, 4, 1, 2, 2)\}$ as represented by the upper of the two nodes succeeding node 2 in the state network. The other is $y = \{2, t', I'\} = \{2, 1, (1, 1, 4, 1, 1, 1)\}$ as represented by the lower of the two nodes succeeding node 2. The only choice at node 2 is node 3, and the traveler arrives at the

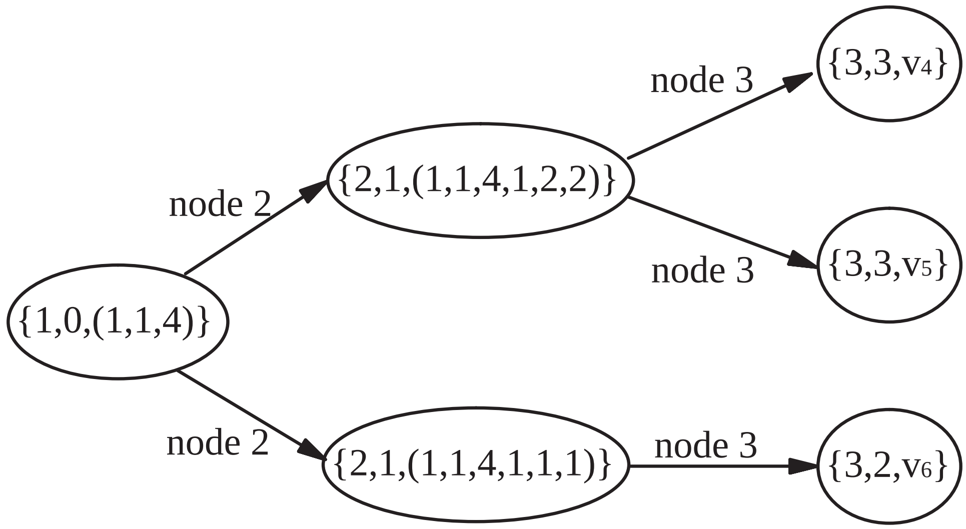


Figure 2-2: The Decision Tree of the Naive Routing Policy

destination (node 3). However, the ending states could be different. From the state $\{2, 1, (1, 1, 4, 1, 2, 2)\}$, the traveler could end up at state $\{3, 3, v_4\}$ or state $\{3, 3, v_5\}$. From the state $\{2, 1, (1, 1, 4, 1, 1, 1)\}$, the traveler could end up at state $\{3, 2, v_6\}$.

There are a total of three state chains in this state network. Note that the arrival times at the destination can be different for each state chain. For all state chains, the arrival time at node 2 is 1, as the travel time on link 1 at time 0 is 1. For the upper two state chains, however, the link travel time of link 2 at time 1 is 2, so the arrival time at node 3 is $3(= 1 + 2)$. For the lower state chain, the link travel time of link 2 at time 1 is 1, so the arrival time at node 3 is $2(= 1 + 1)$.

We see that for a given routing policy and a given initial state, the O-D travel time is a random variable. For example, the O-D travel time as shown in the state network of Figure 2-2 is a random variable with two possible values: 2 and 3. The probability that the O-D travel time is realized as 3 is the probability the state chain is one of the upper two chains, which is the probability that travel time realizations for all links at time 1 is $(1, 2, 2)$. Note that this probability should be evaluated conditional on the fact the link travel time realizations for all links at time 0 is $(1, 1, 4)$. Therefore the probability in question is

$$\frac{p_4 + p_5}{p_4 + p_5 + p_6} = \frac{0.125 + 0.125}{0.125 + 0.125 + 0.125} = \frac{2}{3}.$$

Similarly, the probability that the O-D travel time is realized as 2 is 1/3. Therefore the expected O-D travel time for the routing policy with the given initial state as in Figure 2-2 is $3 \times 2/3 + 2 \times 1/3 = 8/3$, and the variance is $(3 - 8/3)^2 \times 2/3 + (2 - 8/3)^2 \times 1/3 = 5/27$.

The Minimization Problem

In traffic applications, we want to reach the destination in an optimal way. Since link travel times are random variables, there exist multiple criteria on what optimal travel times are. Usually the primary concern of routing is the expected travel times from origins to destinations, i.e. a routing policy with less expected travel time is a better one. However, the variances of O-D travel time random variables are also important. Depending on the traveler's attitude toward risk and the type of trips, he/she will make trade-offs between expected travel times and variances. A routing model should be able to handle this trade-off.

The expected travel time will be used as the only optimization criterion for the time being, and the variance and expected schedule delay will be modeled in Chapter 4. Define t_x as the current time of state x, and $E[Z]$ as the expectation of random variable Z. The **optimal routing policy problem in a stochastic time-dependent network** with one destination node d is to find μ^*, such that

$$\mu^* = \arg\min_{\mu} \{E_{\{x_0, x_1, ..., x_S\} \in M(x_0, \mu)}[t_{x_S} - t_{x_0}]\}, \quad \forall x_0 \tag{2.1}$$

$t_{x_S} - t_{x_0}$ is the travel time from the origin as defined in the initial state x_0 to the destination node d for a given routing policy μ. It is a random variable and each of its possible values is associated with a state chain $\{x_0, x_1, ..., x_S\}$. The minimum is taken over all routing policies. Note an optimal routing policy is optimal for all initial states, not just for a specific initial state.

We can compare the optimal routing policy with a shortest path tree in the deterministic and static all-to-one shortest path problem. The classical all-to-one shortest path problem is to find the shortest paths from all nodes to one destination node in a static and deterministic network. The result is a directed in-tree rooted at the destination node. The shortest path from any node j to d is the path from j to d in the shortest path tree. The shortest path tree can be viewed as a specialized routing policy, where there is only one possible state for a given node and the decision (next node) for that state is the successor node in the shortest path tree. In the classical all-to-one shortest path problem, *all* stands for "all nodes", while in the optimal routing policy problem in an STD network, we have an implicit *all* standing for "all times" and "all current information" as well as all nodes. A counterpart of the shortest path tree in the optimal routing policy problem would be the union of state networks for all the possible states. There is no guarantee that the state network union is acyclic or connected, however, as opposed to the shortest path tree.

The Optimality Condition

Let $e_\mu(x)$ denote the expected travel time to the destination node d when the initial state is x and the routing policy μ is applied. Define $A(j)$ as the set of downstream nodes of node j, $\tilde{C}_{jk,t}|I$ as the travel time random variable for link (j,k) at time t conditional on current information I, and $\tilde{I}'|I$ as a current information random variable at the next node k and at time $t + \tilde{C}_{jk,t}|I$. We make the assumption that there exists at least one path from any node to the destination node d under any possible value of the link travel time vector.

Denote $Z(j,t)$ as the set of all possible current information at node j and at time t. For $\forall j \in N - \{d\}, \forall t \in T, \forall I \in Z(j,t)$, $e_{\mu^*}(x)$ and μ^* are optimal if and only if they are solutions of the following system of equations:

$$e_{\mu^*}(j,t,I) = \min_{k \in A(j)} \left\{ E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + E_{\tilde{I}'}[e_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|\tilde{C}_{jk,t}]|I] \right\} \tag{2.2}$$

$$\mu^*(j,t,I) = \arg\min_{k \in A(j)} \left\{ E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + E_{\tilde{I}'}[e_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|\tilde{C}_{jk,t}]|I] \right\} \tag{2.3}$$

with the boundary conditions: $e_{\mu^*}(d,t,I) = 0, \mu^*(d,t,I) = d, \forall t \in T, \forall I \in Z(d,t)$. Since $\tilde{I}'|I$ is dependent on $\tilde{C}_{jk,t}|I$, we first take the expectation over $\tilde{I}'|I$ with a given realization of $\tilde{C}_{jk,t}|I$ and then take the expectation over $\tilde{C}_{jk,t}|I$. Note that we assume the outcome of the decision is deterministic, i.e. the traveler will end up at node k if he/she chooses node k as his/her next node. Croucher (1978) [20] studies the problem where the outcome of the decision itself is stochastic. We do not discuss this case, as our motivation in studying the ORP problem is for traffic applications where this case rarely arises.

The proof of the optimality conditions is similar to the proof of Proposition 7.2.1 in Bertsekas (2000) [12, p. 368-371]. The problem in Bertsekas (2000) [12] is denoted as a stochastic shortest path problem and is viewed as an infinite horizon dynamic programming problem. The proof provided uses only the node number as a state, yet we can simply replace the state by $\{j,t,I\}$ and the proof becomes valid for our case.

If waiting is allowed, we can add one more choice for the traveler, which is to wait for one time period at node j. The optimality conditions in this case can be written as:

For $\forall j \in N - \{d\}, \forall t \in T, \forall I \in Z(j,t)$, $e_{\mu^*}(x)$ and μ^* are optimal if and only if they are solutions of the following system of equations:

$$\begin{aligned} e_{\mu^*}(j,t,I) = \min_{k \in A(j), j}\{ \quad & 1 + E_{\tilde{I}'}[e_{\mu^*}(j, t+1, \tilde{I}')], \\ & E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + E_{\tilde{I}'}[e_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|\tilde{C}_{jk,t}]|I]\} \end{aligned} \tag{2.4}$$

$$\begin{aligned} e_{\mu^*}(j,t,I) = \arg\min_{k \in A(j), j}\{ \quad & 1 + E_{\tilde{I}'}[e_{\mu^*}(j, t+1, \tilde{I}')], \\ & E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + E_{\tilde{I}'}[e_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|\tilde{C}_{jk,t}]|I]\} \end{aligned} \tag{2.5}$$

with the boundary conditions: $e_{\mu^*}(d,t,I) = 0, \mu^*(d,t,I) = d, \forall t \in T, \forall t \in T, \forall I \in Z(d,t)$.

2.2.2 Taxonomy

In this subsection, we provide a taxonomy for the optimal routing policy problem in an STD network. There are four major objectives of providing this taxonomy:

- To make the abstract framework concrete and applicable to the traffic context
- To show the variety of optimal routing policy problems
- To study the role of information in a stochastic routing context
- To gain insight into the complexity of the problem

The framework is abstract in the sense that no concrete form of current information I is specified. current information depends on two factors: network stochastic dependency, defined as the stochastic dependency of link travel time random variables; and information access, defined as the link travel time realizations that are available to travelers at any given time and given node. The taxonomy of the ORP problem is therefore defined along these two dimensions. We will see that depending on the assumptions about these two factors, we can have a large variety of the ORP problem variants. Some of them are just for the purpose of theoretical analysis, while others are realistic in the traffic context. Specifically we can see the role of information in stochastic routing. In fact, ITS applications largely rely on the acquisition and processing of information on traffic conditions, therefore the study of the role of information is needed. During the discussion of each variant, we give a brief overview of the complexity of the ORP problem and show how the complexity differs from variant to variant.

Taxonomy

Network stochastic dependency is characterized by link-wise and time-wise statistical dependencies of link travel times. At one extreme, all the link travel time random variables are independent, both link-wise and time-wise. At the other extreme, all the link travel time random variables are completely dependent. There are numerous cases between these two extremes, and we denote them as having partial stochastic dependency.

Information access has the following four categories:

- Full information (FI)
- Perfect online information (POI)
- Imperfect online information (IOI)
- No online information (NOI)

Travelers with full information have knowledge of the realizations of all link travel time random variables before the trip. Travelers with perfect online information have knowledge of the realizations of all link travel times up to the current time. Travelers with imperfect online information have knowledge only of part of the link travel time realizations where the restrictions in online information can be temporal, spatial or both. Travelers with no online information have no knowledge of any of the realizations and the only knowledge they have about the current state is the current node and current time. Table 2.2 gives a possible taxonomy along the two dimensions.

	Full Information	Perfect Online Information	Imperfect Online Information	No Online Information
No link-wise and no time-wise dependency	WS (wait-and-see)	Group 1		NOI
Complete dependency		Group 2	Group 3	
Partial dependency				

Table 2.2: Taxonomy of the ORP Problem

Discussion of Taxonomy

In the discussion of the variants listed in Table 2.2, we focus on the specification of current information for each variant and the resulting implications for algorithm design. A general rule in determining current information is as follows: information access determines which link travel times have the potential to be included in current information, while network stochastic dependency determines whether all the available link travel time realizations are necessary. The unnecessary link travel times can be eliminated so that the dimension of current information is minimized. For example, assume we are equipped with the most advanced traffic information system so that we know the realizations of all link travel times up to the current time (i.e. perfect online information). Presumably we hope we can make use of all the available information. However, assume all the link travel time random variables are statistically independent, implying that knowledge about one link at one time cannot help travelers make inferences about any other link at any time, then a large part of the information is not useful and the current information could contain much fewer link travel times than those available. This rather extreme case shows how the two factors act together to determine current information, and we will see more in the following discussion. Throughout the discussion, we assume that a user always knows his/her current node and current time (not necessarily remembering the arrival times at past nodes), and any knowledge about link travel time realizations can only be represented by the current information. This implies that if we say the current information is empty, then the user cannot obtain the travel time on a link by taking the difference between the arrival times at the two end nodes of that link.

The WS variant has full information and any kind of network stochastic dependency. We borrow a term from stochastic programming to denote the variant as WS (wait-and-see). In WS, the current information I includes all the link travel times in all time periods, so travelers can know the network deterministically *a priori*. Mathematically, in a network $G = (N, A, T, P)$ as defined in Section 2.2.1, we have current information $I = A \times T$ and the traveler knows *a priori* which support point of the multivariatelink travel time distribution, $v_1, v_2, ...$ or v_R, the network will take. This variant is not realistic, as in reality the future is always uncertain to some extent. However it is useful here, since for a network with a given type of stochastic dependency, the WS variant gives a solution lower bound for all other variants of the

ORP problem. It can be used as a benchmark in the robustness analysis of solutions to other variants.

Under full information, the ORP problem reduces to multiple deterministic dynamic shortest path problems, each of which works on a deterministic network defined by one of the R support points $v_1, v_2, ..., v_R$. Since we are working only on deterministic networks, network stochastic dependency does not make any difference in algorithm design. Algorithm DOT by Chabini (1999) [17] with a running time of $\theta(SSP + nK + mK)$ for all-to-one shortest path problem, where SSP is the running time of a classical static shortest path algorithm, can be used to solve the individual deterministic dynamic shortest path problems. The complexity of the WS variant is then $\theta(R \times (SSP + nK + mK))$.

The No-Online-Information variant is the other extreme case when no online information (NOI) is available. The lack of information prevents travelers from making any useful inferences about future network conditions. The current information is an empty set at any point in space and time, and decisions depend only on current node and current time. This is true for any kind of stochastic dependency, which is another example beside the WS variant showing the role of information in defining an ORP problem. As the current information is an empty set, we can simply remove it from the current-state, and the optimality conditions in Section 2.2.1 reduce to

$$e_{\mu^*}(j,t) = \min_{k \in A(j)} \{E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + e_{\mu^*}(k, t + \tilde{C}_{jk,t})]\} \tag{2.6}$$

$$\mu^*(j,t) = \arg\min_{k \in A(j)} \{E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + e_{\mu^*}(k, t + \tilde{C}_{jk,t})]\} \tag{2.7}$$

Several algorithms have appeared in the literature to solve this variant (Hall (1988) [29], Chabini (2000) [18], Miller-Hooks and Mahamassani (2000) [35], Bander and White (2002) [8], Pretolani (2000) [42]), yet no explicit discussion of the role of information is provided. In some papers link travel times are assumed to be statistically independent to obtain the optimality conditions shown above. However, the assumption of statistical independence is neither necessary nor sufficient to validate (2.6) and (2.7). It is not necessary, because variants with statistically *dependent* link travel times and no online information can satisfy this formulation. It is not sufficient, because if the realizations of current outgoing link travel times are available in a statistically *independent* network, the current information is no longer an empty set. Instead, it will contain travel times of outgoing links at the current time. Thus the above optimality conditions for the problem are no longer valid.

Algorithm DOT-S by Chabini (2000) [18] has an optimal running time of $\theta(SSP + nKQ + mKQ)$ for the NOI variant, where Q is the maximum number of support points of a single link travel time discrete distribution, in the sense that no algorithms with less theoretical complexity exist. In Section 2.3, the solution algorithms for the NOI variant are extensively discussed and computational tests are presented.

The Independent variants. In the rest of the section, we discuss variants with some online information access. The above discussion shows that sometimes information access alone can determine the current information, as in the case of WS and NOI. On the other hand, network stochastic dependency can sometimes play a very important role in determining the current information. This can be shown by the variants in Group 1 with statistically independent link travel times. First of all, knowledge about the downstream links of the current node at the current time is useful, as they are explicitly included in the optimality conditions. Any other link travel time realizations, however, cannot contribute to the decision making. Define $\delta(j)$ as the downstream links of node j. This fact is stated formally as follows:

Theorem 2.2.4. *For a given network $G = (N, A, T, P)$ as defined in Section 2.2.1 where the link travel time random variables $\tilde{C}_{jk,t}$ are stochastically independent, $\forall (j,k) \in A, \forall t \in T$, define two types of current information:*

1. $I_1 = A \times \{0, 1, ..., t\}, \quad \forall t \in T$

2. $I_2 = \delta(j) \times \{t\}, \quad \forall j \in N, \forall t \in T$

Let μ_1^* and μ_2^* be the optimal routing policies respectively for the first and second definitions of current information. If travel time realizations of $\delta(j) \times \{t\}$ are the same in $I_1(t)$ and $I_2(j,t)$, we have

$$e_{\mu_1^*}(j,t,I_1) = e_{\mu_2^*}(j,t,I_2), \forall I_1, I_2, \forall j \in N, \forall t \in T.$$

Proof. We use induction on time t to prove the theorem. Since travel time realizations of $\delta(j) \times \{t\}$ are the same in $I_1(t)$ and $I_2(j,t)$, let $\pi_{jk,t}$ denote the travel time realization of link (j,k) at time t in both I_1 and I_2. The following notation is used in the proof:

$$I_1' = A \times \{0, 1, ..., t, ..., t + \pi_{jk,t}\}, \quad \forall t \in T$$

$$I_2' = \delta(k) \times \{t + \pi_{jk,t}\}, \quad \forall k \in A(j), \forall t \in T$$

$$\bar{I}_1 = I_1' - I_1$$

$$I_{12} = \bar{I}_1 - I_2'$$

Please see Figure 2-3 for an intuitive representation of the relationships between these variables.

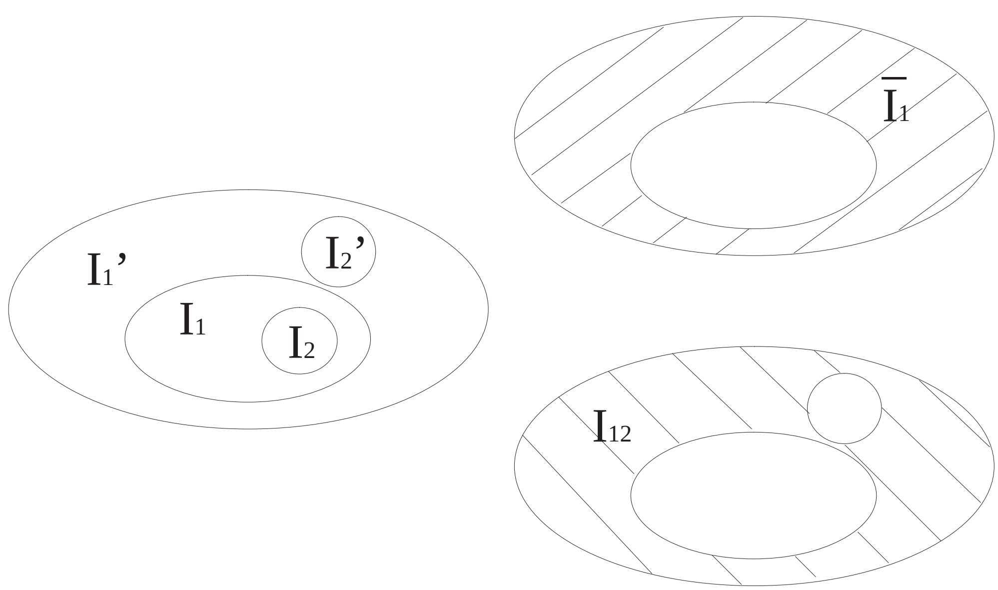


Figure 2-3: Relationship of $I_1, I_2, I_1', I_2', \bar{I}_1, I_{12}$

Induction Base: When $t \geq K-1$, the network is deterministic and static. Therefore the ORP problem reduces to the classical shortest path problem where the shortest distance to the destination only depends on the origin node. Thus $e_{\mu_1^*}(j,t,I_1) = e_{\mu_2^*}(j,t,I_2), \forall I_1, I_2, \forall j \in N, \forall t \geq K-1$.

Induction Assumption: Assume $e_{\mu_1^*}(j,t,I_1) = e_{\mu_2^*}(j,t,I_2), \forall I_1, I_2, \forall j \in N, \forall t \geq l$.

Induction Step: When $t = l-1$,

$$
\begin{aligned}
& e_{\mu_1^*}(j,t,I_1) \\
= \quad & \min_{k\in A(j)}\{E_{C_{jk,t}|I_1}[C_{jk,t}|I_1 + E_{I_1'|I_1}[e_{\mu_1^*}(k, t + C_{jk,t}|I_1, I_1'|I_1)]]\} \\
= \quad & \min_{k\in A(j)}\{\pi_{jk,t} + E_{I_1'|I_1}[e_{\mu_1^*}(k, t + \pi_{jk,t}, I_1'|I_1)]\} \\
= \quad & \min_{k\in A(j)}\{\pi_{jk,t} + E_{\bar{I}_1}[e_{\mu_1^*}(k, t + \pi_{jk,t}, I_1 + \bar{I}_1)]\} \\
= \quad & \min_{k\in A(j)}\{\pi_{jk,t} + E_{I_2'}[E_{I_{12}}[e_{\mu_1^*}(k, t + \pi_{jk,t}, I_1 + I_{12} + I_2')]]\} \\
= \quad & \min_{k\in A(j)}\{\pi_{jk,t} + E_{I_2'}[E_{I_{12}}[e_{\mu_2^*}(k, t + \pi_{jk,t}, I_2')]]\} \\
= \quad & \min_{k\in A(j)}\{\pi_{jk,t} + E_{I_2'}[e_{\mu_2^*}(j, t + \pi_{jk,t}, I_2')]\} \\
= \quad & \min_{k\in A(j)}\{\pi_{jk,t} + E_{I_2'|I_2}[e_{\mu_2^*}(j, t + \pi_{jk,t}, I_2'|I_2)]\} \\
= \quad & \min_{k\in A(j)}\{E_{C_{jk,t}|I_2}[C_{jk,t}|I_1 + E_{I_2'|I_2}[e_{\mu_2^*}(k, t + C_{jk,t}|I_2, I_2'|I_2)]]\} \\
= \quad & e_{\mu_2^*}(j,t,I_2)
\end{aligned}
$$

The first equality is due to the definitions of $e_{\mu_1^*}$. The second equality is due to the definition of $\pi_{jk,t}$. The third equality is due to the statistical independence of link travel times in $\bar{I}_1$ and I_1. The fourth equality is due to the definition of I_{12} and the statistical independence of link travel times in I_{12} and I_2'. The fifth equality is due to the induction assumption. The sixth equality is due to the statistical independence of link travel times in I_{12} and I_2'. The seventh equality is due to the statistical independence of link travel times in I_2 and I_2'. The eighth equality is due to the definition of $\pi_{jk,t}$. The ninth (last) equality is due to the definition of $e_{\mu_2^*}$. □

We can extend the theorem to the case when only part of the downstream link travel time realizations are available. We conclude that current information I for a given current node and a given current time in Group 1 is the **available** current travel time realizations of **downstream** links of the node. Mathematically speaking, $I(j,t) = \delta(j) \cap IA$, where IA stands for information access, i.e. the available link travel time realizations. When there is no knowledge about the downstream links of the current node at the current time, the current information becomes an empty set and the problem has the same current information as in the NOI variant. Note that the name "NOI" represents a variant where current information I is an empty set. No online information is only a sufficient condition to validate the specification of current information. We choose "NOI" as the name, as it is intuitive to link the idea of an empty current information from the no-online-information assumption. However, we should remember that there are other conditions that can correspond to the "NOI" formulation, one of which was just discussed.

Variants with complicated information access and stochastic dependency. Variants in Groups 2 and 3 generally have more complicated current information than others. All available link travel time realizations are potentially useful and could be included in the current information I. Network stochastic dependency can be utilized to eliminate unnecessary link travel times from the current information, as in the independent case, but the judgment sometimes requires a large amount of work and the resulting reduction in the current information dimension may not compensate for this extra effort. In a word, the determination of current information for variants in these two groups depends largely on the actual assumptions on both information access and network stochastic dependency. In Section 2.4, we will discuss the perfect online information variants in Group 2 in more detail.

Most transportation networks belong to Groups 2 and 3. For example, a typical urban traffic network can be divided into several zones and we can assume that traffic within one zone is highly dependent, while weak relationships exist between traffic within one zone and other zones. Furthermore, we can assume that only recent traffic conditions, say within the last hour, are helpful in predicting future conditions. It is also very likely that there are several local traffic information centers that provide information to vehicles within their respective functional ranges. All these assumptions about network stochastic dependency and information access complicate the problem, and careful problem definition is required.

To the best of the author's knowledge, the two papers by the author of this thesis (Gao and Chabini (2002) [27], Gao and Chabini (2004) [28]) are the only research in the literature that studies optimal routing policy problems with complicated information access and stochastic dependency in a time-dependent context, or equivalently, the decisions are dependent on the triple (j, t, I), where I is not empty. All other papers deal with the problems where decisions are dependent on the pair (j, t) only (Hall (1988) [29], Chabini (2000) [18], Miller-Hooks and Mahamassani (2000) [35], Miller-Hooks (2001) [36], Yang and Miller-Hooks (2004) [49], Bander and White (2002) [8], Pretolani (2002) [42]). However, in a time-independent context, quite a few papers address the problems with complicated information access and stochastic dependency (Andreatta and Romeo (1988) [5], Polychronopoulos and Tsitsiklis (1996) [41], Cheung (1988) [19], Provan (2003) [43], Psaraftis and Tsitsiklis (1993) [44]).

We distinguish between Groups 2 and 3, because the complexity of algorithms for variants in each group could differ greatly. Complexity of an algorithm for the ORP problem depends largely on the maximum number of possible distinctive current information values. For the sake of convenience of presentation, assume the partial online information is partial in the spatial dimension, not in the temporal dimension. With perfect online information, the current information is composed of all link travel times up to current time t, and the maximum number of distinct current information values is just the maximum number of support points of the discrete distribution of these tm random variables, which is at most R. With partial spatial online information, however, the current information is composed of links around the path (what specific links are included depends on specific assumptions about "partial" spatial dependency) from the origin to the current node. Therefore the current information can contain travel times of a maximum of $2^{tm} - 1$ different sets of links. As each set of link travel times has at most R distinct values, the maximum number of distinct current information values is $(2^{tm} - 1)R$. The maximum numbers of distinct current information values in these two groups differ by a ratio of $2^{tm} - 1$, which is significant. It is stated in Polychronopoulos and Tsitsiklis (1996) [41] that in a static network, the maximum number of distinct current information values with partial online information is $2^R - 1$. This is quite a loose upper bound, and a tighter upper bound obtained by applying the above logic would be $(2^m - 1)R$.

The dynamic shortest path problem in acyclic networks with independent stationary Markovian arc costs studied in Psaraftis and Tsitsiklis (1993) [44] can be viewed as a variant in Group 3. The following specifications are needed: 1) for each node, there will be an additional outgoing link that ends at the node itself with a travel time of one unit time, which will enable waiting at a node; 2) the number of time periods is infinite; 3) links with the same tail node are dependent link-wise and time-wise in a way described by the Markovian process at the tail node, and links with different tail nodes are independent; 4) knowledge about outgoing link travel time realizations is available at the current node. Please refer to the literature review of Section 2.1 for an introduction of the basic assumptions. The assumption of acyclic networks implies that node j cannot be visited again after the traveler leaves it. Since the Markovian arc costs are independent across nodes, it is not helpful to keep information on any already visited nodes. Thus the dimension problem of current information with partial spatial online information as discussed above does not exist in this case. This assumption along with the stationary assumption makes a polynomial running time algorithm possible.

2.2.3 Policy vs. Path

The minimum expected travel time path problem in a stochastic time-dependent network from origin s at departure time t_0 to destination d can be written as an optimal routing policy problem with some additional constraints. Specifically, the problem is to

$$\begin{array}{ll} \min_{\mu} & \{E_{\{x_0,x_1,\ldots,x_S\}\in M(x_0,\mu)}[t_{x_S}-t_{x_0}]\} \quad \forall x_0|t_{x_0}=t_0, j_{x_0}=s \\ s.t. & \mu(y)=\mu(z), \forall y,z\in M(x_0,\mu)|j_y=j_z \end{array}$$

where j_x and t_x are the current node and current time of state x respectively. The additional constraints make sure that once the origin and departure time is given, the next node to be taken at each successive node is independent of time or information, and thus can be determined *a priori*.

From the knowledge of constrained optimization, an optimization problem with fewer constraints can attain an optimum at least as good as one with more constraints. Therefore, an optimal routing policy can attain an expected travel time at least as low as a minimum expected travel time path can do.

2.3 The No-Online-Information Variant

In this section, we discuss in detail the no-online-information (NOI) variant of the optimal routing policy problems in STD networks. In subsection 2.2.2, we defined the no-online-information variant as a variant of the ORP problem where the current information component of any state is an empty set. We also discussed several situations under which the NOI variant is applicable. These situations include:

- When no knowledge about any of the link travel time realizations is available
- When all link travel time random variables are statistically independent *and* no knowledge about the realizations of current outgoing link travel times from the current node is available

Under either of these situations, current information will not appear in the optimality conditions. In other words, routing decisions only depend on the current node and the current time. We think that the decision dependency (i.e. what the routing policies are based on) is the key in defining a variant, as it directly affects the algorithm design. In light of this, a more general definition of the NOI variant would be: the current information component of any state is the same. In this case, routing decisions also only depend on the current node and the current time, and the current information can actually be ignored in the algorithm design.

2.3.1 Motivation

There are three kinds of motivation for studying the no-online-information variant. Theoretically, the NOI variant is the simplest in terms of algorithm design among all

ORP problem variants with on-line information, due to the lack of current information. It is therefore the basis for the study of more complicated variants. Furthermore, even though it is the simplest, it suffices to show some of the implications and significance of stochasticity in a dynamic context for traffic models. It also shows how information access can affect the routing problem formulation. Practically, there do exist quite a few traffic situations where the NOI formulation is applicable - for example, in a network where link travel times are weakly coupled, or where little or no information is available. Computationally, the NOI variant can be solved in polynomial time, as shown later in this section. This is a very desirable result, as the ORP problem in a STD network generally requires exponential running time to solve. Therefore NOI can be used as an approximation to more complicated variants, as we will discuss in detail in Chapter 3.

As we have discussed in the literature review and in the taxonomy, there have been quite a few papers in the literature that study the NOI variant. This section presents in detail methods in Chabini (2000) [18] and Miller-Hooks (2001) [36], and conducts computational tests for these methods.

2.3.2 Optimality Conditions

The optimality conditions have already been presented in subsection 2.2.2, but we also list them here for ease of reference:

$$e_{\mu^*}(j,t) = \min_{k \in A(j)} \{E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + e_{\mu^*}(k, t + \tilde{C}_{jk,t})]\} \tag{2.8}$$

$$\mu^*(j,t) = \arg\min_{k \in A(j)} \{E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + e_{\mu^*}(k, t + \tilde{C}_{jk,t})]\} \tag{2.9}$$

with boundary conditions: $e_{\mu^*}(d,t) = 0, \forall t \in T$, and $e_{\mu^*}(j,t) = e_{\mu^*}(j, K-1), \forall j \in N, \forall t > K-1$.

Consider a traveler in a network whose level of uncertainty never decreases. The traveler's knowledge about the network remains the *a priori* distribution of link travel times, either because he/she has no en route information access, or because the network is statistically independent and online information cannot help predict the future. Thereafter, one need work only with the unconditional marginal distributions of link travel times, as shown in the optimality conditions, either because there is nothing to be conditional on, or because the conditional probabilities are the same as the unconditional probabilities.

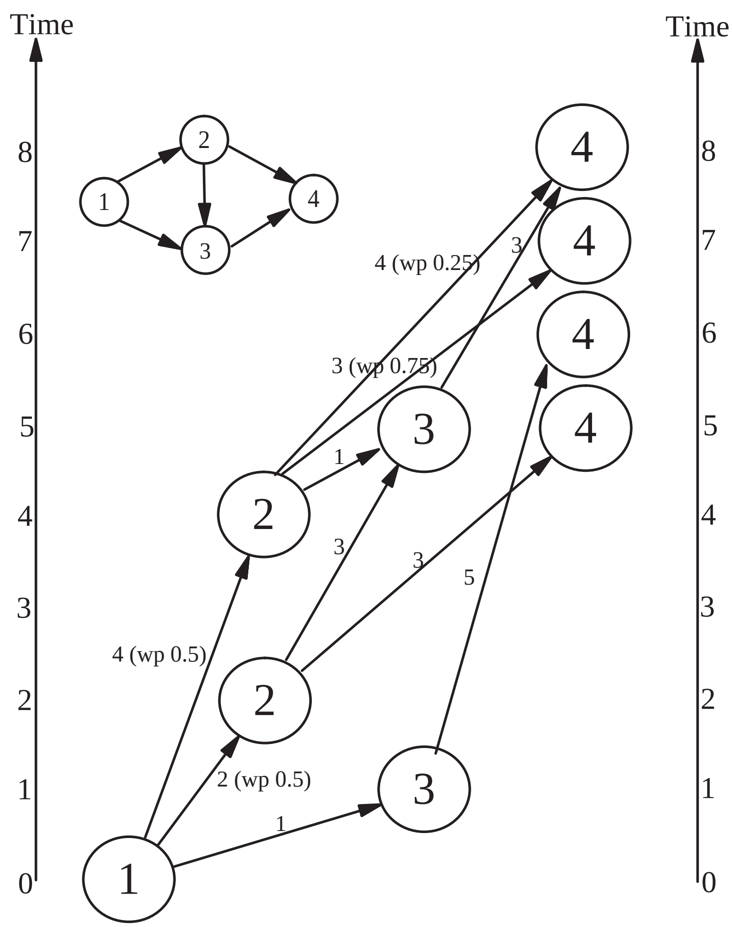


Figure 2-4: An Illustrative Example for NOI Optimality Conditions: Topological Network and Time-Space Network

We will show an illustrative example of how the NOI optimality conditions work.

The topological network is shown at the upper-left corner of Figure 2-4, and the major part of the figure is a time-space representation of the network. In a time-space network, time is shown along the vertical axis (the time axis), and the node number is shown along the horizontal axis (the space axis). Each point in this network represents a node-time pair (j, t), and any link between (j, t_1) and (k, t_2) indicates that link (j, k) has a travel time of $t_2 - t_1$ if departure time from node j is t_1. We are interested in finding the optimal routing policy from node 1 to node 4 at departure time 0, namely $e_{\mu^*}(1, 0)$, and only those node-time pairs and links relevant to the computation are shown.

Figure 2-4 shows the marginal distributions of the link travel time random variables. Link $(1, 2)$ at time 0 could have two values of travel time: 4 *w.p.* 0.5 and 2 *w.p.* 0.5. Link $(2, 4)$ at time 4 could have two values of travel times: 4 *w.p.* 0.25 and 3 *w.p.* 0.75. All other link travel times are deterministic.

We apply the optimality conditions to obtain the value of $e(1, 0)$.

$$e_{\mu^*}(1, 0) = \min\{1 + e_{\mu^*}(3, 1), 0.5 \times (2 + e_{\mu^*}(2, 2)) + 0.5 \times (4 + e_{\mu^*}(2, 4))\}.$$

It can be easily observed from the figure that $e_{\mu^*}(3, 1) = 5$ and $\mu^*(3, 1) =$ (node) 4, and $e_{\mu^*}(2, 2) = 3$ and $\mu^*(2, 2) =$ (node) 4. We apply the optimality condition again to obtain $e_{\mu^*}(2, 4)$:

$$e_{\mu^*}(2, 4) = \min\{1 + e_{\mu^*}(3, 5), 0.25 \times 4 + 0.75 \times 3\} = \min\{1 + 3, 1 + 2.25\} = 3.25$$

and $\mu^*(2, 4) =$ (node)4. With the values of $e_{\mu^*}(3, 1), e_{\mu^*}(2, 2)$, and $e_{\mu^*}(2, 4)$ in hand, we can obtain

$$e_{\mu^*}(1, 0) = \min\{1 + 5, 0.5 \times (2 + 3) + 0.5 \times (4 + 3.25)\} = 6$$

and $\mu^*(1, 0) =$ (node) 3. Therefore the optimal routing policy for node 1 at time 0 turns out to be a path: 1-3-4.

2.3.3 Algorithm DOT-S

We can associate with each pair (j, t) a label which is the upper bound of the minimum expected travel time from node j to the destination node d at departure time t. We will design a procedure to update these labels according to the optimality conditions, until all of them are optimal. Depending on the way the labels are updated, there are two different algorithms.

Algorithm DOT-S is a counterpart of Algorithm DOT by Chabini (1999) [17] which finds the shortest path in a deterministic time-dependent network. DOT stands for "Decreasing Order of Time", and S stands for "Stochastic". It is noted that the update of labels at time t depends only on labels at times later than t, due to the assumption of positive link travel times. Therefore, we can first solve a classical shortest path problem for the deterministic and static period where link travel times are $c_{jk,K-1}, \forall (j,k) \in A$, and set $e_{\mu^*}(j, K-1)$ = shortest path length from j in the classical SSP problem, $\forall j \in N$. We then proceed to the labels of time $K-2$ which only depend on labels of time $K-1$. As labels of time $K-1$ are already optimal, by optimality condition 2.8, the updated labels of time $K-2$ are also optimal. We continue this procedure back in time until time 0, and every label will then be set optimally.

We define $\tau^v_{jk,t}$ as the v^{th} support point of the marginal distribution of travel time of link (j,k) at time t, and $q^v_{jk,t}$ the corresponding marginal probability. We also define Q as the maximum number of support points for a single link travel time marginal distribution. The statement of Algorithm DOT-S is as follows:

Algorithm DOT-S

Step 0: (Initialization)

0.1: Run a shortest path problem algorithm (e.g. Dijkstra's) on the deterministic and static network $G'(N, A)$ where link (j,k) has a travel time of $c_{jk,K-1}, \forall (j,k) \in A$;

0.2: $e_{\mu^*}(j, K-1)$ = Shortest path length from node j to node d;

0.3: $e_{\mu^*}(j,t) = \infty, \mu^*(j,t) = \infty, \forall j \in A - \{d\}, \forall t < K-1$;

$e_{\mu^*}(d,t) = 0, \forall t \in T.$

Step 1: (Main loop)

for $t = K - 1$ to 0

for $(j,k) \in A$

$temp = \sum_v (\tau^v_{jk,t} + e_{\mu^*}(k, t + C_{jk,t})) \times q^v_{jk,t};$

If $temp < e_{\mu^*}(j,t)$

$e_{\mu^*}(j,t) = temp$

$\mu^*(j,t) = k$

Let us now compare solutions from Algorithm DOT-S with those from Algorithm DOT. They look similar, as each pair (j,t) has an associated cost to the destination node, and an associated next node to take. The difference can be obtained by tracing a traveler through the network. Assume a traveler starts from the pair (j,t) in a deterministic time-dependent network and follows the optimal routing decisions computed from Algorithm DOT. As the travel times are deterministic, we can tell with certainty when he/she will arrive at each downstream node and thus the path he/she will take can be determined. Instead of telling him/her to make routing decisions based on current node j and current time t, one can just tell him/her to follow an *a priori* path. However, if the network is an STD network and the traveler has no information access, one cannot tell what path he/she will end up following before the trip begins, as the link travel times are random. In other words, the traveler could arrive at a downstream node at several possible times. Therefore the traveler must have the routing policy $\mu^*(j,t)$ computed from Algorithm DOT-S and make decisions depending on arrival times.

The complexity analysis of Algorithm DOT-S is straightforward. At initialization, a classical shortest path algorithm is run with a running time $\theta(SSP)$, where SSP is the running time of the shortest path algorithm (cf Ahuja, Magnanti and Orlin (1993) [4] for a summary of running times of different algorithms). In the main loop, at each time period of the dynamic period (i.e. $t < K-1$), each arc is visited exactly once with Q arithmetic operations, and each node is visited at least once and at most three times. Therefore the running time of the main loop is $\theta(nK + mQK)$. To sum up, the complexity of Algorithm DOT-S is $\theta(SSP + nK + mQK)$.

It is shown by Chabini (2000) [18] that Algorithm DOT-S is optimal, in the sense that no other algorithm can have better theoretical complexity. The argument is as follows. In order to make sure the solution is optimal, any algorithm has to retrieve the data for each dynamic period, i.e. $t < K - 1$, at least once. The data for the problem is the discretized marginal probability distributions for all links at all times lower than $K - 1$. Thus the retrieval of data takes running time of $\theta(mKQ)$. Furthermore, any solution algorithm must in the worst case compute and output, or at the very least initialize $\theta(nK)$ variables consisting of the values of $e_{\mu^*}(j, t)$ and $\mu^*(j, t)$ for all pair (j, t). Finally, computing all-to-one least expected travel times for departure times beyond the time horizon $K-1$, is equivalent to computing an all-to-one shortest path tree using $c_{jk,K-1}$ as link travel times. In summary, any solution algorithm to the NOI variant has a worst-case complexity of at least $\theta(SSP + nK + mKQ)$. Since Algorithm DOT-S has a worst-case complexity of exactly $\theta(SSP + nK + mKQ)$, it is optimal.

2.3.4 Extension to Minimum Expected Cost Problems

So far we have focused on the minimum expected travel *time* problem. In fact, the minimum expected travel cost problem can be handled with straightforward extension. Define a link cost function $g(C_{jk,t})$ to be the cost of link (j, k) at time t as a function of link travel time $C_{jk,t}$, and $g(0) = 0$. The minimum expected cost problem is to find a routing policy with minimum expected cost from all origins for all departure times to the destination node.

The optimality conditions for the minimum expected cost problem can be obtained by making slight changes from those for the minimum expected travel time problem. For the sake of notational simplicity, we still use $e_\mu(j, t)$ to denote the expected cost of a routing policy μ with origin node j, departure time t, and empty current information set. The optimal routing policy μ^* and the corresponding optimal expected cost e_{μ^*}

are solutions to the following system of equations:

$$e_{\mu^*}(j,t) = \min_{k\in A(j)} \{E_{\tilde{C}_{jk,t}}[g(\tilde{C}_{jk,t}) + e_{\mu^*}(k, t + \tilde{C}_{jk,t})]\} \tag{2.10}$$

$$\mu^*(j,t) = \arg\min_{k\in A(j)} \{E_{\tilde{C}_{jk,t}}[g(\tilde{C}_{jk,t}) + e_{\mu^*}(k, t + \tilde{C}_{jk,t})]\} \tag{2.11}$$

with the boundary conditions: $e_{\mu^*}(d,t) = 0, \forall t \in T$, and $e_{\mu^*}(j,t) = e_{\mu^*}(j, K-1), \forall j \in N, \forall t > K-1$.

Algorithms for the minimum expected cost problem can be obtained similarly. We can see that algorithms for the "cost" problem have the same asymptotic running times as those for the "time" problem, as the only additional operation of the "cost" problem is the mapping from $C_{jk,t}$ to $g(C_{jk,t})$. In actual implementation, the mapping can be done in the data generation. For example, for each possible value of $C_{jk,t}$, one can generate an associated cost. In this case, the "cost" problem algorithms and the "time" problem algorithms have exactly the same running times.

2.3.5 Computational Tests

Extensive computational tests have been carried out. The objectives of the computational tests are to study experimentally the running time of Algorithm DOT-S as a function of various network parameters.

Random Network Generator

The random network generator generates a random directed network on which the algorithms are to be applied. Two sets of data have to be generated, the topology of the network and the discretized link travel time distributions. To generate the network topology, the required input from the users is: 1) the number of nodes n; 2) the number of links m; 3) the maximum in-degree; and 4) the maximum out-degree. By default, the node with the highest number is set to be the destination node. To assure connectivity to the destination node, a directed in-tree rooted at the destination node is generated first. The remaining $m-(n-1)$ links are generated by selecting the head node and tail node randomly, assuring that the maximum in-degree and out-degree constraints are satisfied.

To generate the discretized link travel time distributions, the required input from the users is: 1) the number of time periods K; 2) the number of support points of a single link travel time distribution at a given time Q; 3) the maximum of link travel time values; 4) the minimum of link travel time values; 5) the maximum of link cost values; and 6) the minimum of link cost values. For each time point and each link, two sets of numbers are generated, the first set contains Q uniform random numbers in the range of the given minimum and maximum link travel time values, and the second set contains Q uniform random numbers in the range of 0 and 1. The first set are the link travel time support points for the specific link at the specific time, and the second set normalized by the sum of the Q numbers are the marginal probabilities associated with each support point.

Test Design

Basically the tests can be divided into two parts: those on sparse networks which are usually representative of transportation networks, and those on dense networks. For the tests on sparse networks, we set the ratio of the number of links to the number of nodes to a constant of 3. The maximum link travel time is 25, and the minimum link travel time is 1. The maximum link travel cost is 40, and the minimum travel cost is 1. We examine three different network topologies: 100 nodes, 500 nodes, and 1000 nodes. For each topology, there are 3 different numbers of support points for the link travel time distribution (Q): 5, 10, and 20; and 3 different numbers of time periods (K): 30, 60, and 90. Therefore there are 9 different sets of link travel time/cost data for a given topology, and a total of 27 experiments for sparse networks. We define an experiment as a series of runs with the same topological and link travel time/cost data, namely with the same triple (n, Q, K). 10 independent runs are carried out for each of the 27 experiments. In each run, the running time in CPU seconds for

Algorithm DOT-S for the minimum expected cost problem is recorded. We then take the average of the running times and their ratios over the 10 runs.

For the tests on dense networks, we fix the number of nodes to be 100, and have three different values for the number of links: 1000, 2500, and 5000, with average in- and out-degree of 10, 25, and 50 respectively. The maximum link travel time and maximum link cost are both $2 \times Q$, and the minimum link travel time and minimum link cost are both 1. We will discuss later in the tests results why we choose $2 \times Q$ rather than a fixed number. There are 3 different numbers of support points of the link travel time distribution that are the same as those in the sparse network tests: 5, 10, and 20. The values that the number of time periods can take are different from those in sparse tests: 60, 120, and 240. Therefore in the dense tests, an experiment is defined by a different triple (m, Q, K). Similarly, 10 independent runs are carried out for each experiment and averages of running times for both algorithms and their ratios are taken.

# nodes	# links	# support points	$K = 30$	$K = 60$	$K = 90$
100	300	5	0.012	0.026	0.041
		10	0.025	0.049	0.070
		20	0.045	0.089	0.133
500	1500	5	0.093	0.194	0.296
		10	0.146	0.305	0.463
		20	0.251	0.519	0.780
1000	3000	5	0.204	0.419	0.637
		10	0.308	0.628	0.966
		20	0.518	1.050	1.601

Table 2.3: Summary of Running Times (CPU sec.) – Sparse Networks (#links/#nodes = 3)

We use the classical label correcting algorithm with a complexity of $O(nm)$ to compute the static shortest path at the static period for Algorithm DOT-S. The codes are run on a Dell OptiPlex GX100 workstation with 933MHz CPU, 256 megabytes RAM, running the Red Hat Linux 7.0 operating system. All graphic output is included in Section 2.6 for ease of reading.

# nodes	# links	# support points	$K = 60$	$K = 120$	$K = 240$
100	1000	5	0.097	0.276	0.551
		10	0.183	0.439	0.903
		20	0.324	0.786	1.635
	2500	5	0.194	0.564	0.140
		10	0.372	0.921	1.842
		20	0.668	1.633	3.239
	5000	5	0.412	1.188	2.364
		10	0.756	1.904	3.727
		20	1.347	3.302	n/a

Table 2.4: Summary of Running Times (CPU sec.) – Dense Networks(#links/#nodes ≥ 10)

Test Results for Sparse Networks

The test results for sparse networks are shown in Table 2.3 and in Figures 2-6 through 2-8. Figure 2-6 show the running time of Algorithm DOT-S as a function of the number of links (m), with the number of time periods (K) fixed at 60, and for all three possible number of marginal distribution support points: 20, 10, and 5. We can see that the running time of Algorithm DOT-S increases linearly with the number of links for all three Q values. We can also see this nearly perfect linear relationship with respect to the number of time periods and the number of link time distribution support points in Figure 2-7 and Figure 2-8 respectively. These results are consistent with the theoretical analysis which gives a running time of $\theta(SSP + nK + mKQ)$. As Algorithm DOT-S is a dynamic-programming-type algorithm, the actual running time can be accurately analyzed. This explains the closeness between the theoretical worst case running time and experimental results.

Test Results for Dense Networks

The test results for dense networks are shown in Table 2.4 and in Figures 2-9 through 2-10. We study the running time of Algorithm DOT-S as a function of network parameter d which is defined as the average degree. It is varied by changing the number of links m while keeping the number of nodes n fixed. For algorithm DOT-S, this relationship should be asymptotically linear, yet we observe a relationship a little bit worse than linear in Figures 2-9 and 2-10. This is due to the overhead of shortest path computation in the static phase. Note we use a label correcting algorithm in the static phase, and the actual running time of a label correcting algorithm increases more than linearly with average degree.

2.4 The Perfect Online Information Variant

In the previous section, we studied the no-online-information (NOI) variant in detail. The assumption of an empty current information set is not so realistic in the presence of Advanced Traveler Information System (ATIS) and/or Advanced Traffic Management System (ATMS). On the other hand, a congested traffic network usually has highly inter-dependent link travel times, and thus the assumption of independent link travel times is also questionable. These considerations lead us to a more realistic variant, the perfect online information (POI) variant. As stated in the taxonomy, a traveler with perfect online information has knowledge about realizations of all links up to the current time. To put it another way, the current information I is a set $\{C_{jk,t}|(j,k) \in A, t \leq t_0\}$, where t_0 is current time. We will not make specific assumptions about the network stochastic dependency. Instead, we will adopt the most general probabilistic description of a network, i.e. the joint distribution description, to accommodate all kinds of assumptions on stochastic dependency. In particular, a network with strongly dependent link travel times can be handled with this description. It is a concern that the assumption of perfect online information is not in itself realistic. Acknowledging this, however, as discussed in subsection 2.2.2, variants with perfect online information are easier to study than those with partial online information. Furthermore, with the rapid development in sensor and telecommunication technology, perfect online information may be realistic in the near future. Finally, the algorithm for the POI variant also provides building blocks that can be used in developing algorithms for other variants with online information.

In this subsection, we present an operational algorithm DOT-SPI for the perfect on-line information variant. We introduce the important concept of event set, which is a counterpart of current information in the more general framework and describe the properties of event sets in a POI variant. The general optimality conditions are adopted to the specific case and the algorithm emerges from that naturally. We then proceed with the complexity analysis and point out the importance of finding good approximations for the ORP problems.

2.4.1 Algorithm DOT-SPI

We have a network as described in subsection 2.2.1. We seek to find the least expected travel times from all nodes at all departure times with all possible current information to a certain destination node d. We assume that travelers have perfect on-line information about the link travel times. Mathematically speaking, at any time t, any traveler has knowledge of the realizations of $C_{jk,t'}, \forall(j,k) \in A, \forall t' < t$.

We use a different way to represent the concept of current information sets. The current information defined in the framework of the ORP problem is composed of link travel times. This definition is not convenient for the implementation of the algorithm. At each current time t, each possible distinct set of values of $C_{jk,t'}, \forall(j,k) \in A, \forall t' < t$, corresponds to a unique set of v_r, therefore we define a new term as the counterpart of current information in algorithm design. Let $\pi_{jk,t}$ be the realization of $C_{jk,t}$ we have already learned up to the current time. Define the event collection $EV := \{v_r | C^r_{jk,t'} = \pi_{jk,t'}, \forall(j,k) \in A, \forall t' < t$, for a certain $t\}$. This is the set of support point candidates after we collect information at time t. As we collect more information (i.e. t increases), the size of EV remains the same or decreases. When EV becomes a singleton, we obtain a deterministic network and can apply any deterministic dynamic shortest path algorithm. Let $\mathbf{EV}(t)$ be the set of all possible event collections at time t; an element of $\mathbf{EV}(t)$ is an event collection $EV = v_r | C^r_{jk,t'} = \pi_{jk,t'}, \forall(j,k) \in A, \forall t' < t$. Specifically, $EV(K-1) = \{\{v_1\}, \{v_2\}, ..., \{v_R\}\}$. All the possible event collections can be generated in preprocessing. Here are some important facts about the event collection:

- There is no overlap among elements of $EV(t)$ for a given t, so there are at most R event collections at any time t ($|EV(t)| \leq R$). Thus there are at most RK event collections in total.

- Any element of $EV(t)$ is a subset of an element of $EV(t-1)$.

- $|EV(t)| \geq |EV(t-1)|$.

<table>
<tr><td>$t=0$</td><td colspan="10">$v_1,\ v_2,\ v_3,\ v_4,\ v_5,\ v_6,\ v_7,\ v_8,\ v_9,\ v_{10}$</td><td colspan="5">$v_{11},\ v_{12},\ v_{13},\ v_{14},\ v_{15}$</td></tr>
<tr><td>$t=1$</td><td>v_1</td><td colspan="4">$v_2,\ v_3,\ v_4,\ v_5$</td><td colspan="5">$v_6,\ v_7,\ v_8,\ v_9,\ v_{10}$</td><td colspan="5">$v_{11},\ v_{12},\ v_{13},\ v_{14},\ v_{15}$</td></tr>
<tr><td>$t=2$</td><td>v_1</td><td colspan="4">$v_2,\ v_3,\ v_4,\ v_5$</td><td colspan="5">$v_6,\ v_7,\ v_8,\ v_9,\ v_{10}$</td><td colspan="3">$v_{11}, v_{12}, v_{13}$</td><td colspan="2">$v_{14}, v_{15}$</td></tr>
<tr><td>$t=3$</td><td>v_1</td><td colspan="2">v_2, v_3</td><td colspan="2">v_4, v_5</td><td>v_6</td><td>v_7</td><td colspan="2">v_8, v_9</td><td>v_{10}</td><td>v_{11}</td><td colspan="2">v_{12}, v_{13}</td><td colspan="2">v_{14}, v_{15}</td></tr>
<tr><td>$t=4$</td><td>v_1</td><td>v_2</td><td>v_3</td><td colspan="2">v_4, v_5</td><td>v_6</td><td>v_7</td><td>v_8</td><td>v_9</td><td>v_{10}</td><td>v_{11}</td><td colspan="2">v_{12}, v_{13}</td><td colspan="2">v_{14}, v_{15}</td></tr>
<tr><td>$t=5$</td><td>v_1</td><td>v_2</td><td>v_3</td><td>v_4</td><td>v_5</td><td>v_6</td><td>v_7</td><td>v_8</td><td>v_9</td><td>v_{10}</td><td>v_{11}</td><td>v_{12}</td><td>v_{13}</td><td>v_{14}</td><td>v_{15}</td></tr>
</table>

Figure 2-5: A Possible Scheme of Event Collections

A possible scheme of event collections is shown in Figure 2-5. Each row represents time points in an increasing order, i.e. the first row represents the first time point. Each cell in the last row represents a single support point v_r, which means that the network becomes deterministic beyond time period $K-1$. At each time t, cells within the bold boundary form an event collection. For example, at time 0, $\{v_1, ..., v_{10}\}$ is one event collection, and $\{v_{11}, ..., v_{15}\}$ is the other. At time 1, when more link travel time realizations are available, $\{v_1, ..., v_{10}\}$ is split into three event collections $\{v_1\}, \{v_2, ..., v_5\}$, and $\{v_6, ..., v_{10}\}$. Other event collections are obtained similarly.

Let $e_{\mu^*}(j, t, EV)$ be the least expected travel time to the destination node d if the departure from node j happens at time t with the event collection EV. Let $\mu^*(j, t, EV)$ be the arc to take out of node j to realize $e_{\mu^*}(j, t, EV)$. Assume we select arc (j, k) out of node j. At the end of the journey along arc (j, k), we have a new event collection EV' which is one of the possible event collections at time $t + \pi_{jk,t}$. EV' is a random variable and the probability of a certain EV' can be evaluated as follows:

$$Pr(EV'|EV) = \frac{\sum_{r|r\in EV'\cap EV} p_r}{\sum_{r|r\in EV} p_r}, \forall EV' \in \mathbf{EV}(t+\pi_{jk,t}), \forall EV \in \mathbf{EV}(t).$$

Note that $EV' \cap EV = \emptyset$ or EV'.

The optimality conditions for the problem are:

$$\begin{aligned}
& e_{\mu^*}(j,t,EV) \\
= \ & \min_{k\in A(j)}\{\pi_{jk,t} + E_{EV'}[e_{\mu^*}(k,t+\pi_{jk,t},EV')]\} \\
= \ & \min_{k\in A(j)}\{\pi_{jk,t} + \textstyle\sum_{EV\in \mathbf{EV}(t+\pi_{jk,t})} e_{\mu^*}(k,t+\pi_{jk,t},EV') \times Pr(EV'|EV)\}, \forall j \neq d, \\
& e_{\mu^*}(d,t,EV) = 0, e_{\mu^*}(j,t \geq K-1,EV) = e_{\mu^*}(j,K-1,EV) \\
& \forall t \in T, \forall EV \in \mathbf{EV}(t)
\end{aligned}$$

The solution of these equations can be obtained in decreasing order of time, since the evaluation of $e_{\mu^*}(j,t,EV)$ only depends on $e_{\mu^*}(j,t',EV')$, where $t' > t$. At time $K-1$ or beyond, the network becomes deterministic and static, and we can use any deterministic static shortest path algorithm to compute $e_{\mu^*}(j,t,V), \forall j \in N, \forall t \in K-1, \forall EV \in \mathbf{EV}(K-1)$. Denote the algorithm as DOT-SPI (a counterpart of Algorithm DOT in Chabini (1999) [17] to solve a stochastic problem with perfect information). The statement is as follows.

Algorithm DOT-SPI

Step 0: (Construct $EV(t), t = 0, ..., K-1$)

Call Generate_Event_Collection

Step 1: (Initialization)

1.1 Compute $e_{\mu^*}(j,K-1,EV), \forall j \in N, \forall EV \in \mathbf{EV}(K-1)$

1.2 $e_{\mu^*}(j,t,EV) \leftarrow +\infty, \forall j \in N\backslash\{d\}$,

$e_{\mu^*}(d,t,EV) \leftarrow 0$,

$\forall t < K-1, \forall EV \in \mathbf{EV}(t)$

Step 2: (Main Loop)

For $t = K - 1$ down to 0

For each $EV \in \mathbf{EV}(t)$

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

$temp = \pi_{jk,t} +$

$\sum_{EV' \in \mathbf{EV}(t+\pi_{jk,t})} e_{\mu^*}(k, t + \pi_{jk,t}, EV') \times Pr(EV'|EV)\};$

If $temp < e_{\mu^*}(j, t, EV)$

$e_{\mu^*}(j, t, EV) = temp$

$\mu^*(j, t, EV) = k$

Generate_Event_Collection

$D = \{\{v_1, ..., v_R\}\}$

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 $c^r_{jk,t}, \forall r \in S;$

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

such that $c^r_{jk,t}$ is constant over all $r \in S'_i, i = 1, ..., w$ and $\bigcup_i S'_i = S;$

$D' \leftarrow D' \backslash \{S\} \cup \{S'_1, S'_2, ..., S'_w\};$

Next S

$D \leftarrow D'$

Next (j, k)

$EV(t) \leftarrow D;$

Next t

2.4.2 Complexity Analysis

The basic step in Generate_Event_Collection is the division of S into disjoint sets. This can be done by sorting the elements in S in time $\theta(s \ln s)$, where s is the cardinality of S. For a given time and link, all S are mutually exclusive and collectively exhaustive over all support points. Assume that there are u such disjoint sets for a given time and link, $S_1, S_2, ..., S_u$, and $1 \leq u \leq R$. Therefore the sorting of all the u sets takes time $\theta(\sum_{i=1}^{u} s_i \ln s_i) = \theta(\ln \prod_{i=1}^{u} s_i^{s_i}) = O(ln(s_1+s_2+...+s_u)^{s_1+s_2+...+s_u}) = O(R \ln R)$. On the other hand, the sorting has to retrieve all the R support points at least once, so the running time is also $\Omega(R)$. Altogether constructing event collections takes time $O(mKRlnR)$ and $\Omega(mKR)$. Step 1.1 is solving R static shortest path problems, so the running time is $\theta(R \times SSP)$. Step 1.2 takes time $\theta(KRn)$. At a given time t and for a given link (j, k), the evaluation of all $Pr(EV'|EV)$ takes time $\theta(R)$. There are a total of K time periods and m links, so the main loop has a running time of $\theta(mKR)$.

To sum up, Algorithm DOT-SPI has a complexity of $O(mKR \ln R + R \times SSP)$ and $\Omega(mKR + R \times SSP)$. This algorithm is strongly polynomial in R, however R could be an exponential function of m. If the link travel times are highly interdependent, we expect that R is much less than Qm, where Q is the maximum number of support points for a single link travel time distribution, but it is still very likely that R is exponential in m. Other variants with less online information could also have running time exponential in the number of link travel time random variables involved in current information.

In fact, this is a well-known drawback of dynamic programming, the so-called Bellman's "curse of dimensionality". Approximations and heuristics of dynamic programming have been a very active research topic in the research community of dynamic programming and stochastic control for a long time and many encouraging results exists (Bertsekas (2000) [12]), which can be borrowed and adapted to transportation research. The next steps of this research will focus on finding efficient heuristics that perform well in transportation applications.

2.5 Summary

In Chapter 2, we study the optimal routing policy problem in a stochastic time-dependent network. We first survey the literature on this topic, including deterministic routing problems, routing problems in stochastic static networks and routing in stochastic time-dependent networks. This survey reveals that there are a number of variants of the ORP problem in an STD network, and it motivates the establishment of a formal framework for this problem. We then proceed to describe the framework which includes a general description of an STD network, the decision process, the problem statement and the generic optimality conditions. We then present a comprehensive taxonomy based on assumptions on the network stochastic dependency and information access. A discussion of most of the variants is given, and two variants are studied in detail. The first is the no-online-information (NOI) variant which is easy to understand and can be solved in polynomial time. We give the formulation, an algorithm and computational tests for this variant. The second variant studied in detail is the perfect-on-line (POI) information variant, which is particularly pertinent to transportation applications. This variant has never been studied before in the literature. We give a formulation, an algorithm, and results from computational tests. The complexity analysis shows that the algorithm for the POI variant can be prohibitively time-consuming.

2.6 Graphic Output

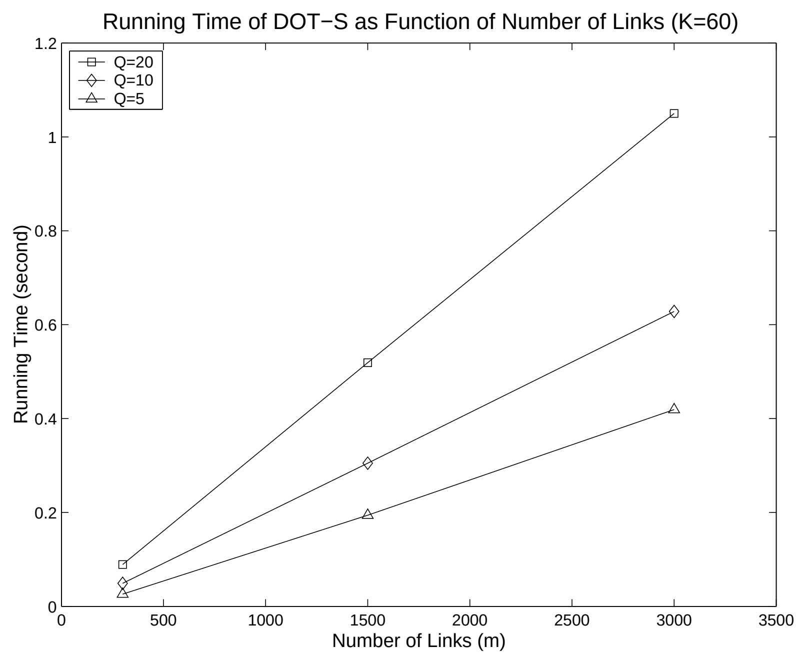


Figure 2-6: Running Time of DOT-S as Function of Number of Links (K=60)

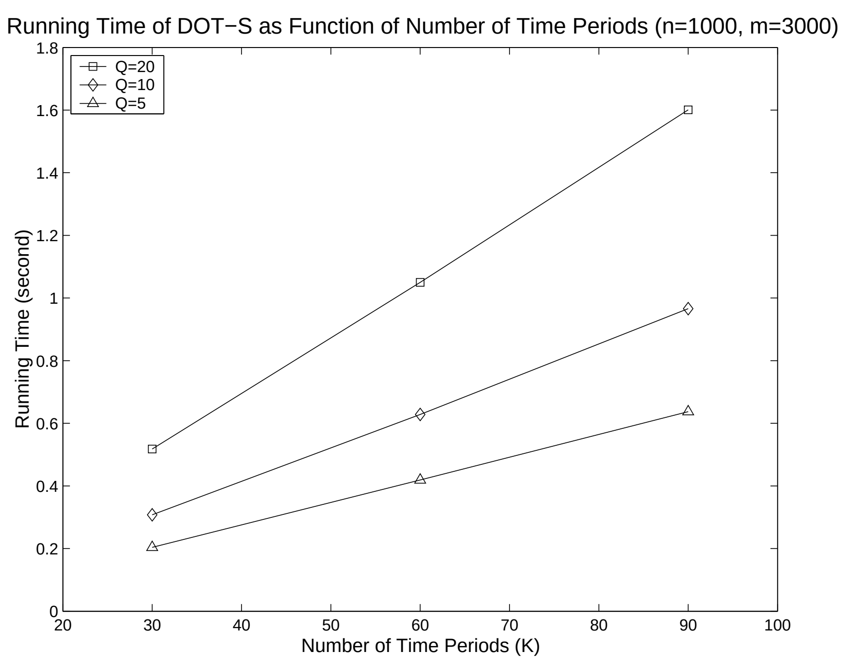


Figure 2-7: Running Time of DOT-S as Function of Number of Time Periods (n=1000, m=3000)

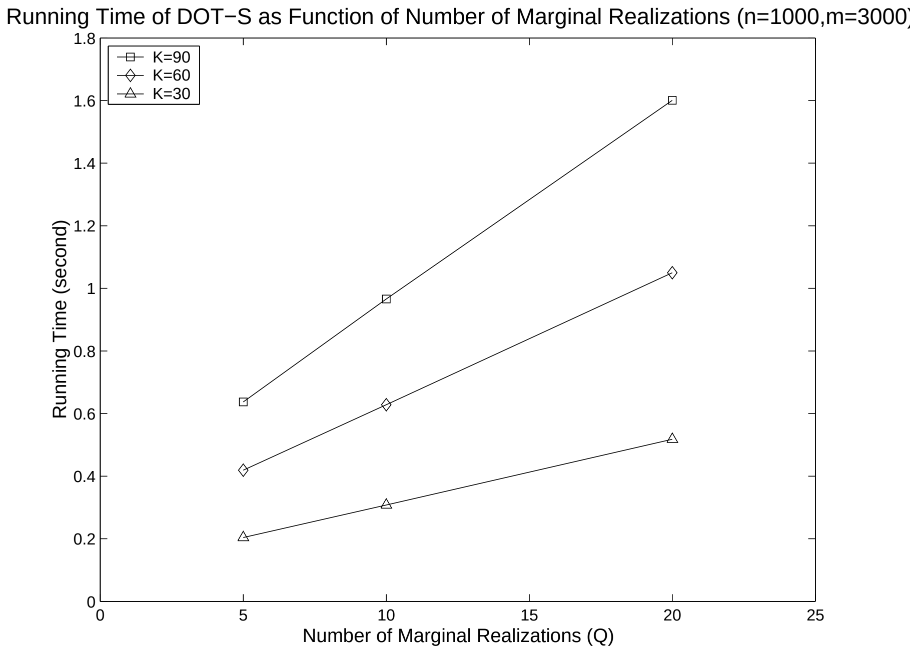


Figure 2-8: Running Time of DOT-S as Function of Number of Support Points (n=1000, m=3000)

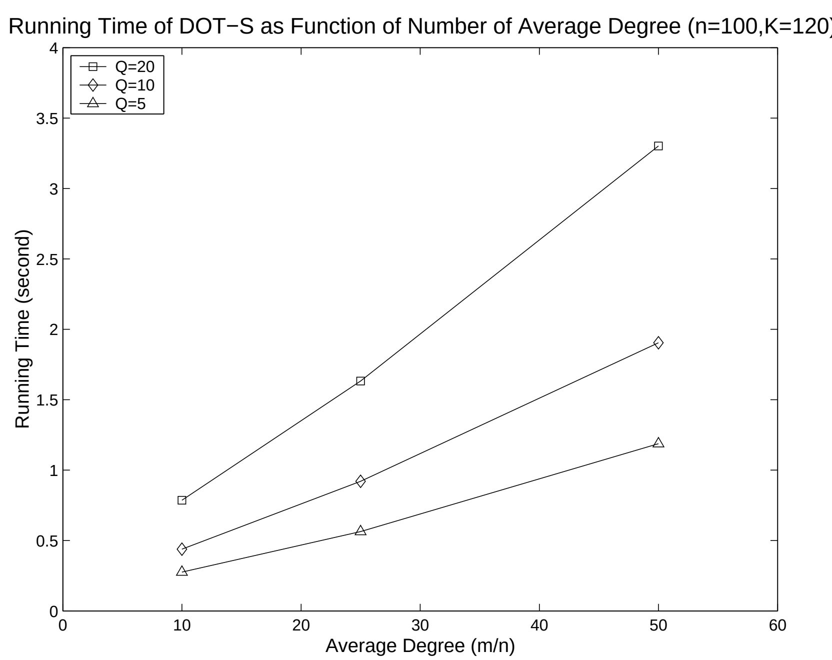


Figure 2-9: Running Time of DOT-S as Function of Average Degree (n=100, K=120)

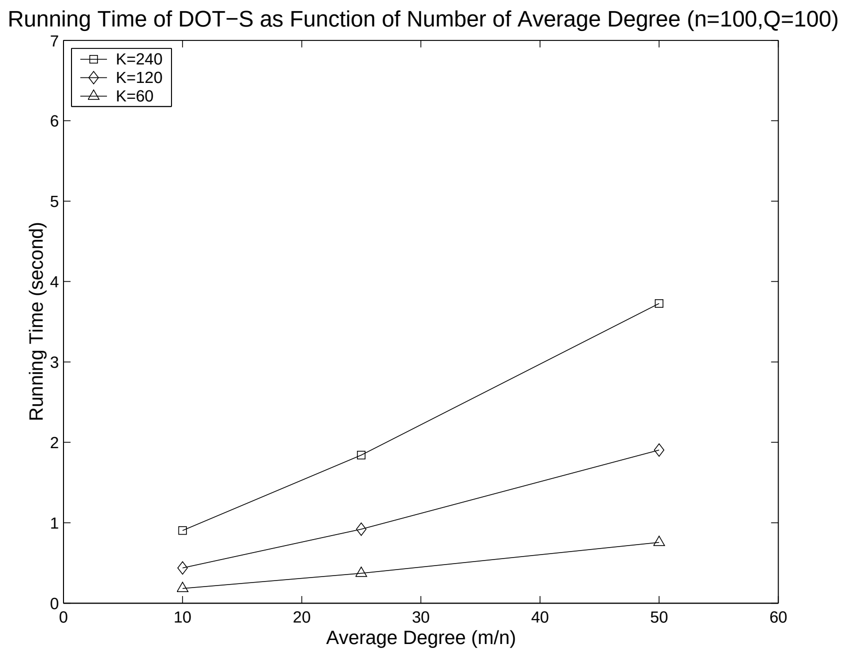


Figure 2-10: Running Time of DOT-S as Function of Average Degree (n=100, Q=10)

Chapter 3

Approximations for ORP Problems in STD Networks

In this chapter, we study approximations to the POI variant introduced in Chapter 2. Four approximations are presented with analysis of their efficiency and effectiveness. This analysis is performed both theoretically and computationally. The computational tests are not comprehensive, but they provide insight into the performance of approximations. Other approximations are suggested, but without computational tests.

3.1 Four Approximations

3.1.1 The Certainty Equivalent (CE) Approximation

The certainty equivalent approximation is most commonly used in traffic applications. The CE approximation replaces every link travel time random variable by its expected value. Thus it transforms the stochastic network into a deterministic network. It then applies any dynamic shortest path problem algorithm (e.g. Algorithm DOT) to obtain an "optimal" path $p^*(j,t)$. Define $CE(j,t)$ as the expected travel time from node j and departure time t when the path $p^*(j,t)$ is taken. The running time of CE is the same as that of a deterministic dynamic shortest path algorithm, but its solution could be arbitrarily worse than the optimal, as shown by the example in Figure 1-1. The "optimal" path output from CE will be path $a - b$. The expected travel time of path $a - b$ is $6 + M/2$, which could be arbitrarily worse than the expected travel time of the optimal routing policy, which is 10.

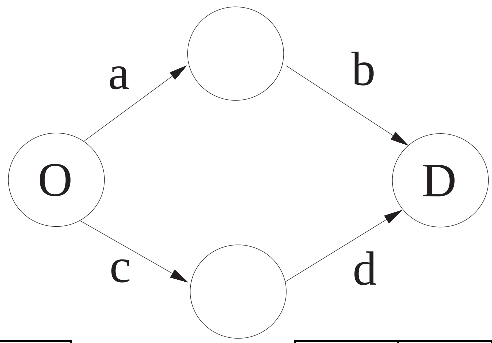
3.1.2 The No-Online-Information (NOI) Approximation

The NOI variant can be solved in polynomial time as stated in the complexity analysis of Section 2.3. The NOI formulation of the ORP problem is valid when the network has time-wise and link-wise independence and the link travel time realizations at the current time are not available. Therefore NOI could serve as a good approximation to POI when the stochastic dependency of link travel times is weak. Note that NOI works with the marginal distributions of link travel times instead of joint distributions. Define $NOI(j, t)$ as the expected travel time from node j at departure time t when the routing policy output from the NOI approximation is applied. However, the performance of NOI as an approximation can also be arbitrarily worse than the optimal. We will not prove this directly, rather it can be proved as a byproduct of the following statement.

$NOI(j, t)$ can be either greater or less than $CE(j, t)$ for a given network. An intuitive argument is that both the NOI approximation and the CE approximation are working on joint distributions different from the original one. Which one leads to a travel cost farther from the optimal solution depends on the data, as illustrated in the following example.

The network in Figure 3-1 has two support points of link travel time distribution. The corresponding marginal PMF is also provided. The expected link travel times are not listed, as they can be easily computed from the marginal PMF. There is only one O-D pair, and we only study departure time 0. The expected travel time of path $a - b$ is $x(1+1) + y(2+3) = 2x + 5y$, and that of path $c - d$ is $x(1+1) + y(2+1) = 2x + 3y$. As y is positive, the expected travel time of path $a - b$ is greater than that of path $c - d$.

Now let us see how the NOI approximation and CE approximation will make the



Time	Link	v_1	v_2
0	a	1	2
	b	n/a	n/a
	c	1	2
	d	n/a	n/a
1	a	n/a	n/a
	b	1	1
	c	n/a	n/a
	d	1	1
2	a	n/a	n/a
	b	1	3
	c	n/a	n/a
	d	3	1

Joint Distribution
$(p_1 = x, p_2 = y = 1 - x)$

Time	Link	Travel Times
0	a	$1(w.p.\ x), 2(w.p.\ y)$
	b	n/a
	c	$1(w.p.\ x), 2(w.p.\ y)$
	d	n/a
1	a	n/a
	b	$1(w.p.\ 1)$
	c	n/a
	d	$1(w.p.\ 1)$
2	a	n/a
	b	$1(w.p.\ x), 3(w.p.\ y)$
	c	n/a
	d	$3(w.p.\ x), 1(w.p.\ y)$

Marginal Distributions
$(x + y = 1)$

Figure 3-1: CE vs. NOI: The Network

routing decisions. When the NOI approximation is applied, we work on the marginal distribution instead of the joint distribution. The "expected travel time" of path $a-b$ computed from NOI would be $x(1+1)+y(2+x+3y)$, and that of path $c-d$ would be $x(1+1)+y(2+3x+y)$. The difference between the expected travel times of path $a-b$ and path $c-d$ in NOI is $2y(y-x)$. Note that in this example, the routing policy from NOI reduces to paths, due to the special topology of the network.

Let $x = 3/4, y = 1/4$, then NOI will choose path $a-b$. Assume the travel time of link b at time 5/4 is greater than the travel time of link d at time 5/4, then CE will choose path $c-d$. In this case, CE is better than NOI. Let $x = 1/4, y = 3/4$, then NOI will choose path $c-d$. Assume the travel time of link b at time 7/4 is less than the travel time of link d at time 7/4, then CE will choose path $a-b$. In this case, NOI is better than CE. Note that we use fractions in departure times only to minimize the effort in presenting data. Actually one can always multiply the existing data by a large enough number to obtain integral data.

Since NOI can have worse solutions than CE, and CE can have solutions that are arbitrarily worse than the optimal, NOI also can have solutions that are arbitrarily worse than the optimal.

3.1.3 The Open Loop Feedback with Certainty Equivalent Approximation (OLFCE)

OLFCE is an improved certainty equivalent approximation. At each decision node, travelers employ a CE that replaces every link travel time random variable at later times by its expected value conditional on the network conditions realized so far. Travelers follow the resulting "optimal" path until a new decision node is reached. At that time, a CE is applied again, conditional on the updated network conditions. Define $OLFCE(j,t)$ as the expected travel time from node j and departure time t when the series of "optimal" paths generated by the open loop feedback with CE approximation are followed. The running time of the OLFCE is $min(K,R)$ times the time to solve one CE, and the performance of OLFCE can be arbitrarily worse than the optimal. To obtain an example to show this, we can set the conditional link travel times as those used to prove the performance of CE could be arbitrarily worse than the optimal.

3.1.4 The Open Loop Feedback with No-Online-Information Approximation (OLFNOI)

Both CE and OLFCE have appeared in the literature for some time. The transition from CE to OLFCE suggests a new approximation OLFNOI developed from the NOI approximation. Similar to OLFCE, at each decision node, travelers employ an NOI approximation that works on the marginal distributions of link travel times conditional on the network conditions realized so far. Travelers follow the resulting "optimal" routing policy until a new decision node is reached. At that time, an NOI approximation is applied again, conditional on the updated network conditions. It is conjectured that OLFNOI will perform at least as well as NOI. However, its performance can still be arbitrarily worse than the optimal.

Similar to the relationship between CE and NOI, results from OLFCE could be either greater or less than results from OLFNOI. To obtain an example to show this, we can set the conditional link travel times as those used to prove $CE(j,t)$ can be either greater or less than $NOI(j,t)$.

3.1.5 Theoretical Study of DOT-SPI vs. Approximations

We define $POI(j,t)$ as the expected travel time from node j at departure time t when the optimal routing policy obtained from Algorithm DOT-SPI is applied. Any routing policy generated by CE, NOI, OLFCE, or OLFNOI is a feasible routing policy for the perfect online information variants. For example, the routing policy generated by CE can be viewed as a routing policy in the POI variant, such that $\mu^*(j,t,EV)$ is constant over all $EV \in \mathbf{EV}(t)$, and all $t \in T$. The routing policy generated by NOI can also be viewed as a routing policy in the POI variant, such that $\mu^*(j,t,EV)$ is constant over all $EV \in \mathbf{EV}(t)$. As Algorithm DOT-SPI solves the POI variant, $POI(j,t)$ is no greater than any one of $CE(j,t), NOI(j,t), OLFCE(j,t), OLFNOI(j,t), \forall j \in N, \forall t \in T$.

3.2 Computational Tests

There is a trade-off between effectiveness and efficiency for all approximations, i.e. an approximation could have satisfactory running times, but its results could be arbitrarily worse in absolute value than those obtained by running the exact algorithm. The effectiveness of approximations depends heavily on the specific application.

In this section, computational tests are designed to study the effectiveness of the four approximations presented in the previous subsection. Algorithms and approximations are run on randomly generated networks. The optimal results from Algorithm DOT-SPI are used as a benchmark. The percent relative difference between approximation results and Algorithm DOT-SPI results is used as the measure of effectiveness. Various parameters that may affect the relative difference are checked. Due to the tremendous computational burden, the results presented in this section are only preliminary. Further tests would need to be done to test the approximations in a larger variety of networks and with broader ranges of parameters.

3.2.1 The Random Network Generator

The computational tests are conducted on randomly generated networks. A multivariate normal distribution is assumed for the joint distribution of all link travel time random variable. The random network generator takes as input: 1) the number of nodes, 2) the number of links, 3) the number of time periods, 4) the number of support points, 5) the common link travel time mean, 6) the common standard deviation of link travel times, 7) the common correlation coefficient of link travel times, 8) the maximum in-degree, and 9) the maximum out-degree.

The topology of the network is randomly generated. The last node is the default destination node. An in-tree rooted at the destination node is generated to ensure the connectivity to the destination node. The remaining links are generated randomly, respecting the maximum in-degree and out-degree.

The joint distribution of all link travel times are generated by a routine that can generate samples from a multivariate normal distribution. The number of random variables is the number of links times the number of time periods. A common mean travel time and a common standard deviation are used for every link travel time random variable. A common correlation coefficient is used for every pair of random variables. The standard deviation should be carefully chosen so that most of the sample values are positive. In cases of a negative value being generated, the absolute value is taken. When the link travel times are read by algorithms or approximations, they are rounded to the nearest integer. The probability of each support point is obtained by first generating R numbers between 0 and 1, and then normalizing the R numbers by their sum.

3.2.2 The Measure of Effectiveness

We study the percent relative difference between results from Algorithm DOT-SPI and results from the four approximations. The definition of the percent relative difference is as follows:

$$\Delta_{approximation} = \frac{\sqrt{\sum_{i=0}^{K-1} \sum_{j=1}^{N} (POI(j,t) - approximation(j,t))^2}}{\sqrt{\sum_{i=0}^{K-1} \sum_{j=1}^{N} (POI(j,t))^2}}$$

where *approximation* can take the value CE, NOI, OLFCE, or OLFNOI. In the computation of percent relative differences, the weight of each (j,t) pair is the same. This implies that we assume the demand to the destination node is distributed evenly across both space and time.

We study the magnitude of the percent relative differences as a function of four different parameters: the common standard deviation of link travel times, the common correlation coefficient of link travel times, the number of support points, and the average in- and out-degrees.

The implemented approximations are just proxies for their real-life counterparts. For example, the expected link travel times for CE or OLFCE should come directly from observation or other estimation methods, not from taking expectation over the joint distribution. Similarly, the marginal distributions of link travel times for NOI or OLFNOI should also come directly from observation or other estimation methods, rather than from aggregating joint distribution. The possible bias between the observed expected link travel times (marginal distributions) and the computed ones from the joint realizations may further complicate the assessment of performance of approximations.

3.2.3 Test Design and Results

Algorithm DOT-SPI and approximations CE, NOI, OLFCE, OLFNOI are implemented in C++ and run on a Dell OptiPlex GX110 workstation with 933 MHz CPU speed and 256 megabytes RAM and under Red Hat Linux 7.0. A test is defined as obtaining results from the five implemented algorithm/approximations for a given combination of the input data to the random network generator. Each test is composed of ten identical runs. The average over the ten runs is taken as the result of this particular test.

The results of the tests are shown in Figures 3-2 through 3-5. The upper graph in each figure shows the results for all four approximations. The lower graph in each figure shows the results for only the two open loop feedback approximations, as they are different in scale from the other two. There are some general observations for all the tests. The magnitude of the percent relative difference for CE and NOI is around 10, and that for OLFCE and OLFNOI is very close to zero. This supports the argument that OLFCE always performs better than CE and OLFNOI always performs better than NOI. The performance of the two open loop feedback approximations is very close to that of Algorithm DOT-SPI, partly due to the small number of support points. When the number of support points is small, the value of online information is large and the network becomes deterministic very soon after the starting time point. Since in a deterministic network, CE, NOI and Algorithm DOT-SPI give the same expected travel times, it is not surprising that OLFCE and OLFNOI have very similar results to Algorithm DOT-SPI. Another interesting observation is that generally NOI performs better than CE, and OLFNOI better than OLFCE, although the theoretical study shows that CE could perform better than NOI and OLFCE could perform better than OLFNOI. This is not surprising, as NOI outputs routing policies that make use of the information on arrival times, while CE outputs paths that ignore any information one may obtain online. It is conjectured that when the average number of possible next nodes is small, as in the example of Figure 3-1, the performance of CE and NOI is close, since routing policies would generally reduce to paths in this situation.

Figure 3-2 shows the percent relative difference as an increasing function of the common standard deviation of link travel times. Note the mean link travel time is also common across time and space. When the standard deviation is large, the link travel times are more dispersed, and thus the expected travel times of different paths (routing policies) are more likely to differ. This magnifies the difference between optimal and sub-optimal solutions. Figure 3-3 shows the percent relative difference as a decreasing function of the common correlation coefficient of link travel times. This phenomenon can be explained by the same logic used in Figure 3-2. A positive correlation coefficient of random variables X and Y provides a measure of the extent to which the signs of $x - E[X]$ and $y - E[Y]$ "tend" to be positive. As we have a common mean for all link travel times, the positive correlation coefficient actually indicates how close x and y are. When link travel times are close, the difference between optimal and sub-optimal solutions is reduced. Figure 3-4 shows the percent relative difference as a function of the number of support points. The number of support points represents, among others, the extent of discretization. There is no definite relationship shown in the figure. Further computational tests are needed to study the effect of discretization in a larger range. Figure 3-5 shows the percent relative difference as an increasing function of average in-degree and out-degree. The two degrees are set to be equal in the tests. As the average degree increases, the travelers have more choices for the next node. Therefore more paths are involved in an optimal routing policy, and the optimal routing solutions have a greater chance to achieve lower travel times than the sub-optimal solutions.

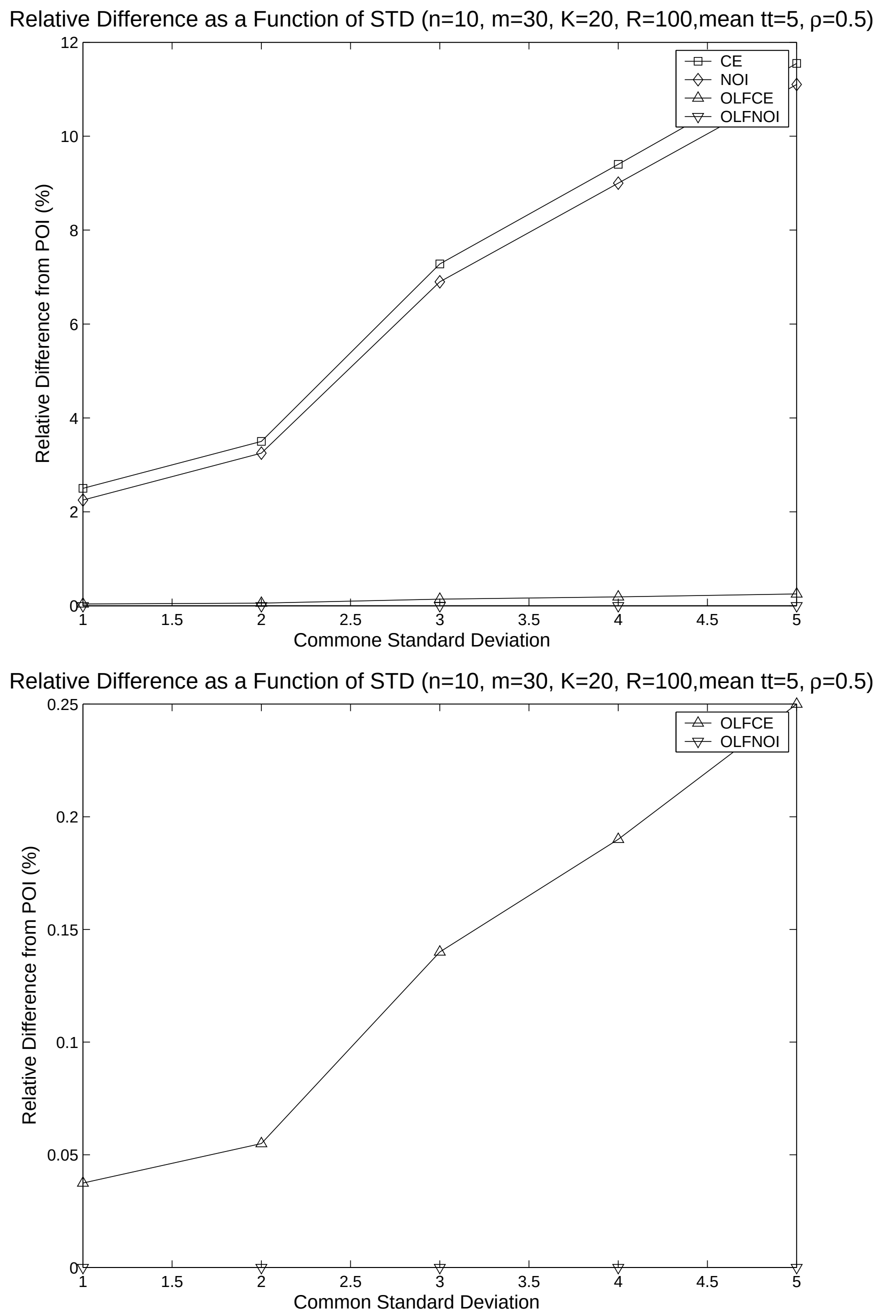
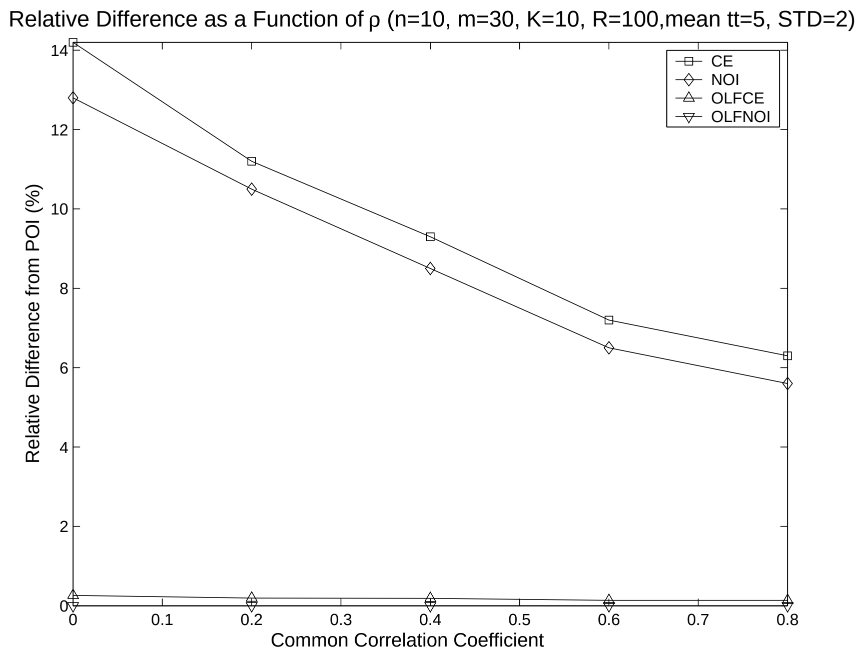


Figure 3-2: Percent Differences of Approximations Relative to Algorithm DOT-SPI as Functions of the Common Standard Deviation of Link Travel Times (with 10 nodes, 30 links, 20 time periods, 100 support points, 10 as the common mean link travel time, and 0.5 as the common correlation coefficient of link travel times



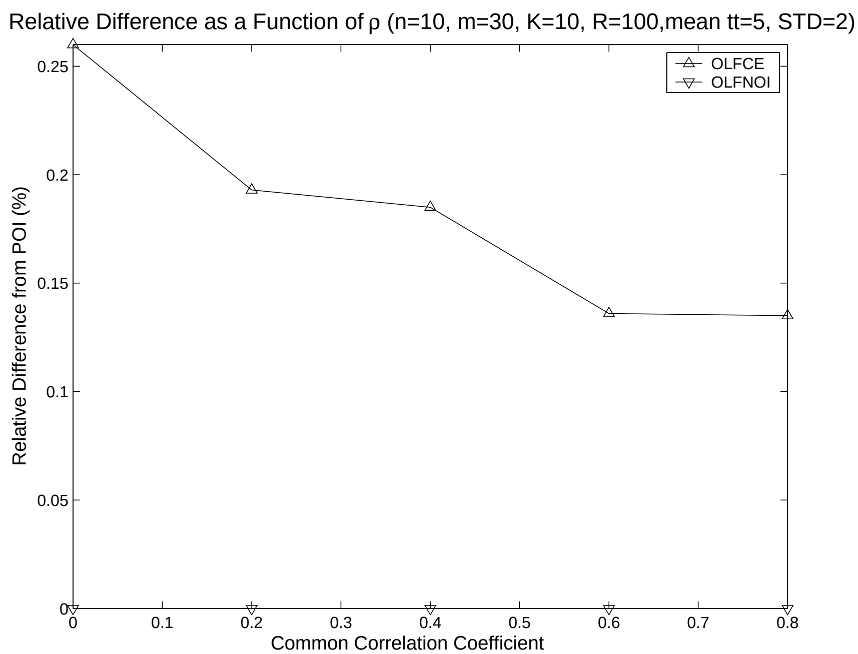


Figure 3-3: Percent Differences of Approximations Relative to Algorithm DOT-SPI as Functions of the Common Correlation Coefficient of Link Travel Times (with 10 nodes, 30 links, 10 time periods, 100 support points, 5 as the common mean link travel time, and 1 as the common standard deviation of link travel times

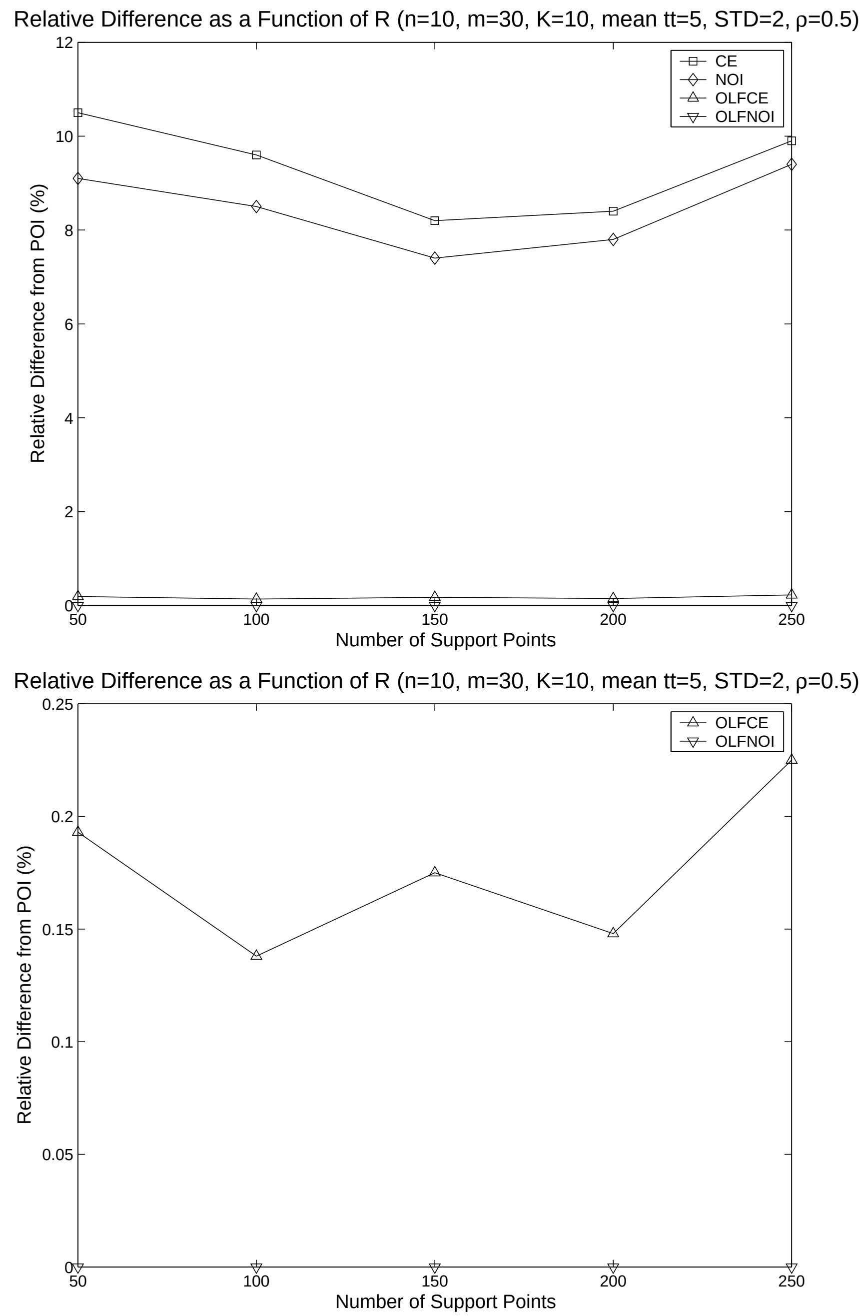
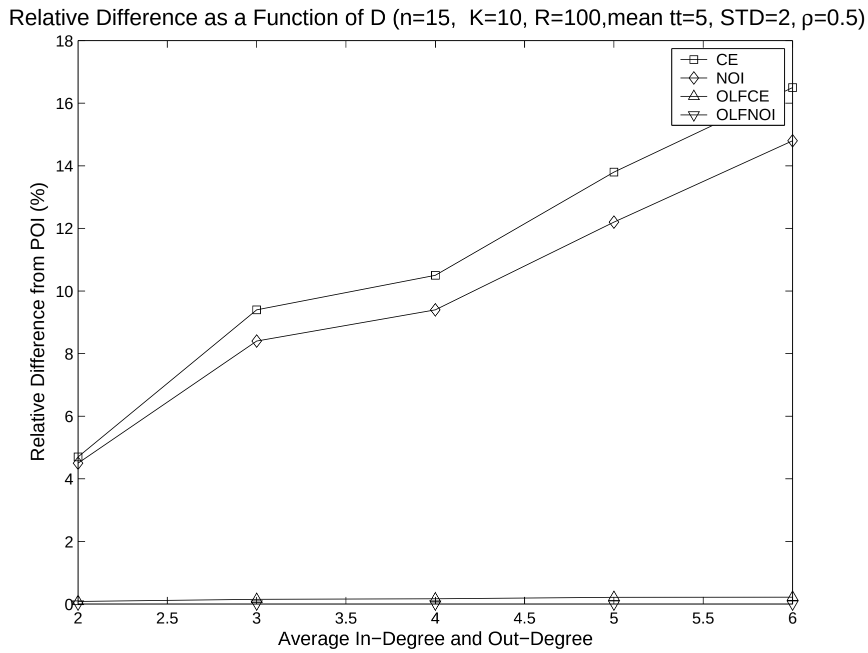


Figure 3-4: Percent Differences of Approximations Relative to Algorithm DOT-SPI as Functions of the Number of Joint Support Points (with 10 nodes, 30 links, 10 time periods, 5 as the common mean link travel time, 2 as the common standard deviation of link travel times, and 0.5 as the common correlation coefficient of link travel times



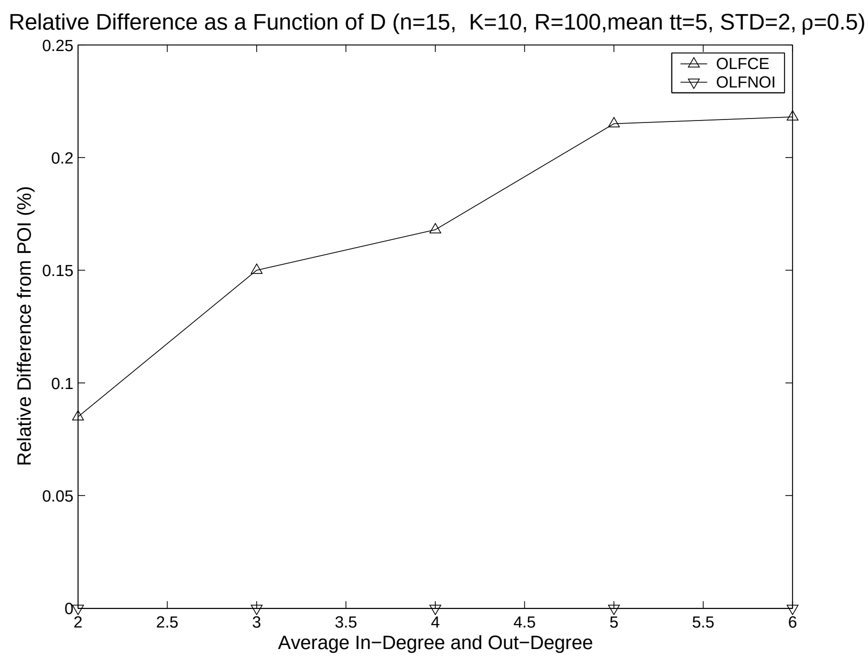


Figure 3-5: Percent Differences of Approximations Relative to Algorithm DOT-SPI as Functions of the Average In-Degree and Out-Degree (with 15 nodes, 10 time periods, 100 support points, 5 as the common mean link travel time, 2 as the common standard deviation of link travel times, 0.5 as the common correlation coefficient of link travel times, and 2 times the average in- and out-degree as the maximum in- and out-degree

3.3 Summary

The complexity analysis in Chapter 2 shows that the algorithm for the POI variant can be prohibitively time-consuming. Therefore approximations are developed in this chapter. Four approximations are presented with an analysis of their efficiency and effectiveness. This analysis is done both theoretically and computationally. There is a trade-off between effectiveness and efficiency for all approximations, i.e. they could have satisfactory running times, but their results could be arbitrarily worse in absolute value than those obtained by running the exact algorithm. The computational tests are not comprehensive, but they provide insight into the performance of approximations.

Chapter 4

Optimal Routing Policy Problems Considering Travel Time Reliability

In this chapter, we present algorithms to compute optimal routing policies with two travel time reliability criteria: variance and expected schedule delay. In the past two chapters, we have been focusing on the study of minimum expected travel time policies, as expected travel time is the primary concern of travelers in making routing decisions. On the other hand, when faced with uncertainty, travelers are also concerned about the reliability of their travel times. For example, unreliable travel times will cause anxiety or cause disutility to travelers because of unexpected late arrival at their destinations. We use travel time variance and expected schedule delay to represent travel time reliability. A routing policy with less travel time variance or less expected schedule delay is viewed as more reliable. Schedule delay is defined as the difference between the actual arrival time and the desired arrival time. For commuters, the desired arrival time in the morning might be some time around the work starting time. For a traveler catching a plane, the desired arrival time might be roughly one hour before the plane departure. It is generally believed and verified by some empirical studies that both early and late arrivals cause disutility to the user. For example, although late arrival at the workplace would cause trouble for a commuter, an arrival too early would also make the commuter feel a waste of time. Therefore in this chapter, we will consider both early schedule delay and late schedule delay, allowing different penalties to be associated with each.

There is a difference between the algorithm design in this chapter and that for the minimum expected travel time policy problems. Since expected travel time is the primary criterion in routing optimization, and reliability measures (variance or expected schedule delay) are secondary, it is not necessary to design algorithms that minimize variance or expected schedule delay only. Instead, we will design algorithms that minimize a linear combination of expected travel time and travel time variance (or expected schedule delay). Therefore in Section 4.1 and Section 4.2, we will develop formulas that describe the relationship between an attribute (variance or expected schedule delay) at a given state (j, t, I) and the attributes at succeeding nodes. Then we present the optimal condition for the policy that minimizes the specific attribute. Please note that these formulas describe generic relationships and generic optimality conditions, in that the network stochastic dependency and information access are not specified. Furthermore, we find that the Bellman's principle of optimality does not hold for the minimum variance policy problem, and thus no exact optimality conditions are available. Instead, an approximation is presented. These two sections provide a theoretical base for the algorithm design in Section 4.3, where the optimal condition for minimizing a linear combination of expected travel time, expected early schedule delay and expected late schedule delay is provided and an algorithm designed.

Throughout the chapter, we adopt the definitions of a stochastic time-dependent (STD) network and a routing policy given in Subsection 2.2.1. The basic difference lies in the optimization criterion, and therefore leads to differences in the minimization problem definition and optimality conditions. Please refer to the subsections "The Network" and "The Decision Process" in Subsection 2.2.1 for details of some basic definitions, as we will not repeat them here. Note that the taxonomy (see Subsection 2.2.2) is also applicable to the problems studied in this chapter, as the taxonomy has nothing to do with the optimization criteria.

4.1 Minimum Variance Policy Problem

4.1.1 The Minimization Problem

As we know, the travel time of a routing policy for a given initial state in a STD network is a random variable. Denote $Var[X]$ as the variance of a random variable X. The **minimum variance routing policy problem in a stochastic time-dependent network** with one destination node d is to find μ^*, such that

$$\mu^* = \arg\min_{\mu}\{Var_{\{x_0,x_1,\ldots,x_S\}\in M(x_0,\mu)}[t_{x_S} - t_{x_0}]\}, \quad \forall x_0 \tag{4.1}$$

The random variable to take variance of is $t_{x_S} - t_{x_0}$, the travel time from the origin as defined in the initial state x_0 to the destination node d for a given routing policy μ. The variance is calculated over all possible state chains, $M(x_0, \mu)$. The minimum is taken over all routing policies.

4.1.2 Variance of a Routing Policy

Before presenting the optimality conditions, we try to find the recursive relationship between variances of a given routing policy starting from two adjacent nodes. This relationship is much more involved than that for the expected travel time of a routing policy. As we know, the expected travel time of a routing policy can be decomposed into two parts: one is the expected travel time of the next link, and the other is the expected travel time from the next state (whose current-node is the next node) to the destination. Let $e_\mu(x)$ denote the expected travel time to the destination node d when the initial state is x and the routing policy μ is applied. Define $A(j)$ as the set of adjacent nodes of node j, $\tilde{C}_{jk,t}|I$ as a travel time random variable of link (j,k) at time t conditional on current-information I, and $\tilde{I}'|I$ as a current-information random variable at the next node k and at time $t + \tilde{C}_{jk,t}|I$. Then we have

$$
\begin{aligned}
e_\mu(j,t,I) \quad &= \quad E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + E_{\tilde{I}'}[e_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|\tilde{C}_{jk,t}]|I] \\
& \quad\quad \text{where } k = \mu(j,t,I)
\end{aligned} \tag{4.2}
$$

Next we develop the recursive equation for the variance of a routing policy. We define additional variables as follows. (Recall that a symbol with a $\sim$ over it is a random variable, while the same symbol without the $\sim$ is one specific realization of the random variable.) Unless otherwise indicated, all routing decisions are made to reach a single destination d.

$\tilde{T}_\mu(j,t,I)$: travel time of policy μ from state (j,t,I)

$v_\mu(j,t,I)$: travel time variance of policy μ from state (j,t,I)

: $v_\mu(j,t,I) = Var[\tilde{T}_\mu(j,t,I)]$

$p_{C_{jk,t}}$: probability that $\tilde{C}_{jk,t}$ takes the value $C_{jk,t}$

$p_{I'|C_{jk,t}}$: probability that $\tilde{I}'|C_{jk,t}$ takes the value I'

$k = \mu(j,t,I)$

In the following mathematical development, all the calculations are conditional on the current-information I. To avoid heavy notation, we omit I in the equations. The major theorem we use is the Law of Conditional Variances:

$$
Var(\tilde{X}) = E[Var[\tilde{X}|\tilde{Y}]] + Var[E[\tilde{X}|\tilde{Y}]]
$$

where $\tilde{X}$ and $\tilde{Y}$ are random variables. Note that $E[\tilde{X}|\tilde{Y}]$ and $Var[\tilde{X}|\tilde{Y}]$ are also random variables. $E[\tilde{X}|Y]$ is a value, which is the expected value of $\tilde{X}$ given that $\tilde{Y} = Y$. $Var[\tilde{X}|Y]$ is a value, which is the variance of $\tilde{X}$ given that $\tilde{Y} = Y$.

Since $\tilde{T}_\mu(j,t,I) = \tilde{C}_{jk,t} + \tilde{T}_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')$, we have

$$
\begin{aligned}
v_\mu(j,t,I) \quad &= \quad Var[\tilde{T}_\mu(j,t,I)] \\
&= \quad E[Var[\tilde{T}_\mu(j,t,I)|\tilde{C}_{jk,t}]] + Var[E[\tilde{T}_\mu(j,t,I)|\tilde{C}_{jk,t}]] \\
&= \quad \textstyle\sum_{C_{jk,t}} p_{C_{jk,t}} \times Var[\tilde{T}_\mu(j,t,I)|C_{jk,t}] + \\
& \quad\quad Var[E[\tilde{T}_\mu(j,t,I)|\tilde{C}_{jk,t}]]
\end{aligned} \tag{4.3}
$$

The first equality is due to the definition of $v_\mu(j,t,I)$. The second equality is due to the Law of Conditional Variances. The third equality is due to the definitions of expected value and variance of a random variable.

Next we will compute the individual components of the right hand side of the last line in Equation 4.3 one by one.

We apply the Law of Conditional Variances again to obtain

$$Var[\tilde{T}_\mu(j,t,I)|C_{jk,t}] \quad = \quad E[Var[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|\tilde{I}']] + Var[E[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|\tilde{I}']] \tag{4.4}$$

For the first component of the right hand side of Equation 4.4, we have

$$\begin{aligned} Var[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|I'] &= Var[C_{jk,t} + \tilde{T}_\mu(k,t+C_{jk,t},I')] \\ &= Var[\tilde{T}_\mu(k,t+C_{jk,t},I')] \\ &= v_\mu(k,t+C_{jk,t},I') \end{aligned} \tag{4.5}$$

The first equality is due to decomposition of travel time from (j,t,I) into two parts. The second equality is due to the fact that $C_{jk,t}$ is a deterministic value and thus does not contribute to the variance of $\tilde{T}_\mu(j,t,I)$. The third equality is due to the definition of $v_\mu()$.

Therefore we have

$$E[Var[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|\tilde{I}']] = \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times v_\mu(k,t+C_{jk,t},I') \tag{4.6}$$

For the second component of the right hand side of Equation 4.4, we have

$$E[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|I'] = C_{jk,t} + e_\mu(k,t+C_{jk,t},I') \tag{4.7}$$

and therefore the expectation can be evaluated as follows:

$$\begin{aligned} E[E[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|\tilde{I}']] &= \textstyle\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (C_{jk,t} + e_\mu(k, t + C_{jk,t}, I')) \\ &= C_{jk,t} + \textstyle\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (e_\mu(k, t + C_{jk,t}, I')) \end{aligned} \tag{4.8}$$

Substituting Equation 4.7 and 4.8 into the second component of the right hand side of Equation 4.4, we obtain

$$\begin{aligned} & Var[E[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|\tilde{I}']] \\ = & \textstyle\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (E[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|I'] - E[E[(\tilde{T}_\mu(j,t,I)|C_{jk,t})|I']])^2 \\ = & \textstyle\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (e_\mu(k, t + C_{jk,t}, I') - \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_\mu(k, t + C_{jk,t}, I'))^2 \end{aligned} \tag{4.9}$$

Substituting Equation 4.6 and 4.9 into Equation 4.4, we obtain

$$\begin{aligned} & Var[\tilde{T}_\mu(j,t,I)|C_{jk,t}] \\ = & \textstyle\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times \quad (v_\mu(k, t + C_{jk,t}, I') + (e_\mu(k, t + C_{jk,t}, I') \\ & \qquad - \textstyle\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_\mu(k, t + C_{jk,t}, I'))^2) \end{aligned} \tag{4.10}$$

Now that we have finished developing the first component of the right hand side of the last line of Equation 4.3, let us study the second component.

$$E[\tilde{T}_\mu(j,t,I)|C_{jk,t}] = C_{jk,t} + \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_\mu(k, t + C_{jk,t}, I') \tag{4.11}$$

Therefore the expectation of $E[\tilde{T}_\mu(j,t,I)|C_{jk,t}]$ is evaluated as:

$$\begin{aligned} E[E[\tilde{T}_\mu(j,t,I)|C_{jk,t}]] &= \textstyle\sum_{C_{jk,t}} p_{C_{jk,t}} \times (C_{jk,t} + \sum_{I'|C_{jk,t}} e(k, t + C_{jk,t}, I')) \\ &= e_\mu(j,t,I) \end{aligned} \tag{4.12}$$

and the second component of the right hand side of the last line of Equation 4.3, which is actually the variance of $E[\tilde{T}_\mu(j,t,I)|C_{jk,t}]$, can be evaluated as:

$$\begin{array}{rl} & Var[E[\tilde{T}_\mu(j,t,I)|\tilde{C}_{jk,t}]] \\ = & \sum_{C_{jk,t}} p_{C_{jk,t}} \times (E[\tilde{T}_\mu(j,t,I)|\tilde{C}_{jk,t}] - E[E[\tilde{T}_\mu(j,t,I)|\tilde{C}_{jk,t}]])^2 \\ = & \sum_{C_{jk,t}} p_{C_{jk,t}} \times (C_{jk,t} + \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_\mu(k, t + C_{jk,t}, I') - e_\mu(j,t,I))^2 \end{array} \tag{4.13}$$

Substituting Equation 4.10 and 4.13 into Equation 4.3, we obtain the final result:

$$\begin{array}{rll} & v_\mu(j,t,I) & \\ = & \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (v_\mu(k, t + C_{jk,t}, I') \\ & & +(e_\mu(k, t + C_{jk,t}, I') - \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_\mu(k, t + C_{jk,t}, I'))^2)) \\ & + & \\ & \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (C_{jk,t} + \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_\mu(k, t + C_{jk,t}, I') - e_\mu(j,t,I))^2 \\ & v_\mu(d,t,I) = 0 & \end{array} \tag{4.14}$$

Please note that all calculations are conditional on the current-information I.

The above equations are for a general definition of current-information. If the current-information is an empty set, i.e. the problem becomes a no-online-information variant (see Section 2.2.2 and Section 2.3 for detailed discussion of this variant), then the equations reduce to:

$$\begin{array}{rll} & v_\mu(j,t) & \\ = & \sum_{C_{jk,t}} p_{C_{jk,t}} \times & v_\mu(k, t + C_{jk,t}) \\ & + & \\ & \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (C_{jk,t} + e_\mu(k, t + C_{jk,t}) - e_\mu(j,t))^2 \\ & v_\mu(d,t) = 0 & \end{array} \tag{4.15}$$

We can see that the equations are much simpler for the NOI variant. Intuitively, we can view the first part as the variance from the next node to the destination, and the second part as the variance induced by including the next link in the policy.

4.1.3 An Approximation Method

The relationship between the variance from the current state and that from the next state is very complex. Bellman's principle of optimality relies largely on the additive structure of the costs, which is not the case for the variance of a policy. We thus hypothesize that the principle does not hold for the minimum variance routing policy problem in a STD network. This is to say that if policy μ has minimum variance starting from state (j, t, I), and $\mu(j, t, I) = k$ and the next state is (k, t', I'), it is not necessarily true policy μ has minimum variance from state (k, t', I'), $\forall t'$ and $\forall I'$ that are possible at node k. An illustrative example is presented below.

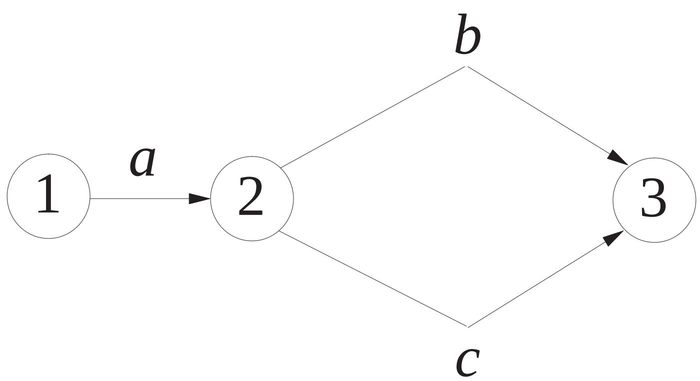


Figure 4-1: The Network (Principle of Optimality for Minimum Variance Problem)

The network has three links and is static. The joint distribution of the link travel times are as follows.

$$(t_a, t_b, t_c) = \begin{cases} 1, 2, 9, & w.p.\ 1/3 \\ 1, 3, 9, & w.p.\ 1/3 \\ 5, 0, 9, & w.p.\ 1/3 \end{cases}$$

As we can see, link a has two possible travel times: 1 and 5. If the value of travel time of link a is 1, then link b can take two different travel times: 2 and 3. If the value of travel time of link a is 5, then link b is deterministic with a travel time 0. Link c is deterministic under any circumstance with a travel time 9. Assume that users learn the travel time realization of a link only after traversing it. Node 3 is the destination node.

There are four policies from node 1 which are shown in Table 4.1 with their respective variances.

	At node 1	At node 2		Variance
		If $t_a = 1$	If $t_b = 5$	
policy 1	link a	link b	link b	1
policy 2	link a	link b	link c	37
policy 3	link a	link c	link b	8.3
policy 4	link a	link c	link c	5.3

Table 4.1: Four Possible Policies from Node 1

The minimum variance policy from node 1 is policy 1, which actually is a path composed of link a and link b. However, if one starts from node 2, taking link c should be the minimum variance policy, as it has zero variance. This example shows that the principle of optimality does not hold for the minimum variance routing policy problem.

It then follows that it is not likely that the problem of minimum variance routing policy in a general stochastic time-dependent network can be solved in polynomial time (on network size and the number of support points R). We will then propose an approximation method to solve this problem. This approximation makes use of Equation 4.14 and acts as if Bellman's Principle of Optimality applies to the minimum variance policy problem.

The approximated minimum variance policy μ^* and the corresponding approximated minimum variance v_{μ^*} for any initial state (j, t, I) can be obtained by solving the following system of equations.

$$v_{\mu^*}(j,t,I) = \min_{k\in A(j)} \left\{ \begin{array}{ll} \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (v_{\mu^*}(k, t+C_{jk,t}, I') \\ & +(e_{\mu^*}(k,t+C_{jk,t},I') - \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_{\mu^*}(k,t+C_{jk,t},I'))^2)) \\ + & \\ \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (C_{jk,t} + \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_{\mu^*}(k,t+C_{jk,t},I') - e_{\mu^*}(j,t,I))^2 \end{array} \right\} \quad (4.16)$$

$$v_{\mu^*}(j,t,I) = \arg\min_{k\in A(j)} \left\{ \begin{array}{ll} \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (\sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times (v_{\mu^*}(k, t+C_{jk,t}, I') \\ & +(e_{\mu^*}(k,t+C_{jk,t},I') - \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_{\mu^*}(k,t+C_{jk,t},I'))^2)) \\ + & \\ \sum_{C_{jk,t}} p_{C_{jk,t}} \times & (C_{jk,t} + \sum_{I'|C_{jk,t}} p_{I'|C_{jk,t}} \times e_{\mu^*}(k,t+C_{jk,t},I') - e_{\mu^*}(j,t,I))^2 \end{array} \right\} \quad (4.17)$$

$$\begin{aligned} e_{\mu^*}(j,t,I) &= E_{\tilde{C}_{jk^*,t}}[\tilde{C}_{jk^*,t} + E_{\tilde{I}'}[e_{\mu^*}(k^*, t+\tilde{C}_{jk^*,t}, \tilde{I}')|\tilde{C}_{jk^*,t}]|I] \\ & \text{where } k^* = \mu^*(j,t,I) \end{aligned} \quad (4.18)$$

with the boundary conditions: $v_{\mu^*}(d,t,I) = 0, e_{\mu^*}(d,t,I) = 0, \mu^*(d,t,I) = d, \forall t \in T, \forall I \in Z(j,t)$.

To the best of the author's knowledge, there is no paper in the literature that deals with minimum variance path (routing policy) problems in stochastic networks. The study here is a preliminary attempt to tackle this problem. The approximation method would conceivably perform well if the component $v_{\mu^*}(k, t+C_{jk,t}, I')$ dominates other components on the right hand side of Equation 4.16. Numerical experiments are needed to understand the performance of the approximation method.

4.2 Minimum Expected Schedule Delay Policy Problem

4.2.1 The Minimization Problem

Schedule delay is defined as the difference between the desired arrival time and the actual arrival time. For travelers with rigid arrival time requirements, schedule delay is an important factor in their routing choices. A commuter in the morning usually wants to arrive at the work place around the work start time. A delivery person with time-sensitive materials might want to arrive at the delivery destination before a given time.

We distinguish between early schedule delay and late schedule delay. Let $[t^* - \Delta, t^* + \Delta]$ be the desired arrival time range. If the arrival time is t, then early schedule delay is $\max(0, t^* - \Delta - t)$, and late schedule delay is $\max(0, t - (t^* + \Delta))$. In a stochastic time-dependent network, if a user takes a routing policy from the origin at a given departure time, the arrival time at the destination is a random variable. Since the schedule delay (early or late) is a function of the arrival time, it is also a random variable. Therefore we would like to minimize the expected value of the random schedule delay.

The **minimum expected early schedule delay routing policy problem in a stochastic time-dependent network** with one destination node d is to find μ^*, such that

$$\mu^* = \arg\min_{\mu}\{E_{\{x_0,x_1,\ldots,x_S\}\in M(x_0,\mu)}[\max(0, t^* - \Delta - t_{x_S})]\}, \quad \forall x_0 \tag{4.19}$$

The **minimum expected late schedule delay routing policy problem in a stochastic time-dependent network** with one destination node d is to find μ^*, such that

$$\mu^* = \arg\min_{\mu}\{E_{\{x_0,x_1,\ldots,x_S\}\in M(x_0,\mu)}[\max(0, t_{x_S} - (t^* + \Delta))]\}, \quad \forall x_0 \tag{4.20}$$

4.2.2 Expected Schedule Delay of a Routing Policy

We follow the procedure in studying the minimum variance routing policy problem. We first derive the equations that represent the relationship between expected schedule delay of a routing policy at the current state and those at possible next states. We use some additional notation here.

$es_\mu(j,t,I)$: expected early schedule delay of policy μ from initial state (j,t,I)

$ls_\mu(j,t,I)$: expected late schedule delay of policy μ from initial state (j,t,I)

The expected early schedule delay of policy μ can be computed using the following equation:

$$\begin{aligned} es_\mu(j,t,I) &= E_{\tilde{C}_{jk,t},\tilde{I}'}[es_\mu(k,t+\tilde{C}_{jk,t},\tilde{I}')|I] \\ es_\mu(d,t,I) &= \max(0,t^*-\Delta-t) \end{aligned} \tag{4.21}$$

The expected late schedule delay of policy μ can be computed using the following equation:

$$\begin{aligned} ls_\mu(j,t,I) &= E_{\tilde{C}_{jk,t},\tilde{I}'}[ls_\mu(k,t+\tilde{C}_{jk,t},\tilde{I}')|I] \\ ls_\mu(d,t,I) &= \max(0,t-(t^*+\Delta)) \end{aligned} \tag{4.22}$$

4.2.3 The Optimality Condition

We present the optimality conditions for the minimum expected early/late schedule delay problems in this subsection. It can be easily verified that the principle of optimality applies to the minimum expected early/late schedule delay problems.

For $\forall j \in N-\{d\}, \forall t \in T, \forall I \in Z(j,t)$, $es_{\mu^*}(x)$ and μ^* are optimal for the minimum expected **early** schedule policy problem, if and only if they are solutions of the following system of equations:

$$
\begin{aligned}
&es_{\mu^*}(j,t,I) = \min_{\forall k \in A(j)} E_{\tilde{C}_{jk,t},\tilde{I}'}[es_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|I] \\
&\mu^*(j,t,I) = \arg\min_{\forall k \in A(j)} E_{\tilde{C}_{jk,t},\tilde{I}'}[es_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|I] \\
&es_{\mu^*}(d,t,I) = \max(0, t^* - \Delta - t)
\end{aligned}
\tag{4.23}
$$

For $\forall j \in N - \{d\}, \forall t \in T, \forall I \in Z(j,t)$, $ls_{\mu^*}(x)$ and μ^* are optimal for the minimum expected **late** schedule policy problem, if and only if they are solutions of the following system of equations:

$$
\begin{aligned}
&ls_{\mu^*}(j,t,I) = \min_{\forall k \in A(j)} E_{\tilde{C}_{jk,t},\tilde{I}'}[ls_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|I] \\
&\mu^*(j,t,I) = \arg\min_{\forall k \in A(j)} E_{\tilde{C}_{jk,t},\tilde{I}'}[ls_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|I] \\
&ls_{\mu^*}(d,t,I) = \max(0, t - (t^* + \Delta))
\end{aligned}
\tag{4.24}
$$

4.3 Minimization of a Linear Combination of Policy Attributes

We do not present algorithms to compute the optimal policies that minimize the specific criteria (variance, expected early schedule, or expected late schedule). Instead, we present an algorithm for computing a linear combination of relevant attributes of a routing policy. Since the exact optimality conditions for the minimum variance problem are not available, we will exclude the variance from the linear combination. The **problem of optimal routing policy with minimum linear combination of expected travel time, expected early schedule delay and expected late schedule delay problem in a stochastic time-dependent network** with one destination node d and the desired arrival time interval $[t^* - \Delta, t^* + \Delta]$ is to find μ^*, such that

$$\mu^* = \arg\min_\mu \{E_{\{x_0,x_1,\dots,x_S\}\in M(x_0,\mu)}[\quad \alpha(t_{x_S} - t_{x_0}) + \gamma \max(0, t^* - \Delta - t_{x_S}) + \eta \max(0, t_{x_S} - (t^* + \Delta))]\}$$

$$\forall x_0, \alpha \geq 0, \gamma \geq 0, \eta \geq 0 \tag{4.25}$$

The reasons for designing an algorithm for the minimization of a linear combination policy attributes rather than for a specific reliability criterion are three-folded. First, the linear combination problem is more realistic, as the expected travel time is usually the primary concern of travelers in a stochastic time-dependent network, while the reliability criteria are secondary. Second, a linear combination is a reasonable way of combining multiple objectives. Third, the minimum expected schedule delay problems can be viewed as special cases of the minimum linear combination problem, where the coefficients for the expected schedule delays are zero.

We first present the recursive equation to compute the linear combination of attributes of a routing policy. Let $V_\mu(j,t,I)$ be the linear combination of the expected travel time, expected early schedule delay and expected late schedule delay of a routing policy μ and assume $k = \mu(j,t,I)$.

$$\begin{aligned}
V_\mu(j,t,I) &= \alpha E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t} + E_{\tilde{I}'}[e_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|\tilde{C}_{jk,t}]|I] \\
&\quad +\gamma E_{\tilde{C}_{jk,t},\tilde{I}'}[es_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|I] + \eta E_{\tilde{C}_{jk,t},\tilde{I}'}[ls_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|I] \\
&= \alpha E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t}|I] \\
&\quad +\alpha E_{\tilde{C}_{jk,t},\tilde{I}'}[e_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|I] \\
&\quad +\gamma E_{\tilde{C}_{jk,t},\tilde{I}'}[es_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|I] + \eta E_{\tilde{C}_{jk,t},\tilde{I}'}[ls_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|I] \\
&= \alpha E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t}|I] + E_{\tilde{C}_{jk,t},\tilde{I}'}[V_\mu(k, t+\tilde{C}_{jk,t}, \tilde{I}')|I] \\
&\quad \forall j \in N - \{d\}, \forall t, \forall I \\
V_{\mu^*}(d,t,I) &= \gamma max(0, t^* - \Delta - t) + \eta max(0, t - (t^* + \Delta)) \qquad \forall t, \forall I
\end{aligned} \tag{4.26}$$

From the above equation, we can easily verify that the principle of optimality holds for the minimization of linear combination of attributes of a routing policy, and thus enables us to design an algorithm to solve the problem. The optimality conditions are:

$$
\begin{aligned}
V_{\mu^*}(j,t,I) &= \arg\min_{k\in A(j)} \left\{\alpha E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t}|I] + E_{\tilde{C}_{jk,t},\tilde{I}'}[V_{\mu^*}(k,t+\tilde{C}_{jk,t},\tilde{I}')|I]\right\} \\
&\quad \forall j \in N-\{d\}, \forall t, \forall I \\
V_{\mu^*}(d,t,I) &= \gamma \max(0, t^*-\Delta-t) + \eta \max(0, t-(t^*+\Delta)) \qquad \forall t, \forall I
\end{aligned}
\tag{4.27}
$$

The main loop of the algorithm (termed Algorithm DOT-SLC) is very similar to that of Algorithm DOT-S or Algorithm DOT-SPI, making use of the property of positive travel times. Note that this algorithm is generic in that the current-information is not specified. One can follow the examples of Algorithm DOT-S and Algorithm DOT-SPI to adapt Algorithm DOT-SLC to either the NOI (no-online-information) variant or the POI (perfect-online-information) variant.

The initialization of Algorithm DOT-SLC is more involved than that of Algorithm DOT-S or Algorithm DOT-SPI, since the minimization criterion is a linear combination of three different attributes that might need different starting points. The most important procedure is to determine the time period beyond which the minimization problem becomes trivial. We realize that this time period is $max(K-1, t^*+\Delta)$, since beyond this period, the network is deterministic and static, and the (expected) late schedule delay is non-zero.

Algorithm DOT-SLC

Step 1: (Initialization)

1.1 (Initialization at the destination)

$e_{\mu^*}(d,t,I) = 0, es_{\mu^*}(d,t,I) = max(0, t^*-\Delta-t)$

$ls_{\mu^*}(d,t,I) = max(0, t-(t^*+\Delta))$

$V_{\mu^*}(d,t,I) = \gamma max(0, t^*-\Delta-t) + \eta max(0, t-(t^*+\Delta))$

$\mu^*(d,t,I) = d$

$\forall t, \forall I \in Z(d,t)$

1.2 $\hat{t} = max(K-1, t^* + \Delta)$

1.3 (Initialization for those beyond time $\hat{t}$)

$e_{\mu^*}(j,t,I)$ =minimum travel time from node j in deterministic static network $\{C_{jk,\hat{t}}\}$

$es_{\mu^*}(d,t,I) = 0, ls_{\mu^*}(d,t,I) = t + (e_{\mu^*}(j,t,I) - (t^* + \Delta),$

$V_{\mu^*}(j,t,I) = \alpha e_{\mu^*}(j,t,I) + \eta(t + (e_{\mu^*}(j,t,I) - (t^* + \Delta))$

$\mu^*(j,t,I)$ = next node of j in the shortest path tree of the deterministic static network $\{C_{jk,\hat{t}}\}$

$\forall j \in N - \{d\}, \forall t \geq \hat{t}, \forall I \in Z(j,t)$

1.4 (Initialization for those before time $\hat{t}$)

$V_{\mu^*}(j,t,I) = +\infty, \forall j \in N - \{d\}, \forall t < \hat{t}, \forall I \in Z(d,t)$

Step 2: (Main Loop)

For $t = \hat{t} - 1$ down to 0

For each I possible at node j and time t

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

$temp = \alpha E_{\tilde{C}_{jk,t}}[\tilde{C}_{jk,t}|I] + E_{\tilde{C}_{jk,t},\tilde{I}'}[V_{\mu^*}(k, t + \tilde{C}_{jk,t}, \tilde{I}')|I]$

If $temp < V_{\mu^*}(j,t,I)$

$V_{\mu^*}(j,t,I) = temp$

$\mu^*(j,t,I) = k$

Note that $e_{\mu^*}(j,t,I), es_{\mu^*}(j,t,I), ls_{\mu^*}(j,t,I)$ are respectively the expected travel time, expected early schedule delay and expected late schedule delay of the optimal routing policy μ^* that minimizes the linear combination of the three attributes. They are different from the minimum expected travel time, minimum expected early schedule delay or minimum expected late schedule delay, that are computed in an algorithm that minimizes the corresponding single criterion.

The complexity of Algorithm DOT-SLC depends on the specification of the current-information. We do not intend to study the complexity for some possible current-information specifications. Instead, we compare the complexity of Algorithm DOT-SLC with that of a corresponding algorithm that only minimizes expected travel time. This corresponding algorithm is Algorithm DOT-S (see Section 2.3) if the current-information is an empty set, and Algorithm DOT-SPI (see Section 2.4) if the current-information is the set of all link travel time realizations up to the current time.

In the initialization phase, Algorithm DOT-SLC does some more work on initializing the expected schedule delays and the linear combination of the three attributes, but the additional work is $O(1)$, and therefore does not affect the complexity of Algorithm DOT-SLC. In the main loop, the number of loops over discrete time periods is $\hat{t} = max(K - 1, t^* + \Delta)$, due to a different starting point in time. Therefore there is a ratio $\hat{t}/K$ between the complexity of Algorithm DOT-SLC and that of the corresponding algorithm minimizing expected travel time only.

4.4 Summary

Travelers are concerned about travel time reliability as well as expected travel time in a stochastic time-dependent traffic network. In this chapter, we study optimal routing policy problems that consider the reliability of travel time. Three measures are proposed for quantifying the reliability: variance, expected early schedule delay, and expected late schedule delay. We follow a similar approach for the study of the three individual problems. We first derive the recursive equation that relates the variance (expected early/late schedule delay) of a given routing policy at the current state and the variance (expected early/late schedule delay) at the next states. We then investigate whether the principle of optimality holds. If so, we can then write down the optimality conditions for the specific problem. In our study, we find that the principle of optimality does not hold for the minimum variance problem, but does hold for the minimum expected early/late schedule delay problem. Since the principle also holds for the minimum expected travel time problem, it holds for the problem of minimizing the linear combination of expected travel time and expected early/late schedule delays. For the minimum variance problem, we write down an approximate optimality condition, but its performance has not been tested computationally and is the topic of future research.

Algorithm DOT-SLC is presented for the problem of minimizing the linear combination, based on the optimality conditions. It is a label-setting type algorithm, making use of the assumption of strictly positive link travel times. We do not design algorithms to minimize individual reliability criterion, as expected travel time is the primary concern of travelers and deserves to be included in the objective function in any case.

Chapter 5

A Policy-Based Stochastic Dynamic Traffic Assignment Model

Dynamic Traffic Assignment (DTA) methods constitute parts in the intelligent core of Intelligent Transportation Systems (ITS). They provide support to the design, evaluation, operation of Advanced Traffic Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS). A DTA model captures the interaction between traffic demand and network supply in a time-dependent context and aims to estimate and/or predict network conditions, such as link travel times, O-D travel times, and link volumes, to support traffic management decision making and travelers' information provision.

Stochasticity in transportation systems is both intuitively prevalent and experimentally shown, as discussed in Section 1.1. In a given STD network, it is better to take adaptive routing decisions, and the optimal adaptive routing problems have been studied extensively in the previous chapters. In the following chapters, we will introduce the interaction between supply and demand in a stochastic time-dependent network, and aim to build an equilibrium traffic assignment model that can predict the traffic conditions of such a traffic network where users make adaptive routing decisions.

This chapter is organized as follows. A literature review on the research body of stochastic (dynamic) traffic assignment models is presented in Section 5.1. In

Section 5.2, we give an illustrative example to show how the policy-based DTA model can work and to explore some unique properties of the model. This is to provide the reader with intuitive understanding of and motivation for a policy-based model. In Section 5.3, a conceptual framework for the policy-based stochastic DTA model is introduced. Rigorous development of the model is given in Sections 5.4.3 through 5.6. In Section 5.4.3, the users' routing choice model is established. The dynamic network loading model is developed in Section 5.5. Finally we propose in Section 5.6 the solution heuristic for the policy-based DTA model.

5.1 Literature Review

Over the years, there have been various approaches to introducing stochasticity in equilibrium traffic assignment models. These approaches differ in many ways, and the four major features are listed as follows.

1. Is within-day time-dependency modeled? In other words, are link travel times explicitly dependent on the entry times? We will label a model with within-day time-dependency as dynamic, and a model without the feature as static. A static model is usually used to predict average traffic conditions over a day or peak hours, while a dynamic model makes the prediction in a finer scale in time.

2. What are the sources of randomness? There are three sources: perception errors, random supply, and random demand. Perception errors are due to users' imperfect knowledge about travel times. Random supply includes random disturbances to the traffic networks, including bad weather, incidents, work zones, vehicle breakdowns, and so forth. Random OD implies that the origin-destination (OD) trip rates are random variables.

3. Do users make adaptive routing choices? Note that a non-adaptive routing decision ignores any pre-trip and online information about the realized network conditions. Pre-trip online information is different from *a priori* information, in that the former is a kind of online information available at the origin node,

while the latter is the knowledge about the distribution of the link travel time random variables.

4. How is the travel time reliability modeled? As discussed in Chapter 4, in a stochastic network, travel time reliability is users' next most important concern besides mean travel time. Thus it is desirable to include it in the assignment model.

The literature review will discuss how the above four questions are addressed in each paper, to give the readers a structured view of the research on stochastic equilibrium assignment models. All the models reviewed are in the family of user equilibrium models. System optimum models are out of the scope of the thesis, as they cannot accommodate realistic users' behavior.

Early developments addressed stochasticity in **static** traffic assignment methods. Daganzo and Sheffi (1977) [21] established the concept of Stochastic User Equilibrium (SUE): in a SUE network no user *believes* he can improve his travel time by unilaterally changing routes. Users have random perception errors of the true travel times. The resulting path choices are therefore naturally random, in the form of path choice probability. Equilibrium path flows, however, are not presented as distributions. Instead, a "large sample" approximation is used, such that the proportion of travelers that take a given path equals its probability to be chosen by an individual traveler. Consequently, an "average" deterministic flow pattern is obtained. In this paper, the stochastic-network-loading (SNL) problem (a special case of SUE for networks with constant link costs) is analyzed. In a later paper by Sheffi and Powell (1982) [46], the SUE problem with flow-dependent link costs is studied, with a proof of convergence. A Probit model is used for users' path choice, and simulation is used to compute the link flows with given true link travel times.

The work on Stochastic User Equilibrium (SUE) by Daganzo and Sheffi (1977) [21] is extended later in two different directions of considering stochasticity. The first direction is to abandon the "large sample" assumption of flows in Daganzo and Sheffi (1977) [21] and Sheffi and Powell (1982) [46], and treat the flows (either path flow or link flow) as random variables. The other direction is to add one more source of stochasticity by treating the underlying traffic network as random. We will review the work in these two directions respectively.

Along the first extension direction, the "large sample" assumption is abandoned, and the route choice of a user with random perception error is treated as a random variable. Under this situation, for a given OD flow, the distribution of path flows is multinomial, and the distribution of an individual path flow is binomial. Cascetta (1989) [16] and Cantarella and Cascetta (1995) [15] adopted a stochastic process (SP) approach to the problem. A random utility model is used for route choice as that in SUE, but the expected cost of a route is explicitly determined from the experience of the last several time epochs (e.g. days). The random component that drives the stochastic process is the route choice behavior: users' actual route choices are sampled from relevant distributions. In particular, the conditions for the existence and uniqueness of a stationary stochastic process are provided and their relationship to an equilibrium state is studied. The relationship between an SP solution and SUE solution is discussed in Cascetta (1989) [16], while some sensitivity analysis of the SP model is carried out in Cantarella and Cascetta (1995) [15]. This stochastic process framework conceivably can take multiple sources of randomness: demand, supply, and perception errors, although only perception errors are explicitly studied in these two papers. It is also stated that the analysis and result can be extended to cover the within-day dynamic models, but are not explicitly dealt with. Later on, Davis and Nihan (1993) [22] generalize the SP approach in Cascetta (1989) [16] by using a general form for the adjustment mechanism to describe how travelers adjust their route selection to existing system conditions.

Another approach along the first extension direction deals directly with equilibrium in path (link) flow distribution without modeling a stochastic process. Hazelton (1998) [30] addresses stochasticity emanating from perception errors. A different assumption on user behavior than that in SUE is proposed, termed "conditional stochastic user behavior", stating that a user selects the route s/he perceives to have minimum cost *conditional* on all other travelers' choices. The conditional stochastic user equilibrium (CSUE) is then defined based on the behavior assumption. A numerical example is used to show that CSUE gives different results from the equilibrium as a "large sample" approximation developed in Daganzo and Sheffi (1977) [21]. The CSUE and SUE tend towards each other as the demand increases, as is expected. In an earlier paper by Hazelton *et al* (1996) [31], a numerical method for simulating CSUE flow patterns is developed using so called Markov Chain Monte Carlo methods.

Watling (2002) [47] [48] also develops an equilibrium model based on network flow distribution, but taking a different approach than that of Hazelton (1998) [30]. Users are assumed to base their choices on a finite collection of the actual link costs experienced in their trips on days in the past. The random link cost variables in the past days are assumed to be independent and identically distributed (i.i.d.). The equilibrium is defined as a fixed point condition on the joint distribution of network flows. Then, an approximation to this condition is derived, equilibrating moments of order n and below of the joint flow probability distribution. The papers go on to focus on the second order model, GSUE(2). Conditions are presented to guarantee the existence of the second order model solutions. A heuristic solution method of GSUE(2) is presented and numerical experiments carried out to compare results with those of SUE and the stochastic process model of Cantarella and Cascetta (1995) [15].

The second direction of extending the classical SUE model is to explicitly incorporate more sources of randomness. Mirchandani and Soroush (1987) [37] considered the case where stochasticity comes from both traveler perception errors and link travel costs themselves. Each user perceives a travel time probability distribution for each route, which may vary from user to user. Each traveler uses a disutility function of travel time to evaluate each route and chooses the route with minimum expected disutility. The expectation is taken over link travel costs, so the expected disutility is still random distributed over population, and thus we can talk about choice probability. The "large sample" approximation of traffic flow proportions is also used in their work. Various disutility functions are utilized to represent various risk taking behaviors: risk averse, risk neutral or risk prone. The solution of a general model is very difficult, but solvable special cases are identified: statistically independent link travel costs and linear, exponential, or quadratic disutility functions. Later on, Liu *et al* (2002) [32] extends the static model to dynamic model.

There are other papers that do not follow the convention of SUE, but also consider stochasticity of traffic networks in one way or another. Lo and Tung (2003) [33] proposed a notion of probabilistic user equilibrium (PUE) in a static network with links that have random capacities. The travel costs, which depend on capacities, are then also random and users are assumed to take minimum expected travel cost routes whose variances are below a given threshold. Arnott *et al.* (1991) [7] and Emmerink *et al.* (1998) [24] adopted a different approach than the conventional one in the transportation research literature. Their studies are conducted on a simple network with one OD pair and two parallel links with stochastic capacities. Travel times on the links are determined by the bottleneck model. In addition, the demand is elastic in Emmerink *et al.* (1998)[24]. Users are provided with online information about the realized link capacities. User equilibrium over departure times and over routes are sought. The impact of information on congestion or consumer welfare is analyzed. It is however conceptually difficult to extend the analysis to a general network.

In the transit assignment research literature, there has been quite a lot research on the hyperpath based assignment (see the references in Marcotte and Nguyen (1998) [34]). The notion of a hyperpath is similar to a routing policy, in the sense that they both represent adaptive (or "strategic") routing choices. A hyperpath is a general concept in graph theory, and was introduced to the transportation research community by Nguyen and Pallottino (1988) [38]. In the transportation research literature, it is usually applied in transit networks and thus its definition is limited in some sense. We introduce a popular definition of hyperpath as adopted in Marcotte and Nguyen (1998) [34]. A strategy is defined as a mapping that associates with every node of a network an ordered set of successor nodes. In a transit network, this set of successor nodes is usually termed as "attractive lines", and users are assumed to take the first available link among the attractive lines. This simplifying assumption allows for mathematically tractable assignment models. The actual choice of the next link is random, as the arrivals of the transit vehicles are assumed random. Thus each link can be associated with an access probability for a given strategy, if the random characteristics of the whole system are well defined. The combination of a strategy and its arc probabilities defines a hyperpath. The relationship between a strategy and the corresponding hyperpath is similar to that between a routing policy and the corresponding state network as discussed in this thesis (see the example in "The Decision Process" of Section 2.2.1 for details about a state network), which is the relationship between a decision rule and the collection of possible outcomes of applying the decision rule in a random network. Although a strategy (as defined above) and a routing policy (as defined in this thesis) are both decision rules, the rules are specified in different ways. A routing policy is more general than a strategy as defined above, because we can express the rule "taking the first available link" as mappings between conditions of attractive lines and the next nodes to take. However, a strategy is more suitable and efficient in a transit network. Because of the definition difference, algorithms of finding a shortest hyperpath and an optimal routing policy are also quite different at first sight. The output from a shortest hyperpath algorithm is the optimal set of attractive lines associated with each node, while that from an optimal routing policy algorithm is a next node associated with each possible state (the node is part of the state). However, they are not fundamentally different, as we can translate a shortest hyperpath into a routing policy.

Most of the hyperpath based equilibrium assignment models are for static transit networks, e.g. a hyperpath formulation of equilibrium assignment model in capacitated network with constant link costs is presented in Marcotte and Nguyen (1998) [34]. Cantarella (1997) [14] presents a general fixed-point approach to static traffic assignment in a general network where users' choice sets include hyperpaths. However, in his definition, link access probabilities are not functions of link flows, which might not be a realistic assumption. Furthermore, link flows and costs are not treated as random variables, although one would imagine that users taking hyperpaths will naturally result in probabilistic flows. Instead, some approximations are taken in relating expected link costs with expected link flows. An interesting extension direction of the existing research on hyperpath formulation of equilibrium assignment in some specialized static networks would be a hyperpath formulation in dynamic general networks, but to the best of the author's knowledge, there has not been papers in this direction.

After the literature review, we conclude that there is no published research on an adaptive routing policy based dynamic traffic assignment model, where users are assumed to make adaptive routing decisions, and the underlying traffic network is both random and explicitly time-dependent. Therefore, this thesis contributes to the existing research on traffic equilibrium by building the first adaptive routing policy based DTA model.

5.2 An Illustrative Example of the Policy-Based DTA

Since the notion of equilibrium traffic assignment based on routing policies is not conventional, we present an illustrative example to explain some of the key concepts, give the motivation for the research and provide some insight into the problem. The example has very simple settings to make it easy to understand and solve. The network is static to make the concept easier to understand. Examples with dynamic networks could also be constructed, but are not presented in this thesis. Note that some of the assumptions are special cases of those in the more general policy-based DTA model, and we will point them out when needed.

For the ease of presentation, we use the following notation:

a	:	index for link number
i	:	index for policy (including path) number
x_a	:	flow on link a
f_i	:	flow on policy (path) i
$C_a(x_a)$	:	link travel time as a function of link flow for link a
e_i	:	expected travel time of policy i

We make use of the well-known Braess network with some changes to make it

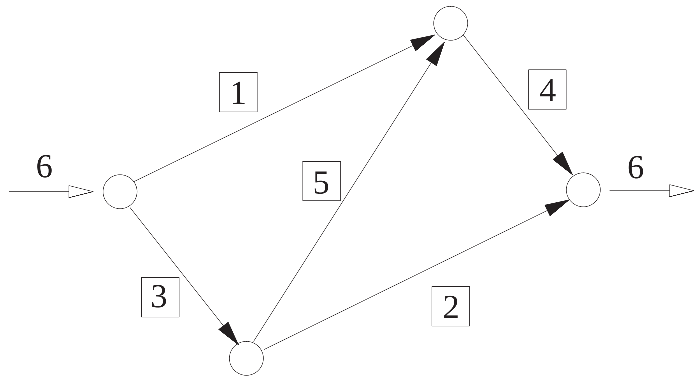


Figure 5-1: An Illustrative Example: the Network

stochastic. The network is shown in Figure 5-1 with 5 links and one OD pair. Three paths are available between the OD pair: path 1 composed of links 1 and 4; path 2 composed of links 3 and 2; and path 3 composed of links 3, 5 and 4. We can view path 1 and path 2 as two arterials between the origin point and the destination point, while link 5 is a shortcut to switch from path 2 to path 1. The travel time on a given link is a function of the link flow.

$$C_1(x_1) = 50 + x_1, C_2(x_2) = 50 + x_2$$

$$C_3(x_3) = 2x_3, C_4(x_4) = 2x_4$$

$$C_5(x_5) = \begin{cases} 10 + x_5, & w.p.\ 0.95 \\ 10 + 10000x_5, & w.p.\ 0.05 \end{cases}$$

Links 1 through 4 have deterministic travel time functions, while link 5 has a random travel time function. Specifically, the travel time on link 5 could take two different forms as a function of link flows, each with a given probability. The randomness could come from multiple sources, e.g. an incident or bad weather effect on a narrow bridge. To make our presentation clearer, one will be called the incident case ($C_5(x_5) = 10 + 10000x_5$) and the other the normal case ($C_5(x_5) = 10 + x_5$), where the incident case is represented by a 10000 times smaller capacity.

5.2.1 Path-Based Approaches

We now study how users will make routing decisions in the stochastic network. Two approaches are available to assign users in a stochastic network: one is based on paths, and the other on routing policies. The user equilibrium is sought. To make the illustration easier, users are assumed to have no perception errors and try to minimize their (expected) travel times. Note that the underlying network is stochastic, and thus the OD travel times are also stochastic, although there are no random perception errors. Following the classical definition of user equilibrium that all used paths have the same and minimum expected travel time, we have the following solutions: $f_1 = f_2 = 2.963180228, f_3 = 0.073639527$, with the expected path travel times: $e_1 = e_2 = e_3 = 59.037$. Because of the symmetry of the network, the equilibrium flows are also symmetric. We can see that only a very small fraction of the travelers take path 3, as the travel time of link 5 in incident is prohibitively high even with very small link flow.

5.2.2 Motivation for Adaptive Routing

As shown in the above discussion, the expected OD travel time in a path-based user equilibrium is around 59. Can travelers further reduce their expected travel times? The answer is yes, provided that information is available on the incident occurrence. Assume a variable message sign (VMS) is put at the end of link 3 and it indicates whether there is an incident on link 5, i.e. which travel time function link 5 will assume. Only those users passing link 3 can see the VMS and thus obtain the information.

Assume users are now in the user equilibrium with path-only choices, i.e. $f_1 = f_2 = 2.963, f_3 = 0.074$. One user then decides to make use of the VMS by taking an adaptive routing choice like this: first take link 2, and if the VMS indicates an incident, take link 2, otherwise take link 5 and continue on link 4. We assume the effect on link travel times of this user's routing change is negligible, which is reasonable when the demand is very large and only one marginal user uses the VMS information (note that the unit of the demand can be thousands of users). Therefore the expected travel time of this user is $C_3(f_2+f_3)+C_2(f_2))\times 0.05+(C_5(f_3)|normal+C_4(f_1+f_3))\times 0.95 = 2\times (2.963+0.074)+(50+2.963)\times 0.05+((10+10000\times 0.074)+(50+2.963+0.074))\times 0.95 = 24.06$. This number is lower than 59 and is a great incentive for the user to adopt the adaptive routing choice. This phenomenon is common in reality, where experienced users make their routing choices *en route* based on realized network conditions, either through their own observation, other users' messages, or information provided by traffic agencies using VMSs or radio. This type of user behavior cannot be adequately modeled in conventional traffic assignment models due to the limitations that users' choices are confined to non-adaptive paths and the underlying networks are usually assumed to be deterministic. It is therefore desirable to build a DTA model where users' routing choices are adaptive and the underlying network is stochastic.

The adaptive choice is also relevant in a route guidance context. Assume a route guidance system is available to users to reduce their travel times. The guidance should be anticipatory and consistent. A formal definition of anticipatory consistent route guidance in deterministic networks can be found in Bottom (2000) [13]. Loosely speaking, anticipatory route guidance is based on prediction of future network conditions, in contrast to that based on prevailing or historical network conditions. Route guidance is consistent if the network conditions based on which the guidance is made are going to be experienced by the users. Consistency is a model property, which simply says that the input and output of a model should not be contradictory. Generally an information sensitive traffic assignment algorithm is used to solve for the predictive network conditions needed in generating anticipatory consistent route guidance.

We impose another criterion on the route guidance: credibility. A necessary condition for a user to follow the route guidance is that by following the guidance, s/he has no larger expected travel times than other users of the traffic network, including both other users who follow the guidance and those who do not follow the guidance. Conceivably, credible guidance must at first be anticipatory and consistent. In our example, suppose the link travel time functions are correct predictions of future link performances and all users will follow the guidance, therefore the guidance generated based on the path-based stochastic user equilibrium assignment is anticipatory and consistent by construction. We now check whether the guidance is credible. It is true that all users who follow the guidance have the same travel time of 59, and the guidance is credible in this sense. However, assume now we have another user who takes an adaptive routing choice as specified above. The user will then experience a lower expected travel time, voiding the credibility of the route guidance. A possible solution to the credibility problem is to generate adaptive guidance. The guidance will not tell the users always to take a fixed path, but rather will provide a decision rule that gives the actions to take based on realized network conditions.

To sum up, in a descriptive sense, it is more realistic to model users' choices as adaptive rather than fixed in equilibrium analysis. Furthermore, in a prescriptive sense, it is desirable to incorporate adaptive routing choices in generating route guidance to solve the credibility problem.

5.2.3 Policy-Based Assignment

As we know, a routing policy in a stochastic time-dependent network (STD) is a decision rule which specifies what node to take next out of the current node based on the current time and information on realized network conditions. In our example, the realized network condition is whether or not the incident has occurred. Generally the information access to the network conditions depends on the actual applications. It could be qualitative or quantitative, local or global, up-to-date or lagged.

Note that a path is also a routing policy, so in this example, we have five routing policies. Policies 1 through 3 are just paths 1 through 3, while policy 4 is the one discussed before: first take link 3, if the incident has occurred, take link 2, otherwise take link 5 and then link 4. Policy 5 is the opposite of policy 4: first take link 3, if the incident has occurred, take link 5 and then link 4, otherwise take link 2. Policy 5 is conceivably not optimal, but is included as the assignment needs all possible routing policies.

We need to define a policy-based user equilibrium for this example. We assume travelers have no perception errors for the sake of simplicity. Recall that in a conventional deterministic path-based model, a user equilibrium is reached when all used paths have the same and minimum travel time. In other words, a traveler cannot improve his/her travel time by unilaterally changing to another path. This condition is based on the behavioral assumption that travelers minimize travel time.

We keep this behavioral assumption, but in addition assume that travelers follow routing policies rather than fixed paths when faced with random traffic conditions. A natural extension from the path-based approach to a policy-based equilibrium is that all used policies have the same and minimum expected travel times. Associated with such an equilibrium is a link time distribution, and if an individual finds an optimal routing policy with this distribution, he/she will find out its expected travel time is the same as experienced by all travelers (including himself/herself). In other words, a traveler cannot improve his/her expected travel time by unilaterally changing to another routing policy.

In Figure 5-2 we write down a system of equations that correspond to the equilibrium condition. The assignment result is $f_3 = 0.0046, f_4 = 5.9954, f_1 = f_2 = f_5 = 0$. We can verify that under these policy flows, the network is in equilibrium, such that policies 3 and 4 have the same and minimum expected travel times. Indeed, under these assignment flows, we can calculate expected travel times of all routing policies as shown in Figure 5-2.

The expected OD travel time for all users is 41.4 and is less than the expected OD travel time when users only choose among paths (59). This is in accord with a general perception that information can improve travel times. However one should not mistake this result as an indicator that more information is always better. In fact, if we change the link travel time functions of link 3 and link 4 to $C_3(x_3) = 10x_3, C_4(x_4) = 10x_4$, the policy-based assignment gives an expected OD travel time of 91.1 which is higher than that in a path-based assignment (83.1). The intuition behind the result is similar to that for the traditional Braess' paradox: users are trying to minimize their own (expected) travel times without coordination. More information could provide "selfish" users more chances to squeeze benefit from the system and in the end result in a worse overall situation. On the other hand, the

$$
\begin{aligned}
\text{With incident on link 5: } x_1 &= f_1 \\
x_2 &= f_2 + f_4 \\
x_3 &= f_2 + f_3 + f_4 + f_5 \\
x_4 &= f_1 + f_3 + f_5 \\
x_5 &= f_3 + f_5 \\
\text{No incident on link 5: } x_1 &= f_1 \\
x_2 &= f_2 + f_5 \\
x_3 &= f_2 + f_3 + f_4 + f_5 \\
x_4 &= f_1 + f_3 + f_4 \\
x_5 &= f_3 + f_4 \\
\text{Therefore} & \\
e_1 &= 50 + x_1 + 2 \times x_4 | normal \times 0.95 \\
&\quad +2 \times x_4 | incident \times 0.05 \\
&= 61.4, \\
e_2 &= 2 \times x_3 + (50 + x_2 | normal) \times 0.95 \\
&\quad +(50 + x_2 | incident) \times 0.05 \\
&= 62.3, \\
e_3 &= 2 \times x_3 + (10 + 10000 \times x_5 | incident \\
&\quad +2 \times x_4 | incident) \times 0.05 + \\
&\quad (10 + x_5 | normal + 2 \times x_4 | normal) \times 0.95 \\
&= 41.4, \\
e_4 &= 2 \times x_3 + (50 + x_2 | incident) \times 0.05 \\
&\quad +(10 + x_5 | normal + 2 \times x_4 | normal) \times 0.95 \\
&= 41.4, \\
e_5 &= 2 \times x_3 + (10 + 10000 \times x_5 | incident \\
&\quad +2 \times x_4 | incident) \times 0.05 \\
&\quad +(50 + x_2 | normal) \times 0.95 \\
&= 62.3.
\end{aligned}
$$

Figure 5-2: Expected Travel Times of Routing Policies policy-based system optimal assignment gives expected OD travel times at least as good as those from the path-based system optimal assignment. We do not elaborate on this point, as our study in this thesis is on user equilibrium.

We conclude that the value of information could be positive or negative, and the magnitude depends on the network characteristics, such as demand scale, incident probability, information penetration, network structure, etc. This raises a set of research questions in route guidance: what information should be provided? Who should be provided with information? When should the information be provided? These questions have been explored in the literature through various approaches. With the routing policy based DTA framework, these problems can be studied systematically.

5.3 A Conceptual Framework for the Policy-Based Stochastic DTA Model

After analyzing the simple illustrative example in the previous section, we proceed to present a conceptual framework for the policy-based stochastic dynamic traffic assignment model to give a big picture on the input, output, model components' interaction, and data flow, as shown in Figure 5-3. The input to the overall DTA model is the stochastic dynamic demand $\tilde{D}$ and supply $\tilde{S}$ represented by a joint discrete distribution with R support points, each of which has a probability $p_r, r = 1, ..., R$. The demand is assumed to be inelastic, i.e. the demand distribution is fixed. In a discrete time representation, any realization of random demand is given as a matrix of time-dependent numbers of O-D trips during all time intervals. $\tilde{D} = \{D^1, D^2, ..., D^R\}$, where D^r is the demand matrix for the r^{th} support point. $D^r = \{D^r_{jd,t}, t = 0, 1, 2, ..., \forall \text{OD pair}\{j, d\}\}$, where $D^r_{jd,t}$ is the number of trips between origin j and destination d for departure time t for the r^{th} support point. The random supply can be represented through the random occurrence, duration and severity of an incident or any other random supply factors: $\tilde{S} = \{S^1, S^2, ..., S^R\}$. Note that the same probability p_r is associated with the outputs computed from S^r, D^r. In the remaining of the thesis, whenever a support point has a superscript r, its associated probability is p_r, otherwise indicated. The output is an equilibrium distribution of link travel times $\tilde{C} = \{C^r_{jk,t}, \forall\{j,k\} \in A, \forall t, r = 1, 2, ..., R\}$, where A is the set of links of the traffic network, and the corresponding routing policy splits $f = \{f^i_{jd,t}\}$, where $\{j, d\}$ is an OD pair, t is the departure time, and i is the index of policies. Other measures of effectiveness of interest, such as the distribution and summary statistics of link volumes, O-D travel times, and path flows can be obtained from the equilibrium distribution of link travel times, if needed. Note that the distributions of all relevant traffic random variables are discrete, as our definition of a routing policy is based on a discrete distribution of link travel times.

There are three major components of the stochastic DTA model:

- the users' routing policy choice model, denoted as U,
- the policy-based dynamic network loading model, denoted as L,
- and the optimal routing policy algorithm, denoted as O.

The following paragraphs will discuss them in turn.

5.3.1 Users' Routing Policy Choice Model

The users' routing policy choice model takes as input a set of routing policies $G = \{\mu_1, \mu_2, ..., \mu_i, ...\}$ generated by the optimal routing policy algorithm, and a joint distribution of link travel times $\tilde{C} = \{C^r_{jk,t}, r = 1, ..., R\}$ generated by the policy-based dynamic network loading model. The method of generating the choice set will be discussed in detail in Section 5.4.3 and Section 5.6. Based on the relevant attributes of candidate routing policies, such as expected OD travel time and travel time standard deviation, the users' policy choice model outputs policy splits f among the routing policies for each OD pair and each departure time.

$$f = U(G, \tilde{C})$$

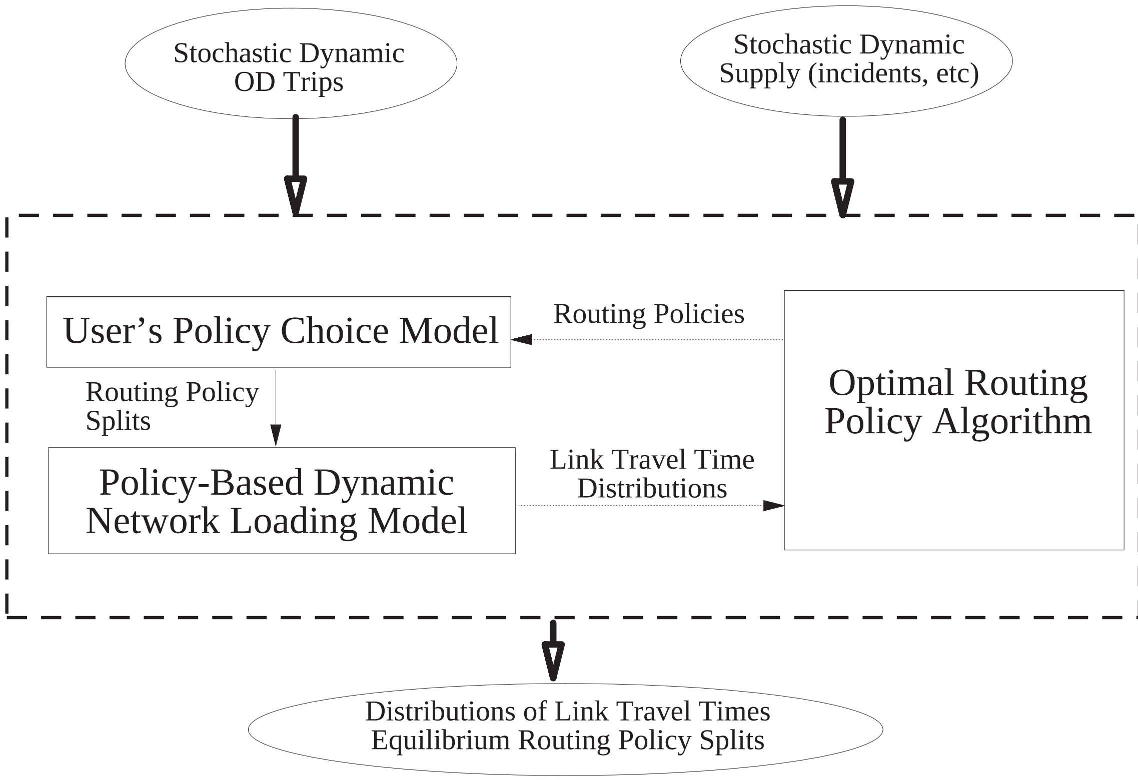


Figure 5-3: A Conceptual Framework of Stochastic Dynamic Traffic Assignment Model

We keep the "large sample" assumption and assume policy splits are equal to corresponding policy choice probabilities. Note that we use "splits" rather than "flows" here: policy splits are deterministic, while policy flows could be stochastic, if the demand is stochastic. Policy splits will be translated into policy flows in the network loading model. The notion of policy flow can be understood as a generalization of path flow. Since a routing policy will manifest itself as a specific path for a given realization of link travel times, a policy flow will become a path flow for each support point of link travel times. Thus a policy flow can be viewed as a set of path flows, each with some probability.

5.3.2 Policy-Based Dynamic Network Loading Model

The demand is then loaded onto the network according the policy flow splits, by the policy-based dynamic network loading model. The stochastic demand and supply play their roles in the loading process. For each support point of the random demand and/or supply, the network loading model outputs a single realization of the link travel time distribution. Therefore through the loading, we obtain the PMF of link travel times from the PMF of demand/supply. Note that although the input demand/supply support points are distinct from each other, the output link travel time realizations are not necessarily distinct. This is why the word "realization" is used here, rather than support point. Nevertheless, the PMF of link travel times is still expressed through the R realizations with the corresponding probabilities.

$$\tilde{C} = L(f, \tilde{D}, \tilde{S})$$

5.3.3 Optimal Routing Policy Algorithm

The routing policy generation algorithm then takes as input the link travel time distribution and produces an optimal routing policy for each destination, which again will be used to generate the choice set for the users' policy choice model.

$$\mu_i = O(\tilde{C})$$

$$G = G \cup \mu_i$$

The two equations can be combined as

$$G = G(\tilde{C}).$$

5.3.4 Policy-Based Equilibrium

The three components interact with each other, and a fixed-point formulation of the policy-based equilibrium can be derived based on the interaction.

$$\tilde{C} = L\left(U\left(G\left(\tilde{C},\right), \tilde{C}\right), \tilde{D}, \tilde{S}\right) \tag{5.1}$$

This thesis does not examine the mathematical properties of this fixed point problem (existence, uniqueness of solutions), but rather uses it as a guide in developing heuristics for the policy-based stochastic DTA problem.

Detailed presentations of the users' policy choice model, policy-based dynamic network loading model, and the overall policy-based DTA model will follow. The optimal routing policy algorithms have been studied extensively in the previous chapters of this thesis.

5.4 Routing Policy Choice Model

We first study the routing policy choice model. It takes as input a set of routing policies $G = \{\mu_1, \mu_2, ..., \mu_i, ...\}$ generated by the optimal routing policy algorithm, and a joint distribution of link travel times $\tilde{C} = \{C^r_{jk,t}, r = 1, ..., R\}$ generated by the policy-based dynamic network loading model, and outputs policy splits among the routing policies for each OD pair and each departure time.

It is an interesting research question to model travelers' routing choices under uncertainty. The expected travel time has been the primary (and in many cases, the only) factor. On the other hand, the reliability of the random travel time is hypothesized to have effects on the routing choice. In regard to this, the literature can be classified into three categories: 1) Random utility choice models, where the effect of travel time reliability is indirectly represented by schedule delay (see for example Abkowitz (1980) [3], Ben-Akiva *et al.* (1986) [9], de Palma *et al.* (1990) [23], Arnott *et al.* (1990) [6], and Noland *et al.* (1995) [39]) ; 2) Disutility models, where each value of the travel times corresponds to a disutility value and the user is assumed to minimize the expected disutility (see for example Mirchandani *et al.* (1987) [37] and Liu *et al* (2002) [32]); 3) Random utility choice models, where the variance of travel times (or equivalently, the standard deviation) is one attribute of the systematic utility (see for example Abdel-Aty *et al.* (1995) [2] [1]). Note in 1) and 3), expected travel times are by default included in the utility functions.

5.4.1 Policy-Size Logit Model

We generalize the classical random utility model for the policy-based users' behavior model. In short, a random utility choice model assumes that users are rational decision makers, and will choose an alternative with the highest utility. The utility function is composed of two parts: a systematic utility and a random error. The systematic utility is usually a linear combination of attributes of the alternative (for route choice, it can be travel time, travel cost, etc) and possibly the characteristics of the traveler. The random error is present because of the traveler's perception error and the limited ability of the modeler to include all factors in the systematic utility. Since the utility is random, we can talk about the choice probability of each route (or policy), which is the probability that this route (or policy) has the largest utility. Different route choice models exist due to different specifications of the systematic utility function and different assumptions about the probabilistic distribution of the random error. We develop a "policy-size Logit" model here.

A traditional Logit model assumes the error term is independently and identically distributed as Gumbel, and thus enables a closed form expression of choice probabilities, which is the outstanding appeal of the Logit model. However, the assumption of independence is usually violated in the case of route (policy) choice due to the overlapping of routes (policies). Intuitively, the perception errors of two paths (policies) with overlapping links should not be independent, as the perception errors of the overlapping links should be closely related, if not the same. There has been extensive study on the violation of the Logit assumption, termed the IIA property (independent from irrelevant alternatives). We adopt an approach proposed by Ben-Akiva and Ramming (1998) [11], the so-called path-size Logit, and generalize it to policy-size Logit.

The policy-size Logit model adds a correction term to the utility of individual policies in a policy choice set. Specifically,

$$P(i|G) = \frac{e^{\ln PoS_i + V_i}}{\sum_{l \in G} e^{\ln PoS_k + V_l}} = \frac{PoS_i \times e^{V_i}}{\sum_{l \in G} PoS_l \times e^{V_l}} \tag{5.2}$$

where V_i is the utility of policy i, PoS_i is policy-size of policy i, and G is the choice set of policies.

When a policy has no overlap with any other policy, it has a policy-size of 1. At the other extreme, when two policies are exactly the same, each of them should have a policy-size of 0.5. Generally, when J policies completely overlap, each has a policy-size of $1/J$. A policy-size is calculated based on link size, as shown in the following equation:

$$PoS_i = \sum_{r=1}^{R} \left(\sum_{a \in B_i^r} \left(\frac{C_a^r}{PC_i^r} \right) \frac{1}{N_a^r} \right) p_r \tag{5.3}$$

with the notation:

PoS_i	:	policy-size of policy i
R	:	number of support points of link travel time distribution
B_i^r	:	set of links of the realized path of policy i for support point r
C_a^r	:	travel time of link a for support point r
PC_a^r	:	realized travel time of policy i for support point r
N_a^r	:	count of paths using link a for support point r

Please note that all the variables are time-dependent, but the time subscript is omitted from the equation to make the notation light. The policy-size can be viewed as expected path-size. Basically, for any given support point of link travel times, a policy manifests itself as a path. We then calculate the path-size for each realization and take the expectation over all possible realizations to obtain the policy-size. In each realization, the contribution of a link is proportional to its relative length ($\frac{C_a^r}{PC_i^r}$) and inversely proportional to number of paths that use this link ($\frac{1}{N_a^r}$).

Next we specify the systematic utility function of a policy i. Theoretically, the proposed framework for the policy-based stochastic DTA model does not restrict the specification. A reasonable specification is to include the primary attribute, the expected travel time, and possibly some other travel time reliability related attributes, such as travel time standard deviation and expected early (late) schedule delay, as shown in Equation 5.4.

$$V_i(j,t,d) = \alpha e_i(j,t,d) + \beta\sqrt{v_i(j,t,d)} + \gamma es_i(j,t,d) + \eta ls_i(j,t,d) \tag{5.4}$$

where

j : origin node

t : departure time

d : destination node

e_i : expected travel time of policy i

v_i : travel time variance of policy i

es_i : expected early schedule delay of policy i

ls_i : expected late schedule delay of policy i

There are other important factors that affect the users' policy choice when they have inflexible arrival times. One example is the probability of being late, as a user might have a very strict requirement for arrival time and being late regardless of the extent of lateness will cause serious consequences. One such situation would be catching a plane. This situation, however, can be approximated by setting the coefficient of expected late schedule delay to be very large.

5.4.2 Choice Set Generation

The choice set should contain routing policies that will be considered by users, i.e. routing policies with fair chances to yield maximum utility in some circumstances for users. A naive approach is to enumerate all policies between each OD pair and for each departure time. Nevertheless, this is an extremely difficult task, as the number of policies is exponential in the network parameters. In fact, enumeration of paths has been a hard task even for traditional path-based route choice models.

There has been research on generating choice sets for path-based routing choice models, and the proposed approaches include but are not restricted to the link penalty method, the link elimination method, the k-shortest path method, and the simulation method. In the link penalty method, at first a shortest path between a given OD pair is identified and is assumed to be in the users' choice set. Next the link impedance of some links is increased and a new shortest path is calculated. This new shortest path is assumed to be a feasible alternative to consider and is then included in the choice set. This operation is repeated for a certain number of links, until a reasonable sized choice set is generated. If the link impedance of some links is increased to infinity, then the link penalty method becomes the link elimination method. The k-shortest path method includes the first k shortest paths between an OD pair in the choice set. The simulation method samples link travel time values from some assumed distribution (usually normal), and then finds the shortest path which will be included in the choice set. A certain number of samples are drawn to obtain a choice set.

We do not aim to explore extensively the policy-based choice set generation methods in this thesis. We will elaborate this approach in Section 5.6, along with the DTA solution algorithm, as the choice set generation method is closely related to the DTA solution method. Briefly speaking, the DTA solution method is an iterative process, and during each iteration, an optimal routing policy is generated and added to the choice set. We show in Chapter 6 that it works well for our numerical tests.

5.4.3 User's Routing Policy Choice Algorithm

The users' policy choice algorithm takes as input the candidate routing policies for an OD pair and departure time, and a given joint distribution of link travel times based on which the attributes of policies (expected travel time, standard deviation, etc) are calculated. It produces policy splits $f^i_{jd,t}, \forall (j,d), \forall \mu_i \in G_{jd,t}, \forall t$, where (j,d) is an OD pair, $G_{jd,t}$ is the choice set of routing policies between origin j and destination d and for departure time t. Let $P^i_{jd,t}$ be the probability that a user with OD (j,d) will choose policy i at time t, which can be computed from the policy-size Logit model specified in Equations 5.2 through 5.4.

We apply the large sample approximation to the policy choices, such that the proportion of travelers taking a given policy is the same as the probability that an individual will take that policy. The algorithm of the policy choice model is then as follows:

$$\forall \text{OD pair } (j,d), \forall \mu_i \in G_{jd}, \forall t, f^i_{jd,t} = P^i_{jd,t}$$

.

We can classify the users according to various characteristics, as is typical in usual path-based DTA models. However, one important characteristic enabled by the use of routing policies is that we can naturally classify users according to their information access, as the information access is embedded in the definition of a routing policy. Please refer to Section 2.2.2 for a detailed discussion about information access.

5.5 The Dynamic Network Loading Model

The policy-based dynamic network loading model takes as input the routing policies, policy splits, the PMF of random demand/supply and the network description, and outputs a PMF of link travel times. Two features distinguish the policy-based stochastic dynamic network loader from a classical path-based deterministic dynamic network loader.

First, the demand and supply can both be stochastic, and are described by a joint discrete distribution $\{D^1, S^1, D^2, S^2, ..., D^R, S^R\}$ and associated $\{p_1, p_2, ..., p_R\}$. Stochastic factors in the supply side can generally be modeled as changes in capacities. For example, an incident may block several lanes, and thus reduce the capacity of the link. The starting time, the location, the duration and the severity of an incident can all be random. Another example would be a traffic light. A red light can be viewed as a device to reduce the outflow capacity to zero, while a green light restores the outflow capacity to the normal value. Therefore the stochastic time-dependent supply can be modeled using time-dependent random capacities. The modeling of stochastic demand is straightforward by treating the number of OD trips at each time period as a random variable. Examples of discrete distributions of random supply/demand can be found in Chapter 6 where computational tests are carried out. This feature actually does not necessitate fundamental changes to a path-based deterministic loader. What is needed is repeated runs of a loader for R times, for each support point of the random demand/supply, to generate the PMF of link travel times. When R is large, exhaustive enumeration of all possible S^r, D^r combinations may not be computationally feasible, thus a sample from the full distribution may be generated instead.

Secondly, users follow routing policies, rather than non-adaptive paths. A routing policy is a decision rule based on on-line information. To determine the next link of policy μ for all links at time t, we must translate the currently available link travel times to a current information at time t as defined by policy μ. Then we move the users according to the computed next links, and repeat the above operation when they reach the end of the next links. Conceivably, the actual path taken by a user is not known until the end of the trip. This is different from a path-based (non-adaptive) network loader, where the sequence of links a traveler will take is known before the trip starts. This feature makes necessary a fundamental revision to the design of a path-based loader, by either of two approaches: chronological or iterative. We elaborate the two approaches as follows.

We consider the policy-based "single-loading" problem, where the demand and supply are set at a given realization, say D^r and S^r. The model we seek to build is an adaptive loader, denoted as AL, such that $C^r = AL(f, D^r, S^r)$, where f represents the policy splits. Assume we also have available a non-adaptive (path-based) dynamic network loader, denoted as NAL, which takes as input *path* splits pf, such that $C^r = NAL(pf, D^r, S^r)$.

The chronological implementation of AL adds one more operation to NAL: whenever a vehicle (or flow) reaches the end of a link, information about the currently realized link travel times is collected, and the next link is determined from the policy definition. The chronological implementation requires a non-trivial change to the codes of NAL. We do not carry out this implementation in this thesis. Instead, we choose to use the iterative approach where an existing NAL is treated as a black box, which is a more convenient way of making use of NAL, as no rewriting of NAL codes is needed.

We will elaborate on the iterative approach. We know that the desired AL loads policy flows ($f \times D^r$) onto the network with supply parameter S^r, and outputs link travel times C^r. As a by-product of the loading, we should also obtain corresponding path splits, as each user will end up taking an actual path. Denote the corresponding path splits as pf, and we have $C^r = NAL(pf, D^r, S^r)$. On the other hand, for a given support point of link travel times, a policy can be translated into a path (details will follow about the translation), and thus policy splits into path splits. Denote the translation as V and we have $pf = V(f, C^r)$. Now we can formulate the adaptive loading problem as a fixed point problem:

$$pf = V\left(f, NAL\left(pf, D^r, S^r\right)\right)$$

A method of successive averages (MSA) can be used to solve this fixed point problem. There is no proof of convergence, and we will carry out computational tests in Chapter 6 to check the convergence experimentally.

Iterative Adaptive Dynamic Loading Algorithm with Support Point D^r, S^r

Step 0 (Initialization)

0.1: N = maximal number of iterations;

0.2: $n = 0$ (the iteration counter);

0.3: $C^r_{(n)}$ = free flow travel times;

0.4: $pf_{(n)} = M(f, C^r_{(n)})$

0.5: $n = n + 1$

Step 1 (Main Loop)

1.1: Non-adaptive loader: $C^r_{(n)} = NAL(pf_{(n-1)}, D^r, S^r)$

1.2: Policy-to-path translation: $pf = V(f, C^r_{(n)})$

1.3: Path flow update: $pf_{(n)} = (1-\alpha) \times pf_{(n-1)} + \alpha \times pf$

where $\alpha = 1/n$

Step 2 (Stopping Criterion)

If $n = N$, then $C^r = NAL(pf_{(N)}, D^r, S^r)$ and STOP

Otherwise, $n = n + 1$, and go to Step 1

The iterative process will be stopped after a maximum number of iterations is reached. This thesis does not provide mathematical proof of the existence of uniqueness of the fixed point problem solution. In Chapter 6 where computational tests are carried out, a convergence check is performed for this fixed point problem, ensuring the difference between two successive iterations is small enough.

We talk more about the translation from policy to path with a given joint realization of link travel times. As we know, a policy is defined over a discrete joint distribution of link travel times $\tilde{C} = \{C^1, C^2, ..., C^R\}$. A computer representation of a routing policy in the variant POI (Section 2.4) is composed of three parts:

- A matrix storing the discrete joint distribution of link travel times $\{C^r_{jk,t}, \forall\{j,k\} \in A, \forall t \in T, \forall r \in \{1,2,...,R\}\}$.

- A matrix storing all the possible current information at all times. An example is shown earlier in Figure 2-5 when the POI variant is studied. Basically, at each time period, the support points that belong to the same current information (or event collection as in the variant POI) are grouped, and any current information can be presented by any of the support points belongs to it. Note the current information is a function of time only, since the POI variant assumes perfect online information, i.e. the information is globally available to travelers anywhere in the network.

- A matrix storing next nodes associated with all possible current information as defined in the previous matrix. If two support points belong to the same current information at a given time, their next nodes are the same.

Denote the given realization of link travel times as C'. At a given time t and at a given node j, we would like to find the support point index r such that the link travel times as defined by the current information are the same in C^r and in C'. For example, in the variant POI (Section 2.4), the current information contains realized travel times of all links up the current time. Thus all link travel times up to time t should be the same in C^r and C'. After that, the policy would map the state triple (j, t, r) to a next node. We continue this operation until the destination is reached, and the sequence of nodes traversed constitute a path.

There are cases when we cannot find such an r. An example is in the iterative adaptive loading algorithm we just presented. We start from free flow link travel time, and update it using MSA. Therefore the link travel times differ from iteration to iteration, and there is no reason to believe that each of them is exactly the same as one of the support points of the link travel time distribution used in the policy definition. In the case that there does not exist an r such that $C^r = C'$ at some point, we could instead find the r, such that $r = \arg\min_{r=1,\ldots,R} ||subset(C^r) - subset(C')||_2$, where the subset function is dependent on the definition of current information, the current time and the current node.

To understand this problem from a higher level, one can view it as a "pattern recognition" problem. A traveler has a knowledge database of link travel times that is built from past experiences on a certain range of traffic conditions: good or bad weather, incidents on certain links or not, and etc. The traveler also has a decision rule that tells him/her what to do for each link travel time pattern. However, the traffic condition unfolded to the traveler on a particular day might not match exactly any of the defined patterns in his/her database. The traveler then has to make a judgment to choose the pattern closest to reality. There are various criteria for defining closeness, and the method of taking the second norm as described above is one of them.

5.5.1 Algorithm of Policy-Based Dynamic Network Loading Model

We give a generic algorithm for the policy-based network loading model. The DNL model takes as input the policy splits $f^i_{jd,t}, \forall$OD $(j,d), \forall t, \forall$policy μ_i, the PMF of stochastic demand/supply $\tilde{D}, \tilde{S}$, and outputs the link travel time distribution $\{C^r_{jk,t}, \forall (j,k) \in A, \forall t, \forall r \in \{1, ..., R\}\}$. The statement of the algorithm is as follows:

Policy-Based Dynamic Network Loading Heuristic

Step 0 (Initialization)

0.1 $r = 1$ (the counter for the number of demand/supply PMF support points)

Step 1 (Loading)

1.1 Perform a single loading: $C^r = AL(f, S^r, D^r)$

where AL is implemented in "Iterative Adaptive Dynamic Loading Algorithm"

1.2 $p_r =$ the probability associated with (S^r, D^r)

Step 2 (Stopping Criterion)

If $r = R$, STOP

Otherwise $r = r + 1$, and go to Step 1

It is difficult (if not impossible) to derive an analytical function that relates the PMF of demand/supply and the PMF of link travel times. Therefore the presented loading method is essentially simulation-based, such that for each possible value of the demand/supply, we do a single loading and obtain one support point and the corresponding probability for the PMF of the link travel time distribution.

5.6 The Stochastic Policy-based DTA Solution Algorithm

In Section 5.3, we formulated the policy-based stochastic dynamic traffic assignment problem as a fixed point problem on the distribution of link travel times. We can also describe the equilibrium by the following equilibrium conditions:

Definition 5.6.1 (Policy-Based Stochastic Dynamic Traffic Equilibrium). *A traffic network is in policy-based stochastic dynamic equilibrium, if each user follows the routing policy with minimum perceived disutility at his/her departure time, and no user can unilaterally change routing policies to improve his/her perceived disutility.*

Disutility is the negative of utility in the users' policy choice model. Conventionally, a random utility model assumes that users maximize utility. Since in a policy choice context, the coefficients of the mean travel time, travel time variability, and expected schedule delay, are usually negative, we rephrase it by saying "minimum disutility". The phrase will become "minimum mean travel time" when mean travel time is the only criterion in policy choice, which is consistent with the phrase in traditional SUE. We use the word "perceived", because we assume random perception errors associated with the disutility.

The idea of the solution algorithm is to find a solution to the fixed point problem (Equation 5.1) by an iterative process on policy splits. At each iteration, the policy splits are updated by combining the results from the current iteration and previous iterations. Since no proof of convergence is available at this moment, the method is heuristic for the DTA problem. The algorithm is presented as follows:

Policy-Based Stochastic DTA Heuristic

Step 0 (Initialization)

0.1: N = maximal number of iterations;

0.2: MSA counter $i = 1$

0.3: $C^r_{(0)}$ = free flow link travel times, $r = 1, ..., R$

0.4: Policy choice set $G_{(0)} = \{paths\}$

0.5: Policy splits $f_{(0)} = 0$

Step 1 (Main Loop)

1.1: Generate an optimal routing policy $\mu_i = O\left(\tilde{C}_{(i-1)}\right)$

1.2: Choice set update $G_{(i)} = G_{(i-1)} \cup \{\mu_i\}$

1.3: Users' choice model $f' = U\left(G_{(i)}, \tilde{C}_{(i-1)}\right)$

1.4: MSA update $f_{(i)} = (1-\alpha)f_{(i-1)} + \alpha f'$, where $\alpha = 1/i$

1.5: Loader $\tilde{C}_{(i)} = L\left(f_{(i)}, \tilde{D}, \tilde{S}\right)$

Step 2 (Stopping Criterion)

If $i = N$, STOP

Otherwise, $i = i + 1$, and go to Step 1

A reasonable value for the maximum number of iterations will be obtained by running the heuristic for a sufficiently large number of iterations and observing the convergence property. Experimental results on this topic will be presented in the next chapter. In Step 0.4, we initialize the policy choice set to include all paths that would have been included in a choice set for a path-based DTA model. Note that for each OD pair and each departure time, there is a choice set, and the initialization is done for all choice sets. The subscripts for OD pair and departure time are omitted to avoid heavy notation. In Step 0.5, we initialize policy splits to be zeros for all OD pairs and departure times. These are infeasible policy splits, and the initialization is just for the convenience of writing a formula in Step 1.4. $f_{(0)}$ is not taken into account in the MSA update, as when $i = 1$, its coefficient is zero.

In each MSA iteration, we generate a new PMF of link travel time distribution, based on the PMF of link travel time distribution from the last MSA iteration. The process is a sequential application of the optimal routing policy algorithm, the users' policy choice model, and the policy-based network loading model. The PMF of link travel time distribution from the last iteration is used in two models: first an optimal routing policy is generated based on this PMF and added to the choice set, then attributes of policies in the choice set are calculated based on this PMF and used in the users' policy choice model. Policy splits obtained from the choice model are then combined with those from the last iteration, and are loaded into the network to generate the PMF of link travel times.

Note that in the users' choice model, attributes of a routing policy are calculated based on the link travel time distribution from the last iteration $\tilde{C}_{(i-1)}$, not based on the link travel time distribution that defines the specific routing policy. As a matter of fact, each routing policy in the choice set is defined over different link time distribution, as each of them is generated in different MSA iteration. In the calculation, we will encounter the same problem as discussed in the network loading model that we might not be able to find an exact match between the current information in a policy's definition and the actually available link travel time realizations. The solution to this problem is presented in Section 5.5.

5.7 Concluding Remarks

In this chapter, we present a routing policy based stochastic dynamic traffic assignment model with the following distinctive features:

1. Users are assumed to take adaptive routing policies
2. The traffic network is described by a general distribution of link travel times

To understand the policy-based DTA model from a high level, one can view it as a mapping from a joint distribution of time-dependent supply/demand to a distribution of time-dependent link travel times, and users make optimal adaptive routing choices based on the link travel time distribution.

We formulate the DTA model as a fixed point problem and propose an MSA heuristic. There is no proof of convergence, and computational tests will be presented in the next chapter to study the convergence properties.

Chapter 6

Computational Tests

In Chapter 5, we established a policy-based stochastic dynamic traffic assignment model, composed of three parts: users' policy choice model, policy-based dynamic network loading model, and optimal routing policy algorithm. In this chapter, we will present a specific computer implementation of the DTA model, and carry out computational tests. The objectives of the computational test are multiple:

- Identify the implementation details of the proposed policy-based DTA model;
- Study the convergence properties of the MSA heuristic;
- Check the validity of the policy-based DTA model and study its behavior through sensitivity analyses;
- Explore the value of online information by comparing results from non-adaptive routing based (path-based) model and adaptive routing based (policy-based) model.

This chapter is organized as follows. First in Section 6.1, we present the four models that will be implemented. In addition to the policy-based DTA is certainly one of them. We have three other models, as we would like to compare results from a policy-based DTA with those from other more traditional models and thus gain more insight into the policy-based model. Implementation details of the the first three models as specializations of a general policy-based DTA model are provided. The test design is presented in Section 6.2, which gives the network to be tested, probabilistic descriptions of random supply, specifications of users' choice model and information access, and the simulation parameters. Results from the tests then follow. In Section 6.3.1, we study the convergence property of the MSA heuristic proposed in Section 5.6. In Section 6.3.3, two sets of sensitivity analyses are carried out with respect to incident probability and market penetration of online information.

6.1 Comparison of Four Models

6.1.1 The Four Models

The motivation for the policy-based DTA model is to be able to model users' adaptive choices in a truly stochastic network. Some interesting questions then arise. What if we do not model the adaptive choices? What if the online information is utilized in a different way than that described in the optimal routing policy algorithm? The comparison of results from different models will give us a better understanding of the policy-based model. We develop four models for comparison purposes as shown in Table 6.1.

	Base	Path Based	Online Path Based	Policy Based
Distributions of demand/supply	No	Yes	Yes	Yes
Online information	No	No	Yes	Yes
Optimal online choice	No	No	No	Yes

Table 6.1: Four Equilibrium Models

In each column, we have one equilibrium model: base model, path based model, en route path based model, and policy based model, respectively. We have three features listed: distributions of demand/supply, online information, and optimal online choice, which specify respectively whether distributions of random demand/supply are considered, whether online information is utilized in routing decision making, and whether online information is utilized optimally by applying the optimal routing policy algorithm developed in Chapter 2. We elaborate on the models one by one.

The first model is the base equilibrium model. It is an assignment model in a deterministic network with deterministic demand. It ignores stochastic disturbances in supply, e.g. assumes no incidents at all in a network. On the other hand, the demand is set at its expected value, if any stochasticity in demand exists. This corresponds to the case where users have no idea about the incident at all and just follow their habitual paths in a normal network. After the equilibrium path flows are obtained, they are loaded onto the true network with stochastic demand and supply, and the resulting measures of effectiveness are calculated.

The second model is the path based model with equilibrium in distribution. In this model, the distributions of both demand and supply are known and are used in the assignment. We seek equilibrium in the distribution of link travel times. Users are assumed to take paths with minimum disutility, which is a linear combination of path attributes such as expected travel time, travel time standard deviation, etc. Please refer to Section 5.4.3 for details. Although in that section, the choice model is based on policies, it can be used in a path-based assignment model by simply restricting the choice set to include paths only. We emphasize that a path is a fixed set of concatenated links. If a user follows a path, then s/he will traverse this set of links one by one, regardless of any online information. Note that online information includes information at the origin node and it should not be restricted to information collected *en route*. Basically it is any information beyond the *a priori* knowledge about the distribution of link travel times.

The third model is the online path based model with equilibrium in distribution. It makes use of online information as compared to the previous model. With the equilibrium link travel time distribution, a user makes routing decisions as follows. For any given state, i.e. node, time and current information, the conditional link travel time distribution is obtained and a path with minimum disutility is sought. Again the disutility is a linear combination of path attributes such as expected travel time and travel time standard deviation. The user then takes the first link of the path. When s/he arrives at the next node with an arrival time and updated online information, a new conditional distribution is obtained with a new minimum disutility path. The user continues on the first link of this new path. The above steps are repeated until the destination is reached. We remark that the outcome of the process is also a routing policy, in the sense that it is a mapping from any state to a next node. It is just that the routing policy is not generated optimally, as each decision is made assuming that no further information will be available. We term a routing policy generated in the above stated process as "online path". On the other hand, since the information is updated quite often (at the same pace as in an optimal routing policy algorithm), the online path could be a good approximation to an optimal routing policy.

The last model is the optimal routing policy based model with equilibrium in distribution. It is different from the online path model, in the sense that it makes optimal use of online information. We note that the routing decisions in both models are link based, in the sense that only a next link is chosen at each decision node. However, the attractiveness or utility of a link is evaluated differently in these two models. In the online path choice model, the utility of a link is based on only one path; while in the optimal routing policy model, the utility of a link is based on a set of paths that have this link in common. Intuitively the second approach should lead to better decisions.

By comparing results of the base model and the path model, we can study the value of *a priori* information on stochasticity of demand/supply, as the base model ignores the stochasticity of demand/supply while the path model makes use of the *a priori* knowledge on distributions of demand/supply. By comparing results of the path model and the last two models (online path model and policy model), we can study the value of online information. Finally, by comparing results of the online path model and policy model, we can study the value of making optimal use of online information.

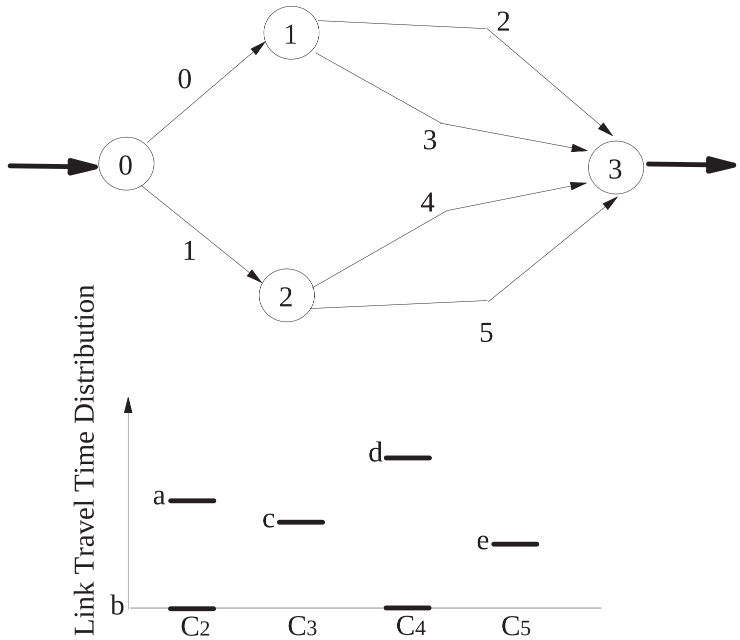


Figure 6-1: An Illustrative Example for the Difference Between the Online Path Model and Policy Model

6.1.2 An Illustrative Example

The difference between the online path model and the policy model might be hard to comprehend, and an illustrative example could be helpful. Figure 6-1 gives the topology of the network and an illustration of distributions of relevant link travel times.

This example is actually for routing only, and there is no supply/demand interaction involved, as the difference between the two models lies in how a routing decision is made with fixed network characteristics. Travelers are going from node 0 to node 3. The network is static. Travel times on links 0 and 1 are the same and deterministic, and thus can be omitted from the calculation of expected travel times of any path or policy between nodes 0 and 3, as they will not make any difference. Distributions of travel times on links 2, 3, 4 and 5 are as follows. C_i denotes the travel time on link i.

$$C_2(x_2) = \begin{cases} a, & w.p.\ 0.5 \\ b, & w.p.\ 0.5 \end{cases} \qquad C_3(x_3) = \begin{cases} c, & w.p.\ 1.0 \end{cases}$$

$$C_4(x_4) = \begin{cases} d, & w.p.\ 0.5 \\ b, & w.p.\ 0.5 \end{cases} \qquad C_5(x_5) = \begin{cases} e, & w.p.\ 1.0 \end{cases}$$

The relationship among these letters are: $e < c < a < d, (a+b)/2 < e$.

The network settings can be understood as follows. Links 2 and 4 have the same travel time b in normal conditions. There are chances that incidents happen on the two links. The incident on link 4 is more severe than that on link 2, such that the travel time on link 4 under incident (d) is higher than that of link 2 (a). On the other hand, both links have alternative links for diversion. Link 2 has link 3 as the diversion link with a travel time of c, and link 4 has link 5 as the diversion link with a travel time of e. Link 5 is a better diversion than link 3, as $e < c$.

A critical assumption here is the online information access. It is assumed that actual realizations of travel times on outgoing links are known when a traveler is at a node.

First the routing decision process in an online path model is studied. At node 0, four paths are available: the path passing link 2 with an expected travel time $(a+b)/2$, the path passing link 3 with an expected travel time c, the path passing link 4 with an expected travel time $(d+b)/2$, and the path passing link 5 with an expected travel time e. The path passing link 2 has the minimum expected travel time, and thus the traveler takes link 0 which is the first link along that path. Upon arriving at node 1, the traveler learns the actual realizations of travel times on links 2 and 3, and chooses the faster one, i.e. link 2 w.p. 0.5 and with a travel time b and link 3 w.p. 0.5 and with a travel time c. Therefore the expected travel time from node 1 to node 3 is $(b+c)/2$. In summary, the expected travel time from node 0 to node 3 following the routing process in the online path model is $(b+c)/2$ plus the travel time on link 0.

Then the routing decision process in a policy model is studied. At node 0, the traveler is actually comparing the attractiveness of links 0 and 1. An optimal routing policy should consider the fact the once the traveler arrives at the next node, he/she will follow an optimal routing policy from there to the destination. From the analysis of the previous paragraph, the minimum expected travel time from node 1 to node 3 is $(b + c)/2$ and the corresponding optimal policy is to take the faster one of links 2 and 3. Similarly, the minimum expected travel time from node 2 to node 3 is $(b+e)/2$ and the corresponding optimal policy is to take the faster one of links 4 and 5. With this calculation in hand, the traveler evaluates at node 0 and decides that taking node 2 is optimal ($(b+e)/2 < (b+c)/2$). In summary, the expected travel time from node 0 to node 3 following the routing process in the policy model is $(b + e)/2$ plus the travel time on link 1. As the travel times on links 0 and 1 are the same, and $e < c$, the routing in the policy model is more efficient and makes better use of online information than that in the online path model.

6.1.3 Implementations

We implemented the four models presented using the C++ programming language in a Red Hat Linux environment. The MSA heuristic for a policy-based DTA has been presented in Chapter 5, which is a general description of all the four models. To obtain an implementation of the base model, we apply the following restrictions to the general heuristic algorithm of policy-based DTA:

- Set the joint distribution of demand/supply to a single set of values with probability 1, where the demand is set at its expected value, and the supply is set as normal. If the supply stochasticity comes from incidents, we set the incident probability to be 0;
- Set the users' choice set to include paths only;
- Set the maximum number of iterations for the iterative network loader to be 1;
- Omit the optimal routing policy generation component.

The first two restrictions are self-explanatory. For the third restriction, recall that we use an iterative procedure over a non-adaptive loader to do adaptive loading. While in the base model, the routing choices are already non-adaptive (paths), we then do not need to execute the iterative procedure. For the last restriction, recall that we apply the optimal routing policy algorithm to generate one more alternative in the choice set at each MSA iteration. Since the base model is path-based and the choice set is composed of all feasible paths which are all generated in the initialization period, we do not need update the choice set dynamically. Therefore the optimal routing policy generation component is not needed.

To obtain an implementation of the path model, we apply the following restrictions to the general heuristic algorithm of policy-based DTA:

- Set the users' choice set to include paths only;
- Set the maximum number of iterations for the iterative network loader to be 1;
- Omit the optimal routing policy generation component.

Compared to the base model, we have one less restriction, in that the distribution of demand/supply is considered here.

To obtain an implementation of the online path model, we apply the following restrictions to the general heuristic algorithm of policy-based DTA:

- Modify the optimal routing policy algorithm to follow the decision rule of the online path model specified in Section 6.1.

The only difference between the online path model and the policy model, is that the decision rule is generated differently: a user in the online path model is assumed to follow a sub-optimal policy, versus an optimal policy in the policy model. A policy is optimal, if it is generated by implementing the optimality condition specified by Equations 2.2 and 2.3 in Section 2.2.1. From the descriptions above, we can see that a policy-based DTA model is a generalization of a path-based DTA model, where the members of a choice set are generalized from paths to routing policies. This generalization potentially allows for more flexibility in depicting travelers' behavior.

At the end of this section, we give a brief introduction on the non-adaptive loader utilized in the iterative dynamic network loader. DynaMIT is a computer system

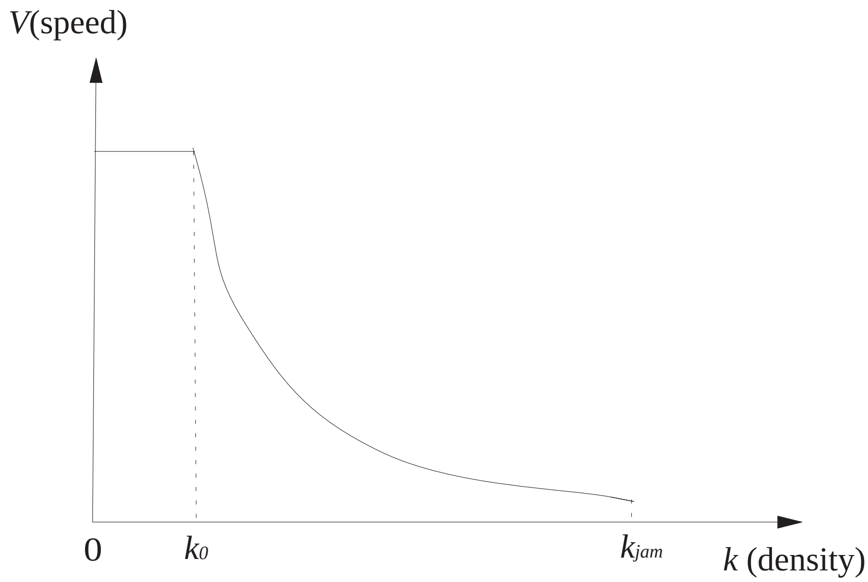


Figure 6-2: Speed-Density Relationship of DynaMIT Supply Simulator

for traffic estimation, prediction, and generation of traveler information and route guidance, developed by the ITS program of MIT (Ben-Akiva *et al* (1997) [10]). It contains a mesoscopic traffic simulator which allows for a detailed representation of traffic, where each individual vehicle can potentially be represented, and an analytical description of traffic dynamics, based on established traffic laws. Special care has been invested into representation of queues and spill-back phenomena. Indeed, these are the main sources of delays in an urban context, and need to be accurately captured by the simulator. We take the mesoscopic traffic simulator of DynaMIT as our non-adaptive loader.

Vehicles in the DynaMIT supply simulator move according to speed-density relationships. The vehicle speed on a segment is a non-linear function of its density, represented by the following relationship:

$$V = \begin{cases} V_{max} & , \quad k \leq k_0 \\ V_{max}\left[1 - \left(\frac{k-k_0}{k_{jam}}\right)^{\beta}\right]^{\alpha} & , \quad k > k_0 \end{cases} \tag{6.1}$$

The relationship is illustrated in Figure 6-2. Parameters α and β are to be determined for each segment through calibration using traffic data on representative segments. The jam density (K_{jam}), measured in vehicles per mile per lane can also be specified for each segment.

Vehicle trajectories are the output of the DynaMIT traffic simulator, including the time stamps at which each vehicle enters and exits each link along its path. The travel time of link a at time t is obtained by averaging travel times on link a of all vehicles that enter the link during $[t, t+1)$. Note t is the index for discretized time periods. Time-dependent path travel time is then obtained by composing link travel times along the path.

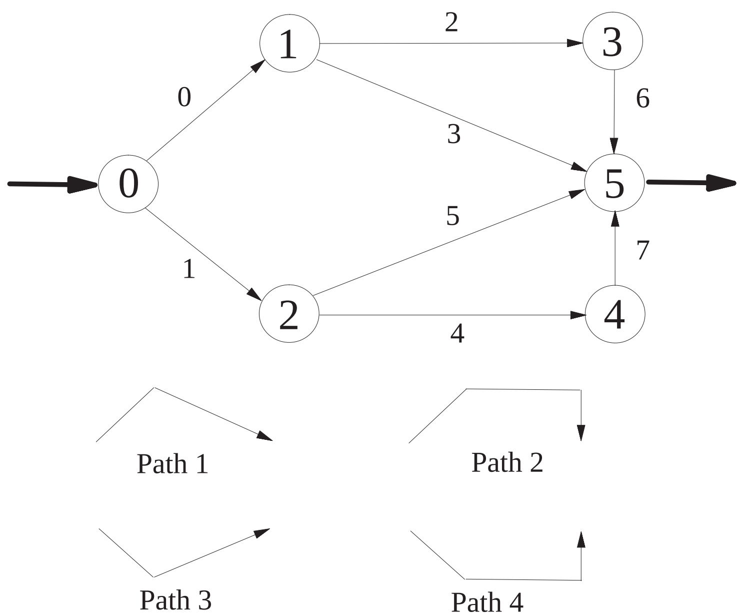
We make the DynaMIT loader deterministic by suppressing all random factors in the loader. This is to make sure that the stochasticity of the model comes solely from random incidents, and thus we can make better analysis of test results.

6.2 Experimental Design

6.2.1 The Test Network

We conduct computational tests on the simple hypothetical network shown in Figure 6-3. The network has 6 nodes and 8 directed links. It is symmetric with respect to the horizontal line passing through nodes 0 and 5. The link data is summarized and shown under the network.

We deal with one OD pair between node 0 and node 5. We assume zero flows between any other OD pair. Four paths exist for OD pair (0,5) as shown in Figure 6-3, with online diversion possibilities at nodes 0, 1 and 2. The study period is from 6:30am to 8:00am. The time resolution is 1 minute for the optimal routing policy algorithm and users' behavior model. The DynaMIT loader works at a finer resolution (5 sec) for the simulation, but the post-processed link (path) travel times are also by minute. Therefore we have 90 time periods in the tests.



	Link 0(1)	Link 2(4)	Link 3(5)	Link 6(7)
Length (mile)	0.5357	0.7576	0.8470	0.3788
Number of Lanes	2	1	1	1
Free Flow Speed (mph)	40	30	20	30
Free Flow Time (sec)	48	91	152	45
k_{jam} (veh/link/meter)	0.30	0.15	0.15	0.15
α	3.733	3.733	3.733	3.733
β	1.907	1.847	1.847	1.847
Output Capacity (veh/link/sec)	1.1	0.5	0.5	0.5
k_0	0.016	0.012	0.012	0.012

Figure 6-3: Test Network and Link Data of Policy-Based DTA

6.2.2 Random Incidents

We have random incidents in the network. A incident in DynaMIT is defined by the segment ID, start time, duration and capacity reduction factor. A segment is part of a link, and a link can be composed of one or multiple segments. In our network, each link is composed of only one segment. If an incident starts from 8:00am and lasts for 20 minutes with a capacity reduction factor 0.5 on link 0, then the output capacity of link 0 will be $0.5 \times 1.1 = 0.55$ veh/link/sec from 8:00am to 8:20am, and will revert to the original value 1.1 veh/link/sec from 8:20am on. As the capacity reduction is with respect to output capacity, an incident could only happen at the end of a link.

The random incident is defined as follows.

- There is at most one incident during the study period for any given day;
- The incident has a positive probability of occurring on link 0, 2, 3 and 6, but zero on links 1, 4, 5 and 7;
- The probability of incident occurrence on a link is proportional to the link's length (for links 0, 2, 3 and 6);
- If an incident occurs on a link, the start time can be 6:30am, 6:40am, 6:50am, ..., 7:50am with equal probability;
- The duration of any incident is fixed at 10min, and the capacity reduction factor is fixed at 0.3;
- The probability of no incident in the network is $1 - p$.

Based on the above description, the random incident can be described by the joint distribution of link ID l and start time t_0. Denote l_0, l_2, l_3, l_6 as the length of link 0, 2, 3 and 6 respectively and $L = \sum_{i=0,2,3,6} l_i$.

$$
(l, t_0) = \begin{cases} (0, \text{6:30 or 6:40 ... or 7:50}), & w.p.\ p \times l_0/L/9 \\ (2, \text{6:30 or 6:40 ... or 7:50}), & w.p.\ p \times l_2/L/9 \\ (3, \text{6:30 or 6:40 ... or 7:50}), & w.p.\ p \times l_3/L/9 \\ (6, \text{6:30 or 6:40 ... or 7:50}), & w.p.\ p \times l_6/L/9 \\ (\text{non-exist, non-exist}), & w.p.\ 1-p \end{cases}
$$

6.2.3 Demand

We assume that the demand for OD pair (0, 5) is always deterministic. The profile is shown in Figure 6-4. It is flat between 6:30am and 7:00am, and between 7:00am and 8:00am, with at a higher level.

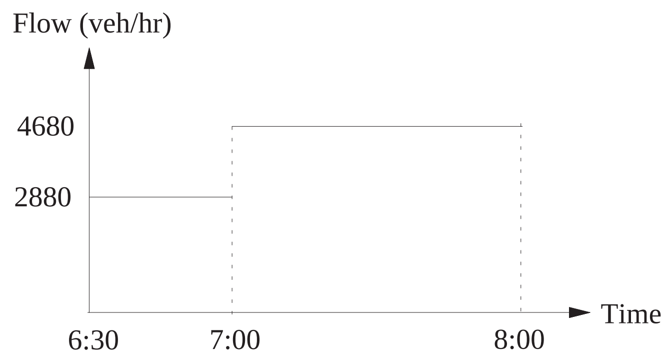


Figure 6-4: Demand for OD Pair (0, 5)

Users are assumed to minimize expected travel time with perception errors, i.e. the utility function of the policy-size Logit model in Section 5.4.3 takes into consideration the expected travel time. The coefficient of expected travel time is negative with a large enough absolute value (-6.0) to approximate a fastest policy (path) choice situation. All users have perfect online information in the online path model and policy model, i.e. knowledge of travel time realizations on all links up to the current time. Obviously, users have no online information in the base model and the path model.

6.3 Results

We carry out a wide range of computational tests to study the behavior of the policy-based stochastic dynamic traffic assignment model. First we study the convergence properties of the MSA heuristic proposed in Section 5.6. This builds the foundation for all later tests. Next we give a detailed comparison of the four model results for a specific test setting, so that a good understanding of the four models is gained. Finally we perform a series of sensitivity analysies with respect to incident probability and market penetration rate of online information, where stochasticity comes from random incidents.

Before presenting the results, we need to determine the period for which statistics of travelers (such as expected OD travel times) departing during this period are collected and compared. The study period is one and half hours long (from 6:30am to 8:00am), yet we should not view the whole period as valid for statistics collection. Two considerations are involved in determining the valid statistics collection period. First, an implicit assumption of the incident distribution is that there is zero probability of incidents outside the study period (6:30-8:00). This assumption is not realistic, as naturally random incidents can happen during any time of the day. In order to eliminate the effect of this assumption, we choose a statistics collection period such that travelers departing during that period are not affected by this assumption. Specifically, we should have:

- Incidents before 6:30am have no effect on travelers departing during the period;
- Incidents after 8:00am have no effect on travelers departing during the period.

Secondly, since OD travel times are needed in the equilibrium assignment, we require that the computation of OD travel times are correct. For travelers who cannot finish their trips before the end of the simulation, the computation of travel times cannot be correct. Therefore it is required that travelers departing during the statistics collection period can leave the network before 8:00am. We note that this criterion is also a sufficient condition for the requirement that incidents after 8:00am have no effect on travelers departing during the collection period. Indeed, if travelers leave the network before 8:00am, then any incident after 8:00am clearly has no effect on them.

We conduct a series of tests to determine the statistics collection period. Each of the tests is a path-based equilibirum assignment with one deterministic incident. Since we have 36 support points for the incident distribution, the series contain 36 tests. For the four tests with incidents starting at 6:30am (recall that an incident can occur on link 0, 2, 3, or 6), we observe that the effects of incidents disappear before 7:00am. Therefore we conclude that incidents before 6:30am have no effects on travelers departing after 7:00am. For all the tests, travelers departing before 7:30am are able to leave the network by 8:00am. To sum up, we set 7:00am 7:30am as the statistics collection period. Note that these tests are different from what we will carry out later, and they provide a good guess of the collection period. In later tests, we need to check that for each test, the statistics collection period is indeed valid.

Next we discuss the test results in detail. We include all graphic output in Section 6.6 for ease of reading.

6.3.1 Convergence Study

First we study the convergence of the iterative loader. Recall that we implement the adaptive network loader by iterating over a non-adaptive loader (see Section 5.5). An MSA method is used for updating path splits. We then check the relative differences of time-dependent path splits from two successive iterations. After checking quite a number of tests, we find that all relative differences reduce to less than 5% after 5 iterations. We then set the maximum number of iterations for the iterative loader to 5. Note that this is just a good guess and, in a few cases, we need more iterations.

During quite a few initial tests, we found a problem that does not arise in the theoretical development of the policy-based DTA model, but is significant in the implementation. Recall that routing policies are defined over link travel time distributions. At each MSA iteration, one routing policy is generated based on the link travel time distribution obtained in that iteration. However, as we start from free flow travel times, link travel time distributions at early iterations could be quite far away from the equilibirum distribution, and policies defined over them can be very misleading and make the MSA stick in some non-equilibrium point. This problem does not exist in a path-based equilibrium assignment model, as a path is completely defined topologically and has nothing to do with travel times. An intuitive idea for solution is that policies should be defined over distributions that are close to (if not equal to) the equilibrium values. We then adopt the reset method proposed by Cascetta and Postorino which is discussed in Bottom (2000) [13]. The method reset the MSA counter to 1 after i, $2i$, $3i$,... iterations. For example, if $i = 7$, then the MSA counter will be reset to 1 at the 7th, 21st, 42nd, ... iteration. By resetting the MSA counter, we eliminate the effects of all early policies and effectively use the latest link travel time distribution as an initial solution to a new round of MSA iterations. There is no guarantee that this method will lead to better performance of the MSA heuristic. However it is consistent with the promising idea that policies should be defined over link travel time distributions that are close to equilibrium values, and is thus adopted.

In principle, the convergence indicator is the time-dependent OD travel time in the statistics collection period for all support points. As an equilibrium is sought in distributions of traffic variables (link travel time, OD travel time, and etc), it is required that the convergence indicator is checked for each support point. When the OD travel time for any departure time in 7:00am through 7:29am and for any support point is stable for a satisfactorily number of iterations, it is believed that a convergence has been reached.

In practice, a more convenient convergence indicator is used, which is the time-dependent expected OD travel time in the statistics collection period. Conceptually, the expected OD travel time is the minimization criterion of an user's policy choice behavior. If this minimization crition is stable from iteration to iteration, it is not likely that an user would change his/her behavior from iteration to iteration. Computationally, in all the tests that have been carried out, the stability in expected OD travel time is always equivalent to the stability in OD travel time distribution (i.e., for all support points). Therefore, the convergence study will be based on expected OD travel times. Note that the expected OD travel time is aggregated for each time interval (1 min) over all trips departing during that time interval, so the total expected travel time can be obtained by multiplying time-dependent OD travel times with time-dependent demand and then summing over all time intervals. The total expected travel time is a higher-level measure of system state, and is interesting in some applications. The convergence processes for all four models are shown through tests with incident probability $p = 0.9$.

Figure 6-6 shows the convergence process of the base model, where no incident exists. It is a traditional path-based deterministic assignment model and we do not apply the reset method. Since the network is deterministic, the convergence indicator is then OD travel time rather than expected OD travel time. OD travel times are plotted against number of iterations for all 30 departure times from 7:00am to 7:29am. The x-axis represents number of iterations, while the y-axis represents expected OD travel times. We can see that OD travel time reaches a stable state very quickly, after less than 10 iterations.

Figure 6-7 shows the convergence process of the path model with the specified incident distribution. Again since the assignment is path-based, we do not apply the reset method whose major purpose is to reduce the effect of inaccurate link travel time distributions used for defining policies. Expected OD travel times are plotted against number of iterations for all 30 departure times from 7:00am to 7:29am. The x-axis represents number of iterations, while the y-axis represents expected OD travel times. The convergence is also quite evident, with stable expected OD travel time values attained after less than 10 iterations across all time periods.

Figure 6-8 shows the convergence process of the online path model. The graphs have the same x-axis and y-axis as in Figure 6-7. As we can see, generally expected OD travel times attain rather stable values after about 30 iterations. The spikes after iteration 7 and 21 are due to the reset of the MSA counter.

Figure 6-9 shows the convergence process of the policy model. The graphs have the same x-axis and y-axis as in the previous two figures. Again we see that the after around 30 iterations, expected OD travel times become stable. We note that the two adaptive models generally need more iterations than the two path-based models.

6.3.2 Solution Discussion

We discuss the solutions of the four assignment models and compare them when appropriate. Note that we focus on the statistics collection period 7:00am through 7:30am, although statistics for all time intervals are presented. Special caution should be taken when reading statistics close to 8:00am, as there are unfinished trips during that period and the calculation of travel times could be mistaken.

First we check the result of the most simple case: the base model. Figure 6-10 shows the time-dependent OD travel times at equilibrium from the base model. Strictly speaking, we term this result as nominal base travel times. The true base expected travel times are obtained by loading the base path flows into the network with random incidents. Since the network is symmetric with no incidents and the demand is symmetric, path flows are also symmetric. Path 2 (link 0-2-6) and path 4 (link 1-4-7) carry most of the flows and path 1 (link 0-3) and path 3 (link 1-5) carry only a small part, largely because of the large free flow travel times of link 3 and 5. Figure 6-11 gives the path flows at equilibirum. No congestion is observed in Figure 6-10, which is expected, as all link flows are below corresponding link capacities. The jump at 7:00am is due to the increase of demand at that time. The system is in a steady state, indicated by the relatively flat OD travel times across departure times.

Figure 6-12 shows the OD travel time as a function of departure time for all 37 support points when the base path flows are loaded into the random network. Each plot in the figure is of the 37 discrete supporting points for the distribution of OD travel times. The x-axis represents departure time, while the y-axis represents OD travel time. Recall that in support points 1 through 9, the incident is on link 0 and with 9 different start time from 6:30am to 7:50am. Then in support points 10 through 18, the incident is on link 2; in support points 18 through 27, on link 3; and in support points 28 through 36, on link 6; and finally in support point 37, there is no incident in the network. The incident link ID and incident start time are listed on the top of each graph in the figure. We can observe the effect of the incident in the solution. For example, for support point 13 shown in Figure 6-13, the incident on link 2 starts at 7:00am. We see an increase in OD travel time for travelers leaving a little earlier than 7:00am, as travelers leaving the origin not early enough before the incident will encounter the incident during their trips. We can see the increasing effect of the incident as the congestion builds up, and a decreasing effect as the incident is removed 10 minutes later. Throughout the support points, the incident effect is less obvious in the first half hour due to smaller demand during that period. On the other hand, incident effect is different when located on different links. Generally incidents on link 0, 2 and 6 have larger effect than those on link 3. This is largely because flows on link 0, 2 and 6 are much larger than those on link 3.

We then present the equilibrium OD travel time distribution of the path model in Figure 6-14. We still see the effects of incidents, but generally at a smaller scale than those in the base model. This is because the path model takes into account the incidents when assigning flows. Figure 6-15 gives the time-dependent path flows at equilibrium. Compared with those in Figure 6-11, we can see that paths 3 and 4 take more flows, and paths 1 and 2 take less flows. This is because of positive incident probabilities on paths 1 and 2, while zero incident probabilities on paths 3 and 4. Some flows will then be diverted from path 1 and 2 to path 3 and 4, until expected path travel times of all four paths are roughly equal. Note that the path flows remain constant for all support points, as the path model does not allow adaptive routing choices.

Next we present the equilibrium OD travel time distribution of the online path model in Figure 6-16 with the same format as in Figure 6-14. For ease of comparision, OD travel times from the path model are also plotted with dashed lines. Generally, the online path model gives lower OD travel time, and the savings are quite outstanding in some cases (e.g. when incidents are on link 4 and start from 7:00 and 7:10). This is largely due to the flexibility gained through adaptive routing. Figure 6-17 gives time-dependent path flow distributions for path 2, and we can see flows on path 2 change from different support points, while in a path model, path flows are fixed across support points. We omit the presentation of other path flow distributions for the sake of brievity. When an incident happens, affected links will have longer travel times. Furthermore, at different stages of an incident, the realized link travel times so far are different. For example, if we are at a point when an incident just begins, then link travel times along the time axis would be flat at normal values and then jump to higher values. If we are at a point when an incident just ends, then we would see a longer period during which link travel times are at high values. If we are at a point when an incident has ended for a while, then we would be able to see link travel times first increasing and then decreasing. To sum up, the message contained in the current realized link travel times makes us adaptive to incidents. A very distinctive feature of the flows in Figure 6-17 is the decrease around incident across all support points. As we arrange the graphs by the start time of incident, we can see a moving "pit" in path flow. This is more intelligent than deterministic path flows as in path model. Note that policy flows are deterministic. As a policy will manifest itself as different paths in different incident support points, path flows are random and we can talk about their distributions.

Figure 6-18 and 6-19 show the OD travel time distribution and path 2 flow distribution of the policy model respectively. We see very similar patterns in these two figures as in the online path model. In fact, these two models are both based on routing policies, and it is just that the methods of generating optimal routing policies are different. We expect that results from the two models are not significantly different in our simple test network, due to the limited diversion nodes. Further computational tests on larger networks are desirable to study the differences between these two models.

Next we compare expected OD travel times from all the four models in Figure 6-20. Expected OD travel time is the major measure of effectiveness in our tests. We observe that the path model gives lower expected OD travel times than the base model, and the two adaptive models (online path model and policy model) provide further travel time savings. Figure 6-21 gives the time-dependent OD time standard deviations. Although travelers are minimizing expected travel time only, their travel time variances are also reduced by taking adaptive routing choices. This is due to the fact that their travel times are reduced in incident scenarios, and thus more smooth across support points. Note that Figure 6-20 and 6-21 are statistical summaries (first and second moments) of Figure 6-12, 6-14, 6-16 and 6-18.

6.3.3 Sensitivity Analysis

In the previous section, we discussed in detail the results for a specific test setting (incident probability $p = 0.9$). We are also interested in learning the behavior of the models when we vary the incident probability. On the other hand, in reality online information could be provided only to part of the travelers, thus it is desirable to study how the traffic conditions change as a function of market penetration of online information. We define a single measure of effectiveness (MOE) to be compared in the sensitivity analysis, which is the expected OD travel time averaged over the statistics collection period: 7:00am through 7:29am.

First we carry out the sensitivity analysis with respect to incident probability p. We vary p from 0 to 1.0 by a step size of 0.1. The result is plotted in Figure 6-22 and summarized in Table 6.2. For each of the models, the average expected OD travel time increases as incident probability increases, but different model has different increasing rate. This increasing function seems intuitively correct, as a more likely incident increases the probability that a network is congested, and thus a higher expected travel time. We note that the policy model gives a higher value for $p = 0.9$ (216.06) than for $p = 1.0$ (216.00). We believe that this difference is too small to be significant, and are inclined to believe that they are the same.

The relationship for the base model is linear. The explanation is as follows. First, the path flows are the same for various incident probabilities, since the base model does not consider incidents at all. Then the OD travel time for each incident support point is calculated, and a weighted average is taken to obtain the expected OD travel time, where the weight is the probability of an incident support point. As we can see from the design of incident distribution, incident probabilities are linear functions of p. Therefore the expected OD travel time is also linear function of p. While in

p	Base	Path	Saving (%) (path vs. base)	Online Path	Policy	Saving (%) (policy vs. path)
0.0	208.7	208.7	0.0	208.7	208.7	0.0
0.1	211.2	211.1	0.008	210.4	210.1	0.478
0.2	213.6	213.5	0.020	211.1	210.9	1.206
0.3	216.0	215.1	0.419	212.3	212.0	1.420
0.4	218.4	216.7	0.762	212.9	212.6	1.892
0.5	220.8	218.8	0.906	213.0	212.8	2.725
0.6	223.2	219.9	1.471	214.3	214.2	2.603
0.7	225.6	221.9	1.662	215.3	214.9	3.160
0.8	228.1	223.5	1.978	215.4	215.1	3.790
0.9	230.4	224.6	2.533	216.2	216.1	3.815
1.0	232.9	226.0	2.938	216.8	216.0	4.445

Table 6.2: Expected OD Travel Time Averaged Over 7:00-7:29 (second) as a Function of Incident Probability (p)

other three models, random incidents are considered in the equilibrium process and equilibrium path (policy) flows differ when p differs. Therefore the relationship is in general nonlinear.

In general, the path model gives less expected travel time than base model, and the two adaptive models (online path model and policy model) give less expected travel time than the path model. The savings (path over base, and adaptive over path) increase as incident probability increases, both in absolute values and in relative percentage savings. The relative saving of the path model over the base model is in the range of $0 \sim 2.9\%$, and the relative saving of adaptive models over the path model is in the range of $0 \sim 4.4\%$. This increasing function suggests that values of both *a priori* and online information are more evident when traffic conditions are worse. This could be reasonable in reality when traffic conditions without incident are not too congested, as then there is enough room for diversion. This is actually the setting of our tests, recalling that traffic is almost in free flow state with no incident (see Figure 6-10). We expect that when a network is already quite congested without incident, this function might become flat after some point.

Next we carry out sensitivity analysis with respect to market penetration of online information. For a given penetration k which is a value between 0 and 100%, we assign k of the demand to take minimum expected travel time routing policies, while the remaining $1 - k$ of the demand to take minimum expected travel time paths. Equilibrium is sought by an MSA heuristic that updates the path splits and policy splits simultaneously. The algorithm statement is presented below.

Stochastic DTA Heuristic with Online Information Penetration k

Step 0 (Initialization)

0.1: N = maximal number of iterations;

0.2: MSA counter $i = 1$

0.3: $C^r_{(0)}$ = free flow link travel times, $r = 1, ..., R$

0.4: Policy choice set $G_{(0)} = \{paths\}$; Path choice set $H = \{paths\}$

0.5: Policy splits $f_{(0)} = 0$; Path splits $g_{(0)} = 0$

Step 1 (Main Loop)

1.1: Generate an optimal routing policy $\mu_i = O\left(\tilde{C}_{(i-1)}\right)$

1.2: Choice set update $G_{(i)} = G_{(i-1)} \cup \{\mu_i\}$

1.3: Users' choice model $f' = U\left(G_{(i)}, \tilde{C}_{(i-1)}\right)$

$g' = U\left(H_{(i)}, \tilde{C}_{(i-1)}\right)$

1.4: MSA update $f_{(i)} = (1 - \alpha)f_{(i-1)} + \alpha f'$

$g_{(i)} = (1 - \alpha)g_{(i-1)} + \alpha g'$, where $\alpha = 1/i$

1.5: Loader $\tilde{C}_{(i)} = L\left(kf_{(i)}, (1 - k)g_{(i)}, \tilde{D}, \tilde{S}\right)$

Step 2 (Stopping Criterion)

If $i = N$, STOP

Otherwise, $i = i + 1$, and go to Step 1

The optimal routing policy (minimum expected time path) algorithm, the users' choice model, and the MSA update work in the same way as in separated policy model (path model), but the loading is carried out by assigning $1 - k$ of the demand to path flows and k of the demand to policy flows.

We have the result for $p = 0.1$ in Figure 6-23, that for $p = 0.5$ in Figure 6-24, and that for $p = 0.9$ in Figure 6-25. We see similar trends in all figures. The average expected OD travel time is at its largest value when market penetration of online information is zero. At that time, if one traveler is intelligent enough and take a routing policy rather than a path, he/she can save travel time. More and more of them find the benefits of online information, and they gain travel time savings and thus bring down the average expected travel time. However, in a congested traffic network, the changing of users' behavior changes the network-wide traffic conditions through interaction between supply and demand. As seen from the figure, the saving in travel time becomes less evident when penetration goes from 20% to 40% and from 40% to 60%. Later on, higher penetration actually does not bring any more savings. We see an increase in travel time from 60% to 80%. At las from 80% to 100%, for $p = 0.1$, the increase continues, and for $p = 0.5$ and $p = 0.9$, a slight decrease in travel time is observed. We then conclude that the savings gained from online information is larger when market penetration is lower. After some point, more online information could actually make things worse. Therefore the function of travel time saving against market penetration is not monotonic. Despite the varying effect of online information, travel time savings are always positive with online information, compared to no-online-information case. This analysis might only be valid for the test setting, and caution should be taken if one intends to generalize the result.

6.4 Tests with Stochastic Demand

A limited number of tests have been carried out with stochastic demand and deterministic supply. These tests are not as comprehensive as those with stochastic supply, and the objective of these tests is gain some initial insight into the convergence property of the MSA heuristic with stochastic demand and how users adapt to random demand. Note that in these tests, the supply is deterministic (i.e. no random incident), such that the only source of randomness is the demand.

Test Design

There are no incidents in the network at any time. The OD demand between nodes 0 and 5 remains the same as in all previous tests, shown in Figure 6-4, and thus is still deterministic. The demand between nodes 1 and 5 is assumed to be random with five support points, each with probability 0.2, and is shown in Figure 6-5. The randomness comes from the start time of the 20-minute peak in OD demand. This peak increases the congestion level on links 2, 3 and 6, and in this sense, it can be viewed as mimicking the effect of incidents on links 2, 3 and 6. The demand between nodes 2 and 5 is still zero. Users' behavior assumptions are the same as in previous tests and are applicable to all users for all OD pairs.

Three models will be tested: the path model, the online path model and the policy model. The base model is eliminated, as the definition of a base model is not clear here. In the previous random incident tests, equilibrium path flows are first obtained by assuming no incidents in the network, and then the expected OD travel time under a base model is obtained by loading the path flows from the base model to a network **with** random incidents. Correspondingly, the base model in a stochastic demand situation supposedly works as follows. First, equilibrium path flows are obtained by assuming no demand between nodes 1 and 5. Secondly, the path flows are loaded into the network with random demand between nodes 1 and 5. However, since no assignment result is available for demand between nodes 1 and 5, the above loading cannot be performed.

Results

Statistics are collected for users traveling between nodes 0 and 5 only. They are faced with random surge in downstream demand which will affect their travel times. The same statistics collection period 7:00 through 7:29 is used, such that there is a warm-up period 6:30 through 6:59 and there is enough time for travelers to leave the network by 8:00.

Convergence results of the three models are shown in Figures 6-26 through 6-28.

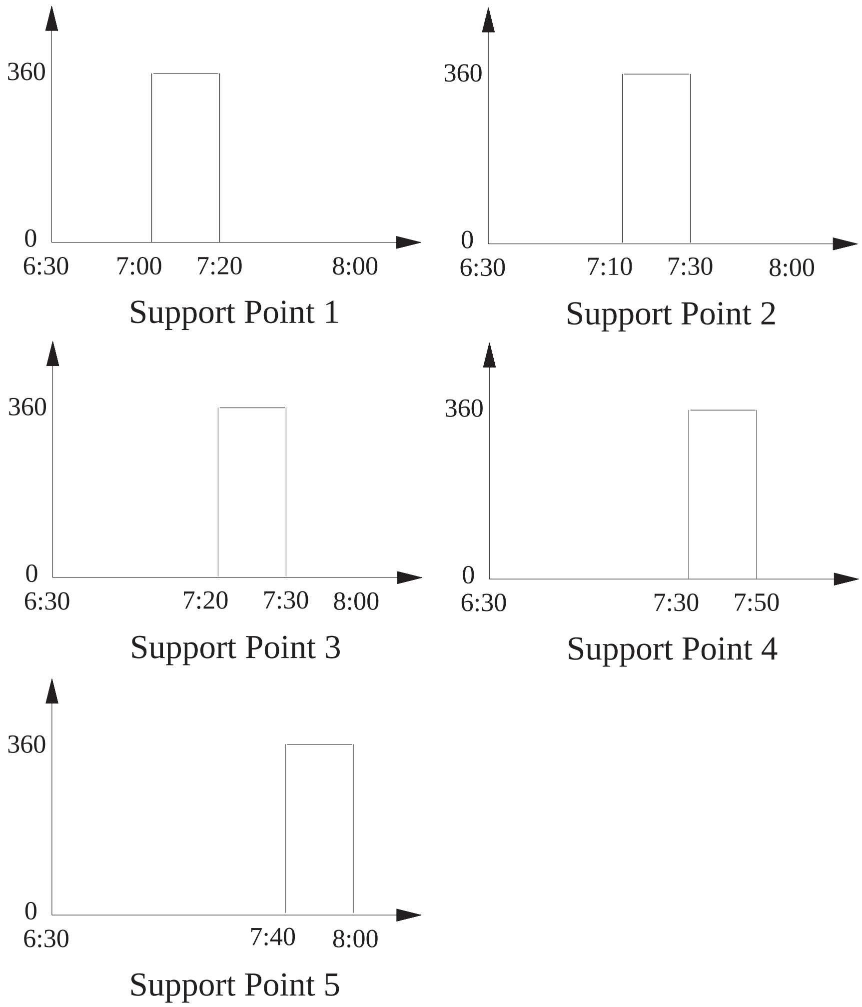


Figure 6-5: The Distribution of Random Demand Between OD (1,5) (Each Support Point with Probability 0.2)

The same convergence indicator is used which is the time-dependent expected OD travel time between nodes 0 and 5 during the statistics collection period 7:00 through 7:29. All three models converge satisfactorily within the given number of iterations.

Figure 6-29 shows the distribution of OD travel time between nodes 0 and 5 in the statistics collection period for both the path model and the policy model. Results from these two model are put together for the ease of comparison. Dotted lines represent results from the path model and solid lines represent results from the policy model. In Figure 6-30, the comparison is made between the path model and the online path model. The five support points correspond to the support points of the random demand shown in Figure 6-5, i.e. support point 1 corresponds to the case where the peak in demand starts from 7:00, and so on.

In both figures, the effects of random demand between nodes 1 and 5 are evident. For example, for support point 1 where the peak is present between 7:00 and 7:20, the OD travel times between nodes 0 and 5 for both models increase during 7:00 and 7:20. The policy model, however, manages to reduce the travel time compared to the path model. The same observation can be made for support points 2 and 3. The difference between the path model and the policy model is not evident in support points 4 and 5, partly because the calculation of travel times after 7:30 is not assured to be correct, as not all travelers departing after 7:30 can exit the network by 8:00.

Figure 6-31 shows the expected OD travel time between nodes 0 and 5 for the three models. For most part of the statistics collection period, the reduction in expected travel time from the path model to the two adaptive models is quite small, but still discernible. For some departure times, the two adaptive models actually give higher expected OD travel time than the path model does. Overall, the savings are positive and this shows the value of online information (or adaptive routing) in the stochastic demand situation. It is hard to make a direct comparison between the adaptive savings in the stochastic incident situation and those in the stochastic demand situation, as it is hard to derive the relationship between an random incident and a random OD trip. More tests are desirable to study the savings as a function of test parameters, such as the random demand scale.

Tests with higher demand scales are carried out, but without satisfactory convergence performance. It is found that large oscillation persists even after a very large number of MSA iterations. Investigation into this phenomenon is ongoing and will be one of the topics for future research.

6.5 Concluding Remarks

In this chapter, we implement the policy-based stochastic DTA model as well as three other models (base model, path model, online path model) to be compared with the policy model. We then carry out some computational tests on a simple network to obtain understanding of the policy model.

The MSA heuristic with reset works reasonably with the test network under the specific test settings. As there is no proof of convergence for the heuristic, caution should be taken when we change the test network and settings.

We analyze the behavior of adaptive models and find that the adaptiveness leads to travel time savings. As a byproduct, travel time reliability is also increased, as the travel time peak caused by a random incident is cut.

Sensitivity analysis shows that the value of online information is an increasing function of incident probability for specific demand scale and network topology. This result should be generalized with caution. It is also shown that online information penetration plays an important role in travel time savings. The savings are high when market penetration is low. The function of travel time savings against penetration is not monotonic. This suggests that when implementing a traveler information system, one needs to consider the penetration of information to maximize benefits.

6.6 Graphic Output

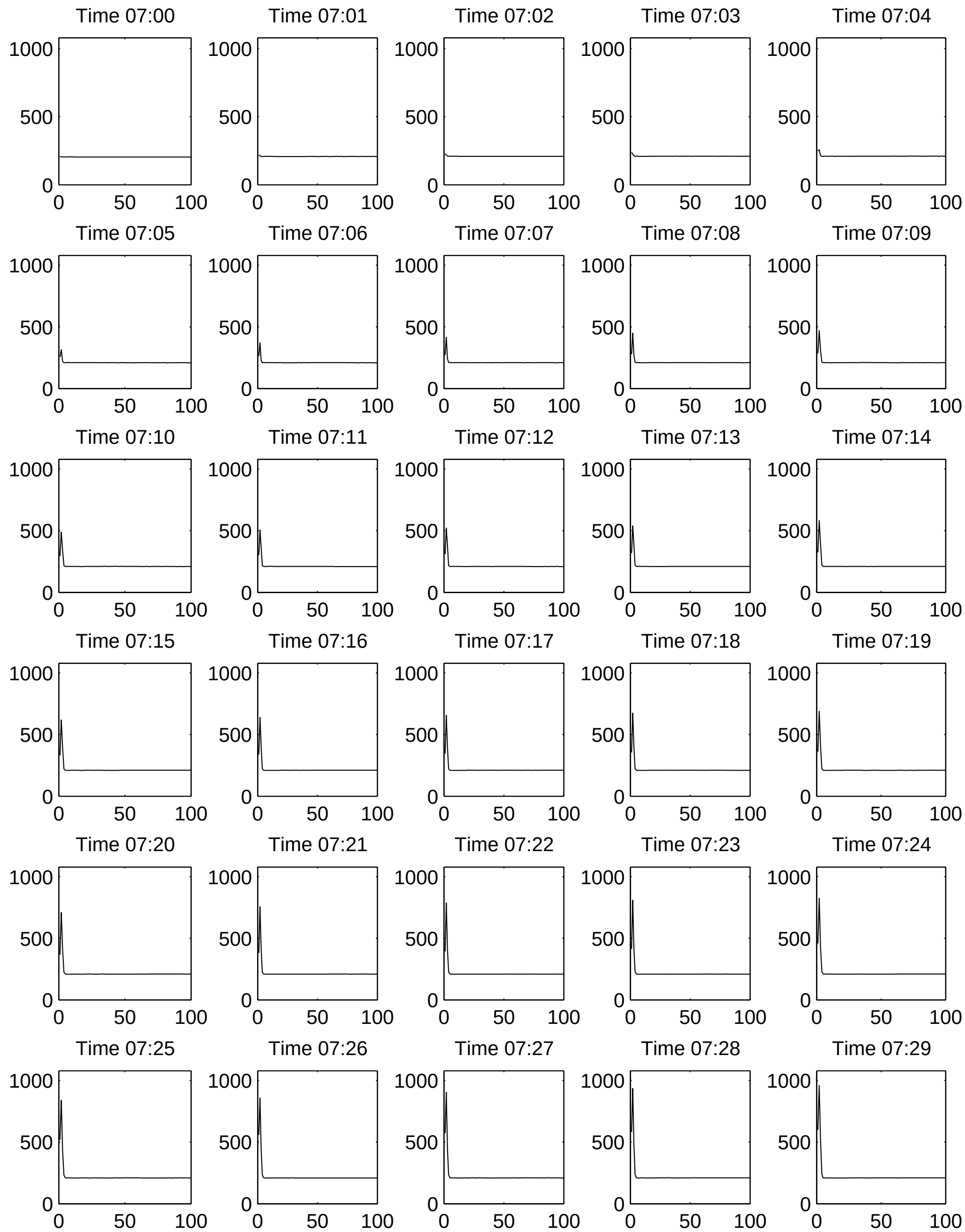


Figure 6-6: Convergence of Base Model (X-Axis: Number of Iterations; Y-Axis: OD Travel Time (sec); $p = 0.9$)

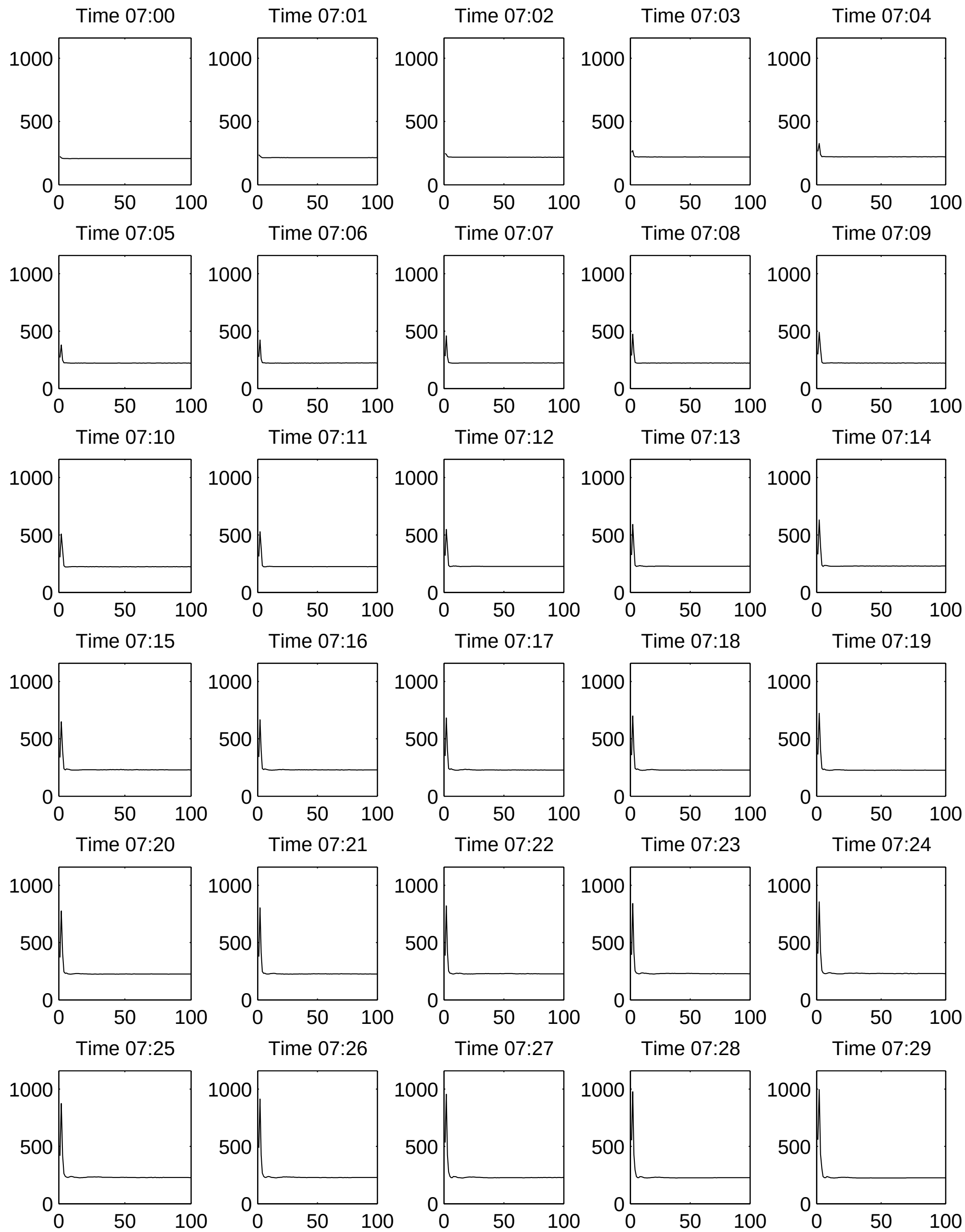


Figure 6-7: Convergence of Path Model (X-Axis: Number of Iterations; Y-Axis: Expected OD Time (sec); $p = 0.9$)

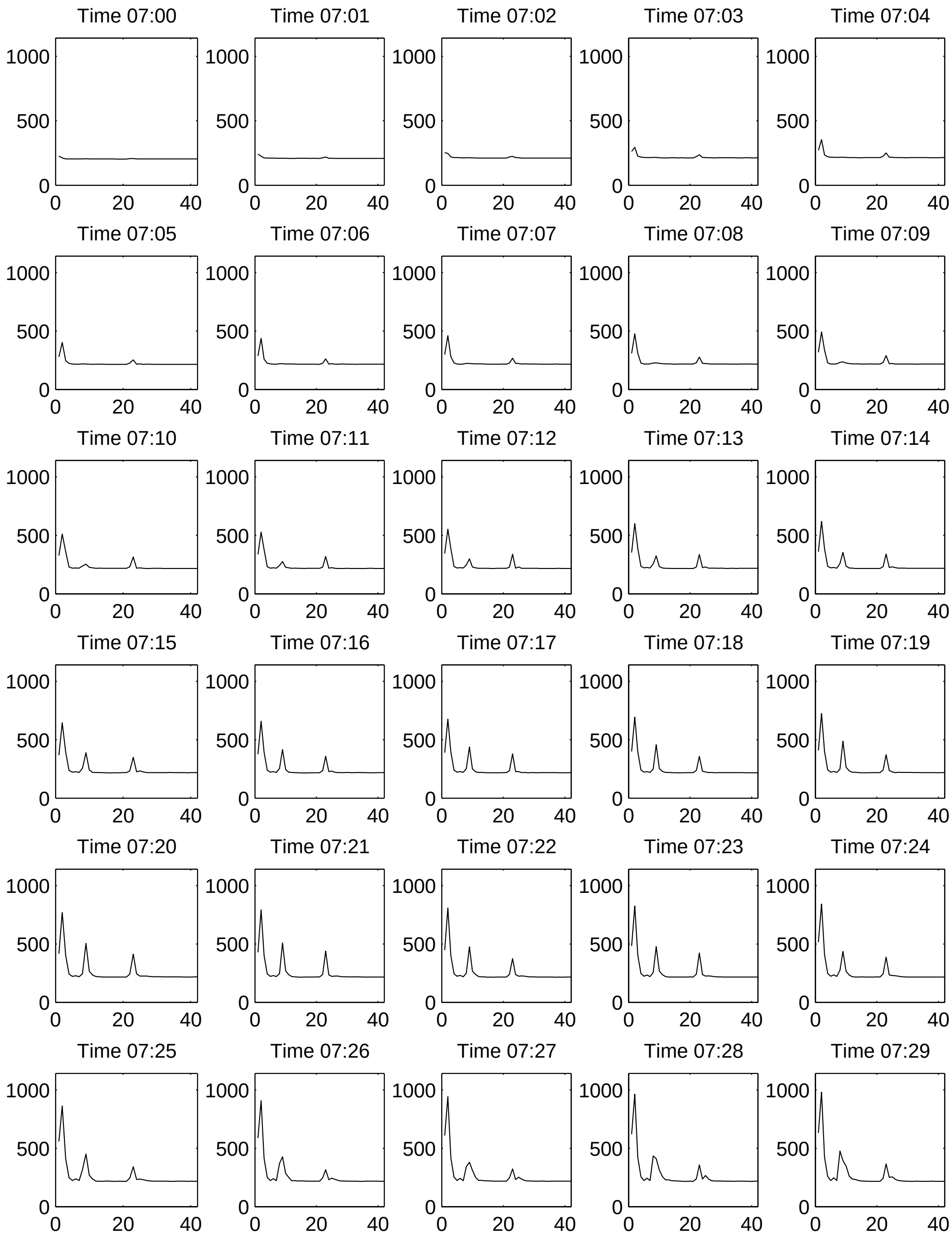


Figure 6-8: Convergence of Online Path Model (X-Axis: Number of Iterations; Y-Axis: Expected OD Time (sec); $p = 0.9$)

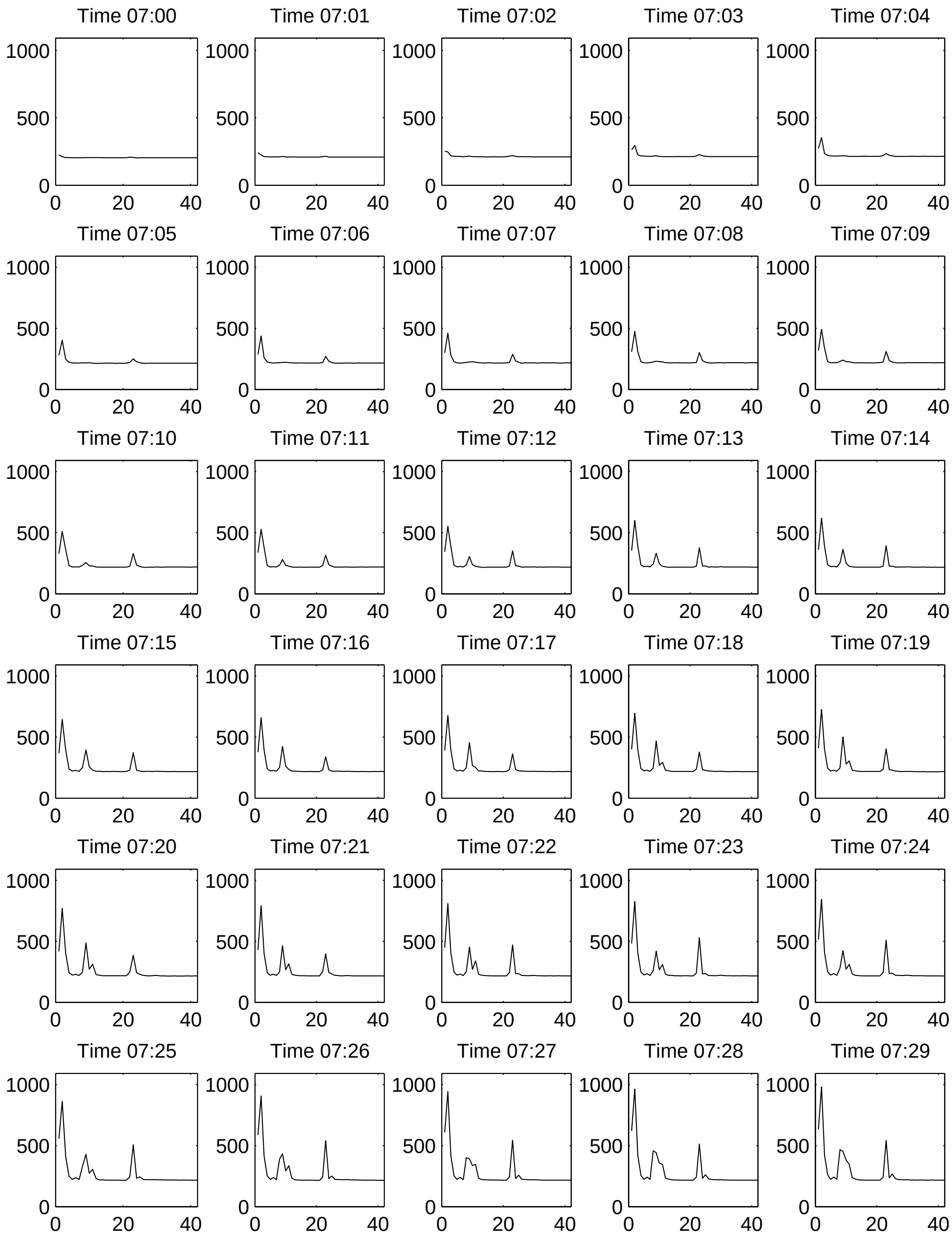


Figure 6-9: Convergence of Policy Model (X-Axis: Number of Iterations; Y-Axis: Expected OD Time (sec); $p = 0.9$)

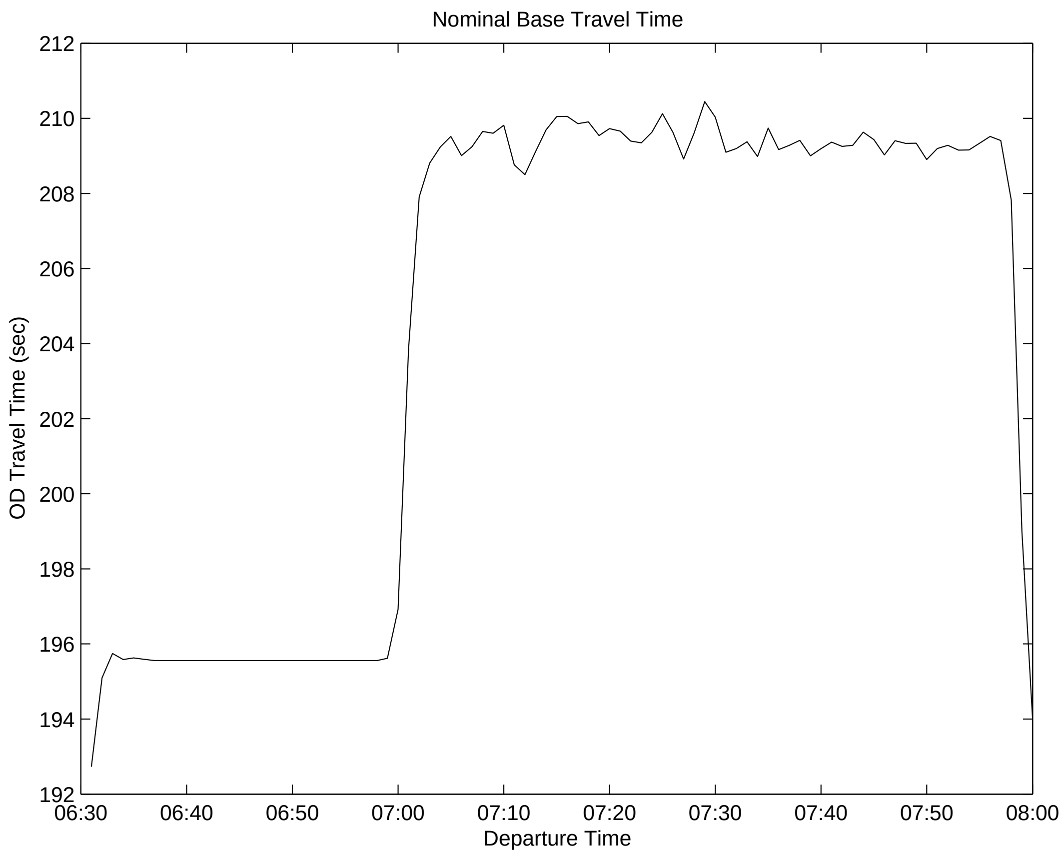


Figure 6-10: Nominal OD Travel Times of Base Model ($p = 0.9$)

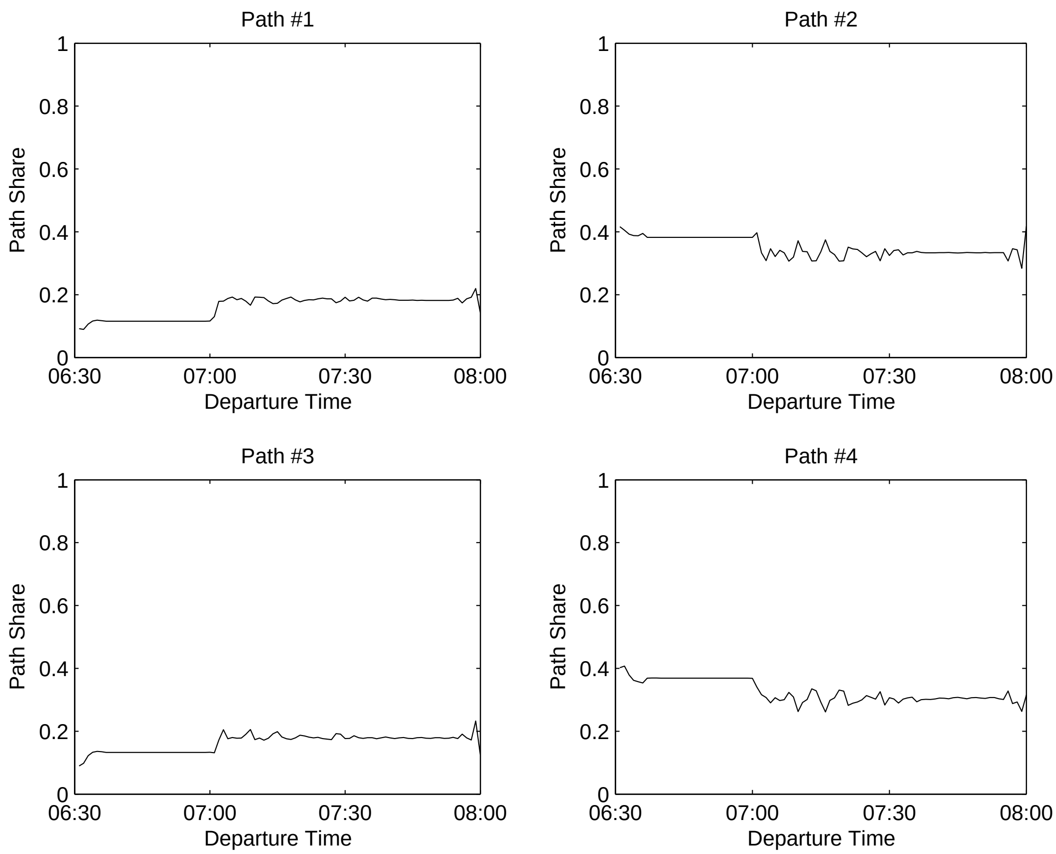


Figure 6-11: Equilibrium Path Flows of Base Model($p = 0.9$)

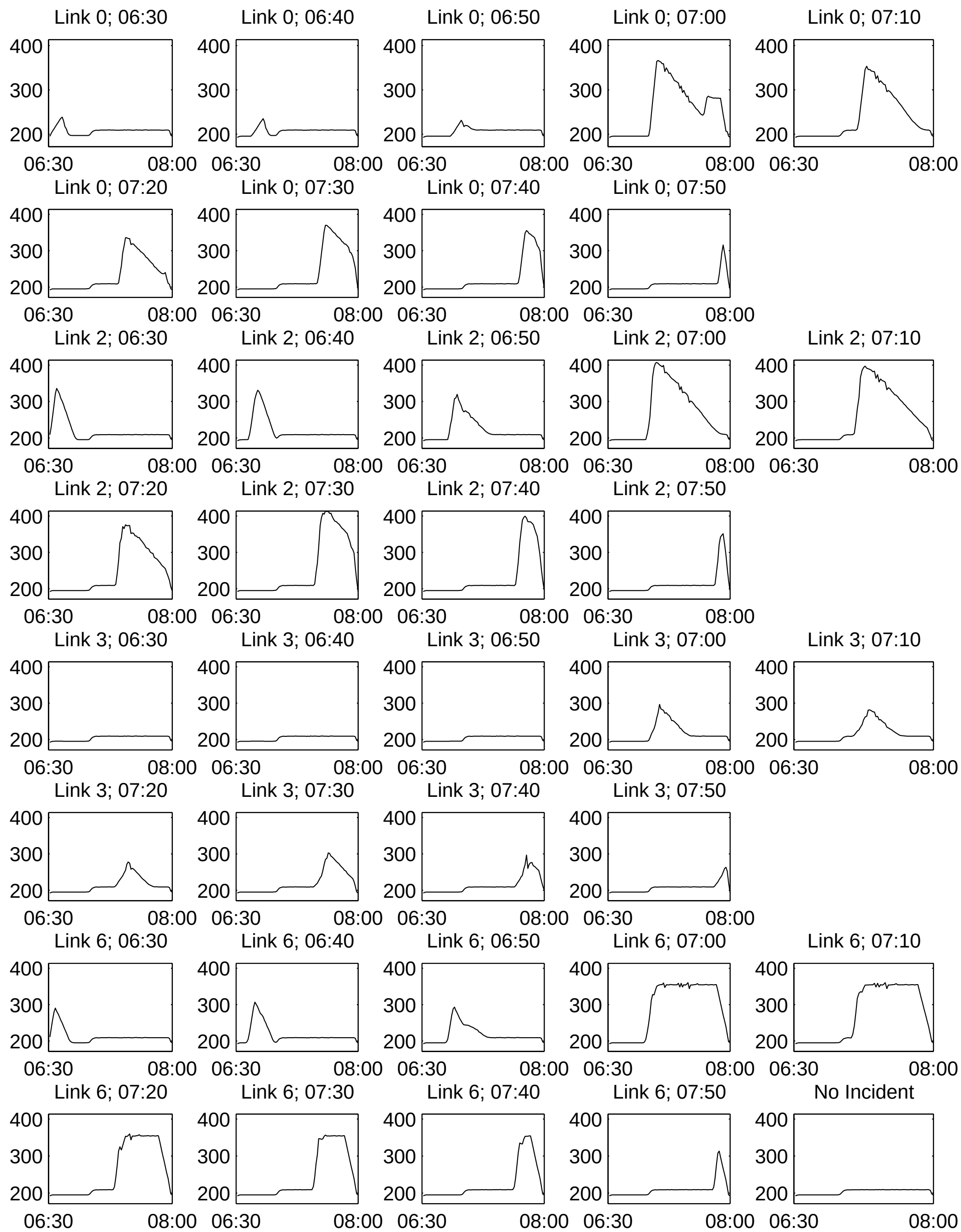


Figure 6-12: OD Travel Time Distribution of Base Model(X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); $p = 0.9$)

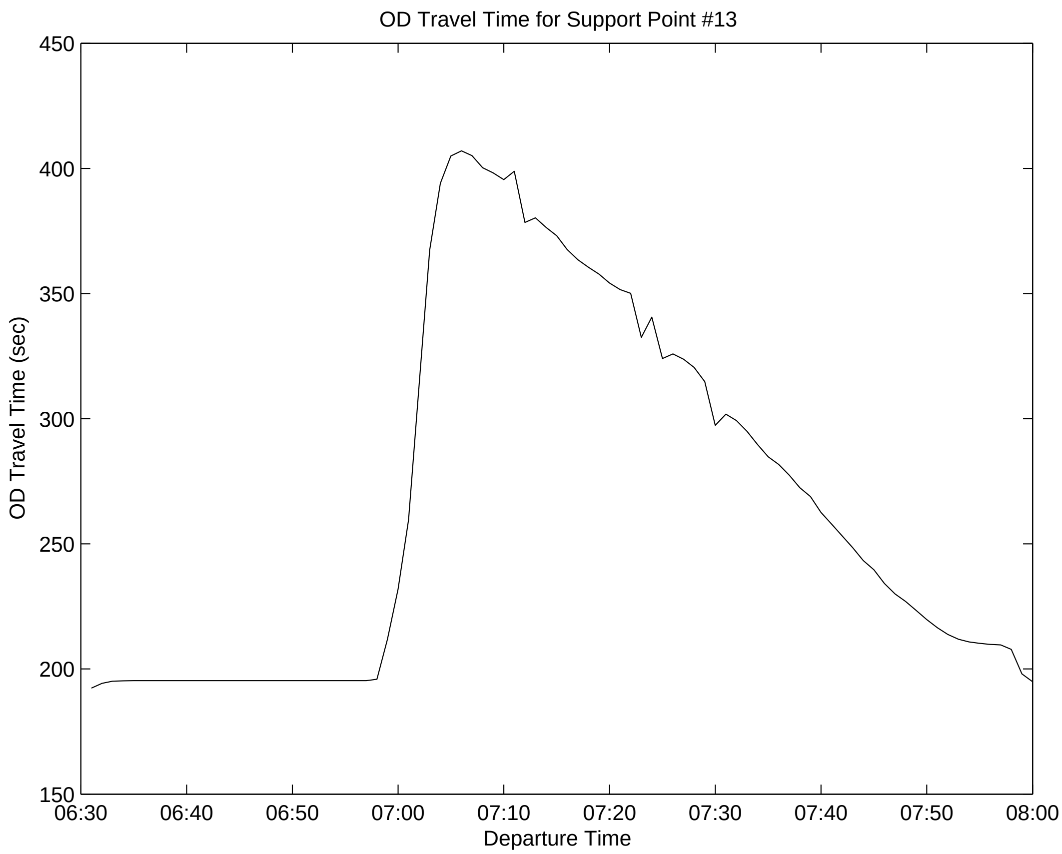


Figure 6-13: OD Travel Time in Support Point 13 of Base Model ($p = 0.9$)

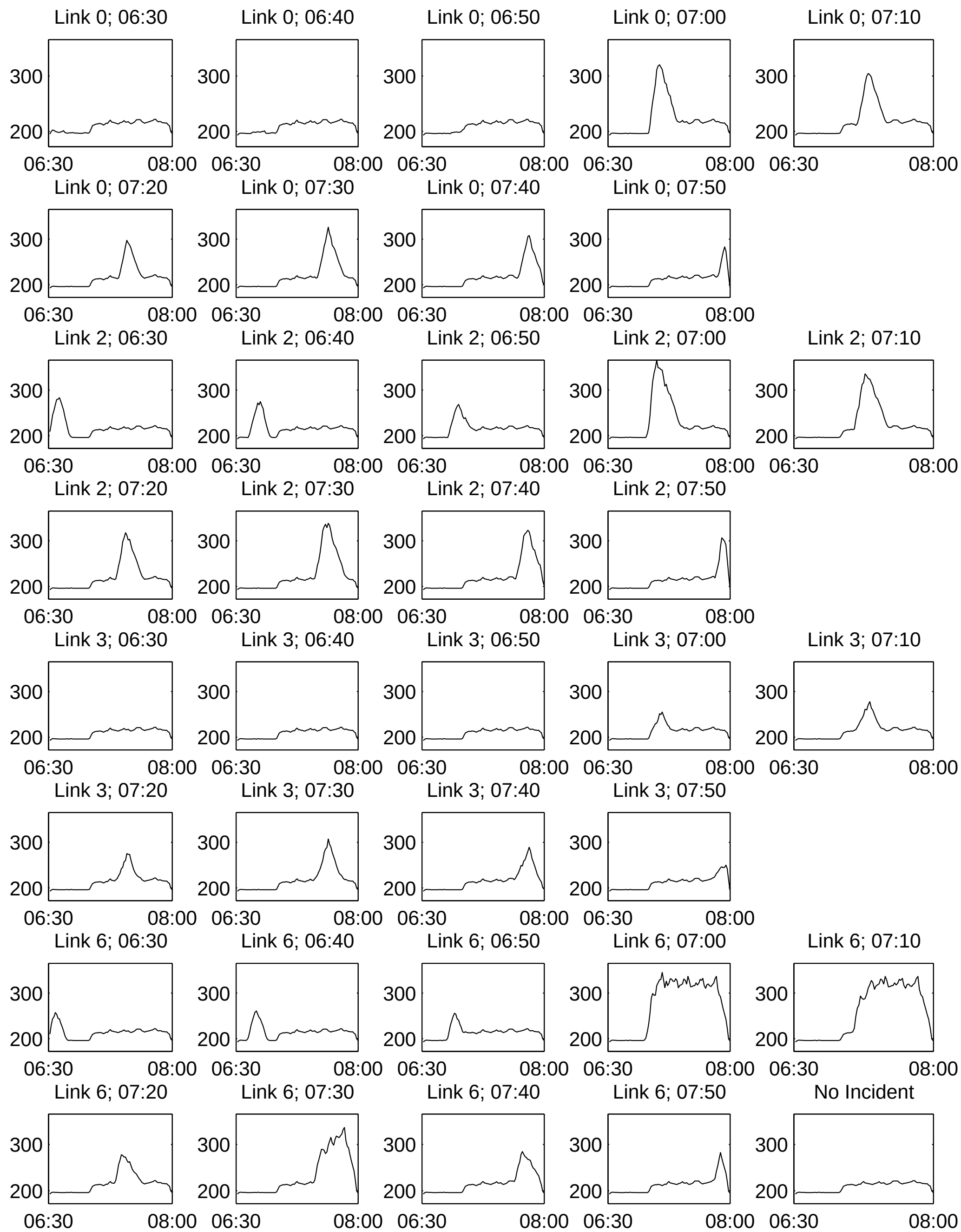


Figure 6-14: OD Travel Time Distribution of Path Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); $p = 0.9$)

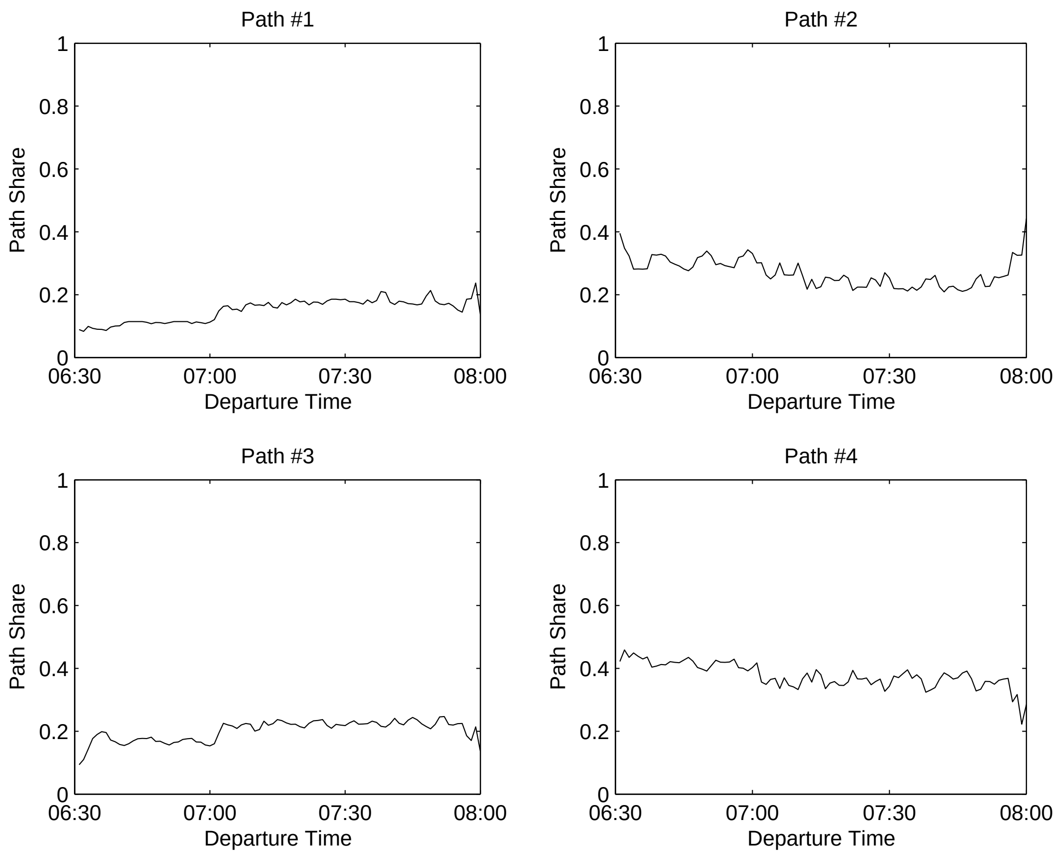


Figure 6-15: Path Flows of Path Model ($p = 0.9$)

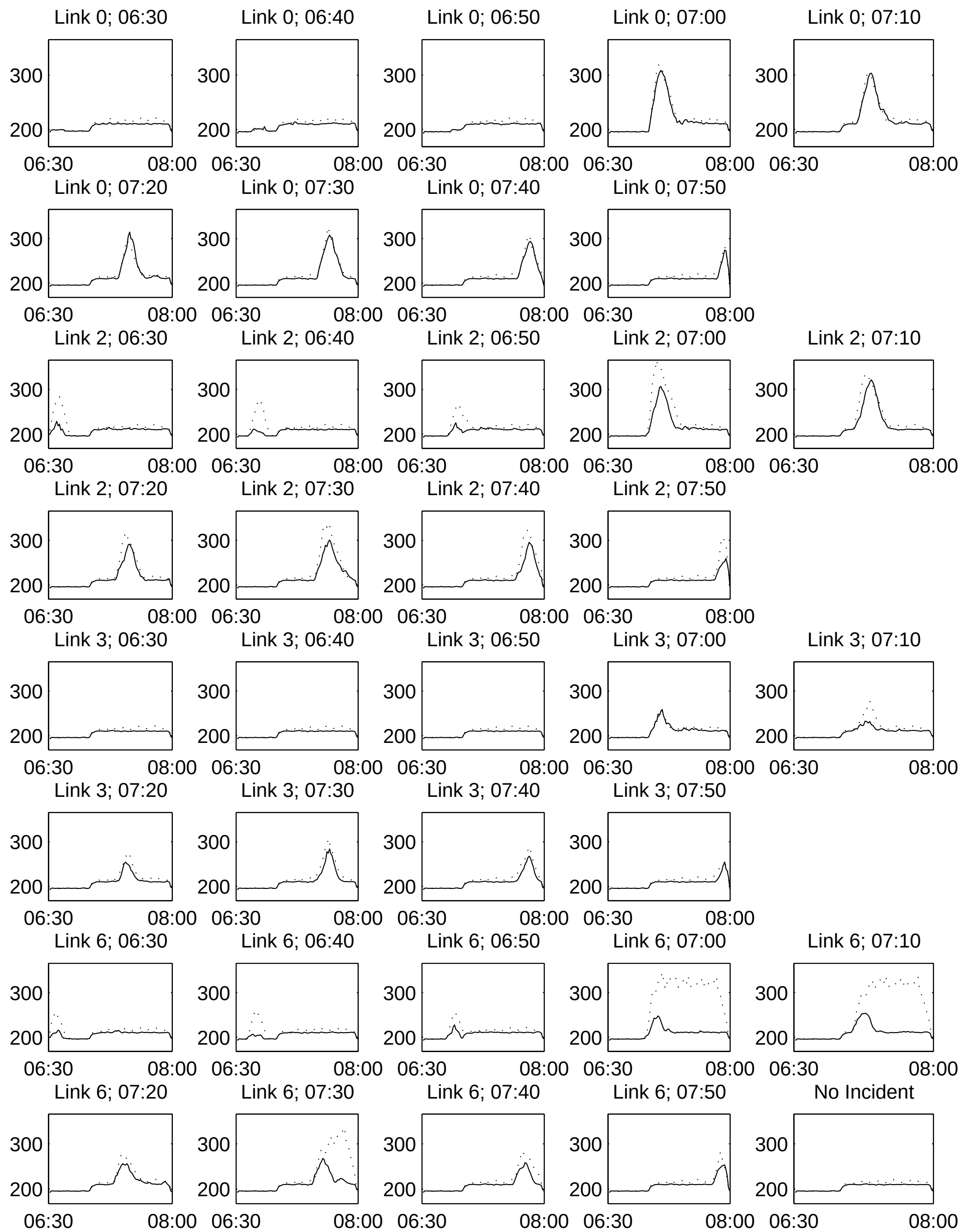


Figure 6-16: OD Travel Time Distribution of Online Path Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); $p = 0.9$)

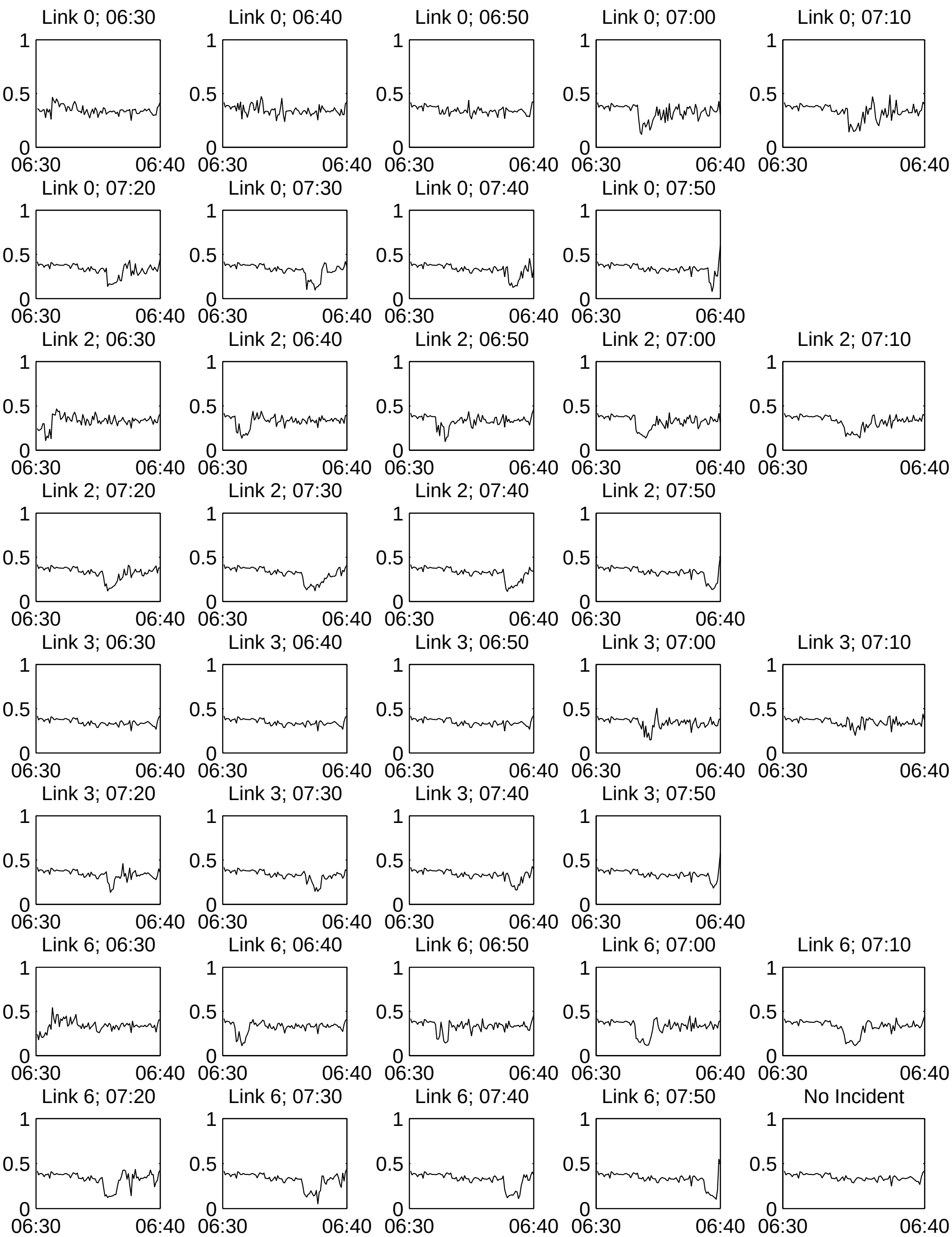


Figure 6-17: Path 2 Flow Distribution of Online Path Model(X-Axis: Departure Time; Y-Axis: Path Share; $p = 0.9$)

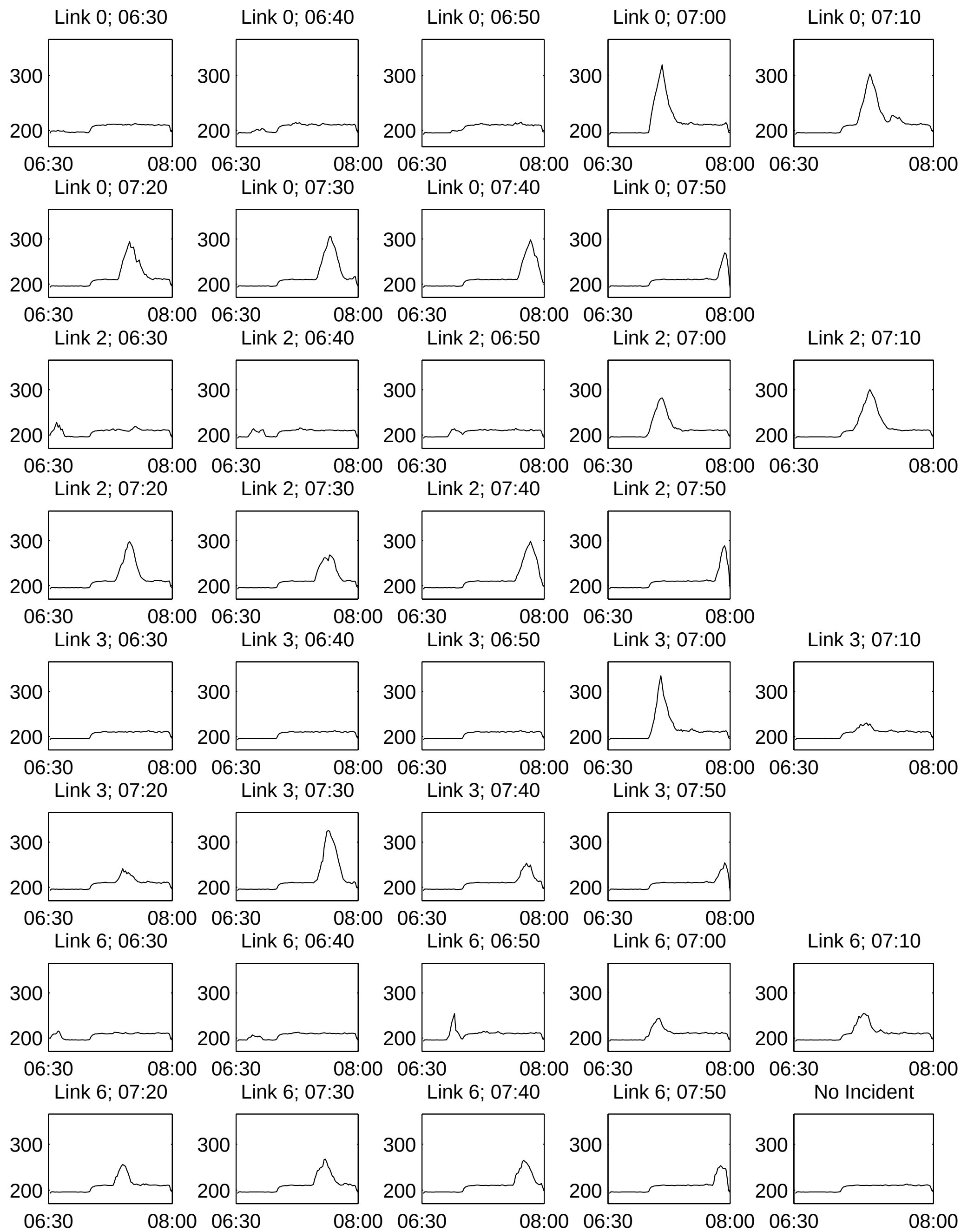


Figure 6-18: OD Travel Time Distribution of Policy Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); $p = 0.9$)

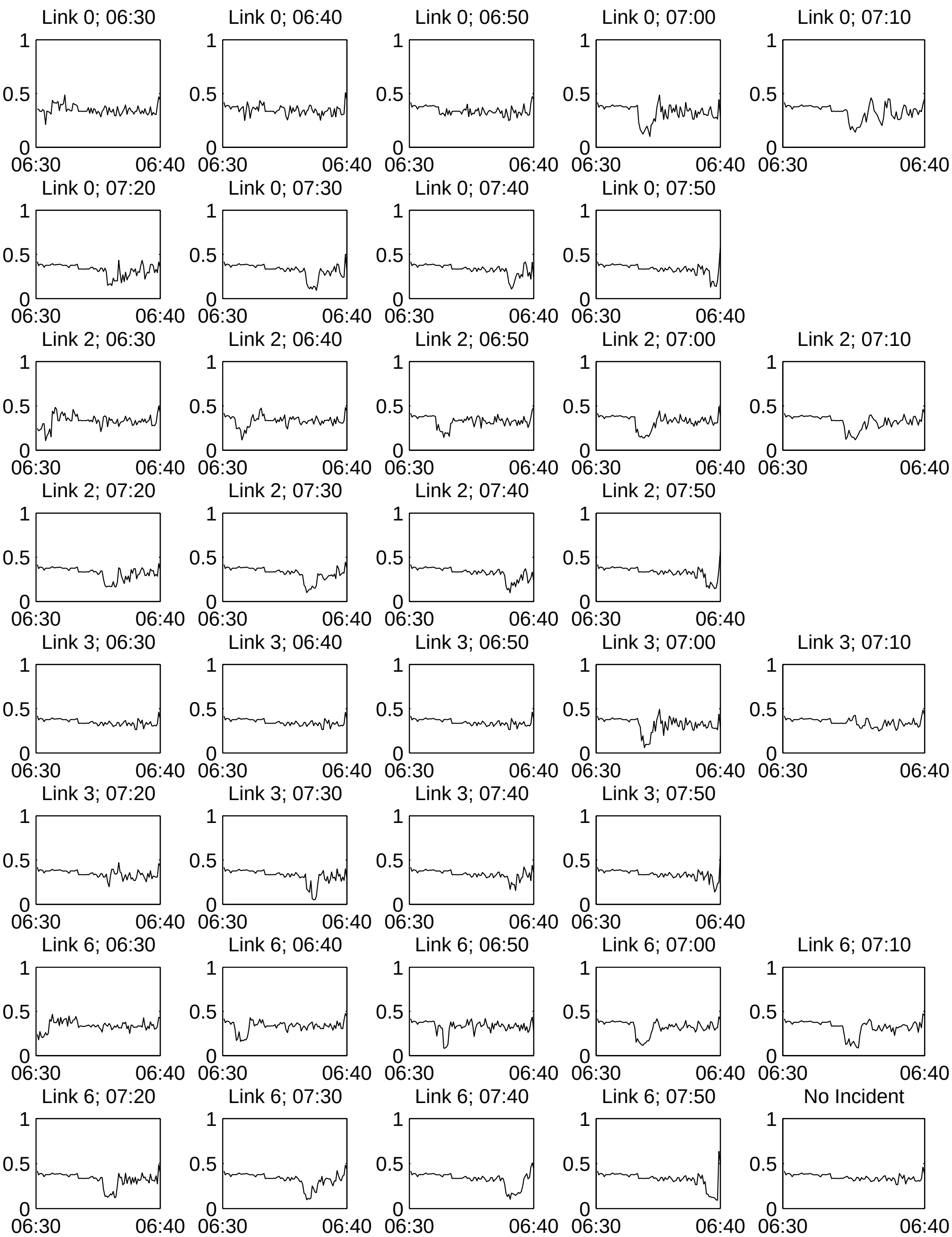


Figure 6-19: Path 2 Flow Distribution of Policy Model(X-Axis: Departure Time; Y-Axis: Path Share; $p = 0.9$)

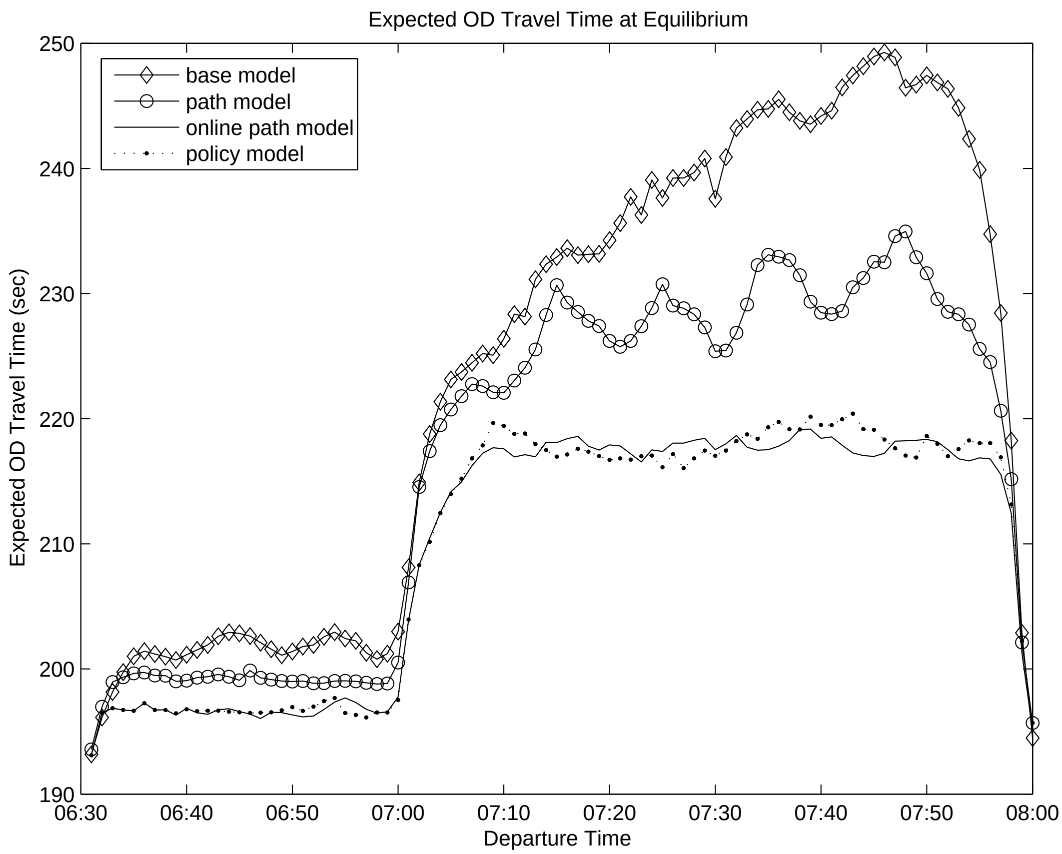


Figure 6-20: Expected OD Travel Time at Equilibrium of All Four Models ($p = 0.9$)

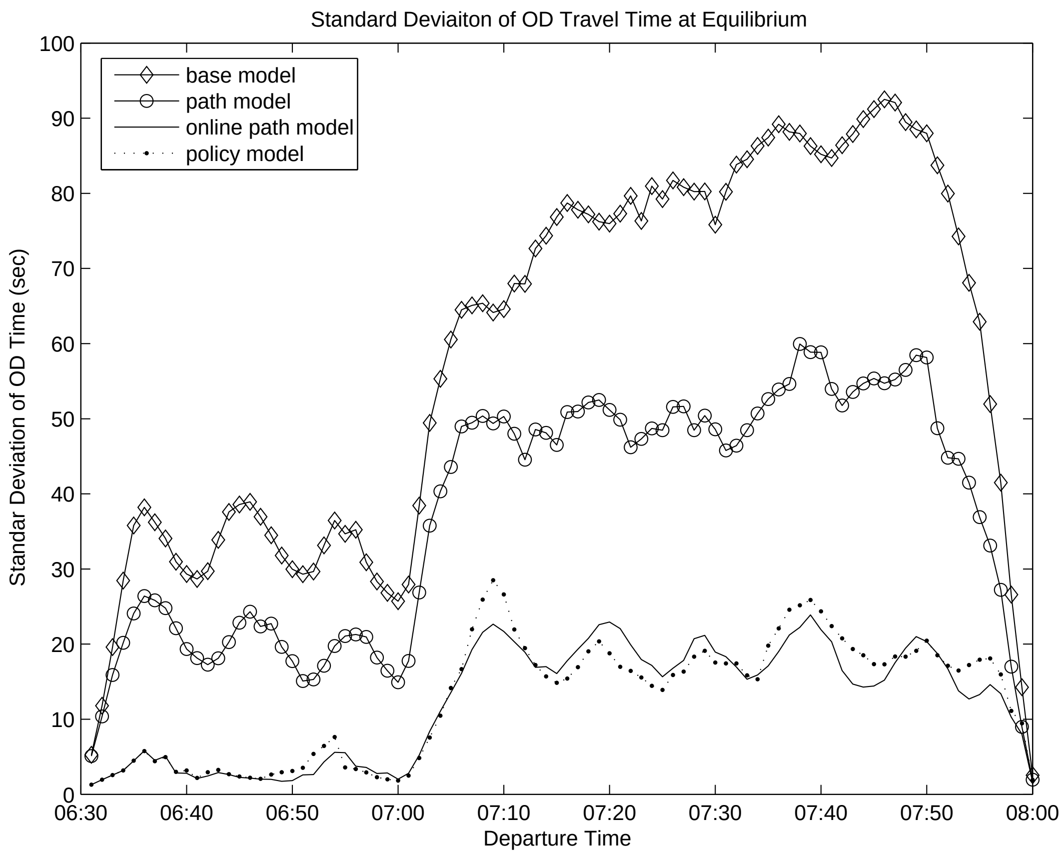


Figure 6-21: Standard Deviation of OD Travel Time at Equilibrium of All Four Models ($p = 0.9$)

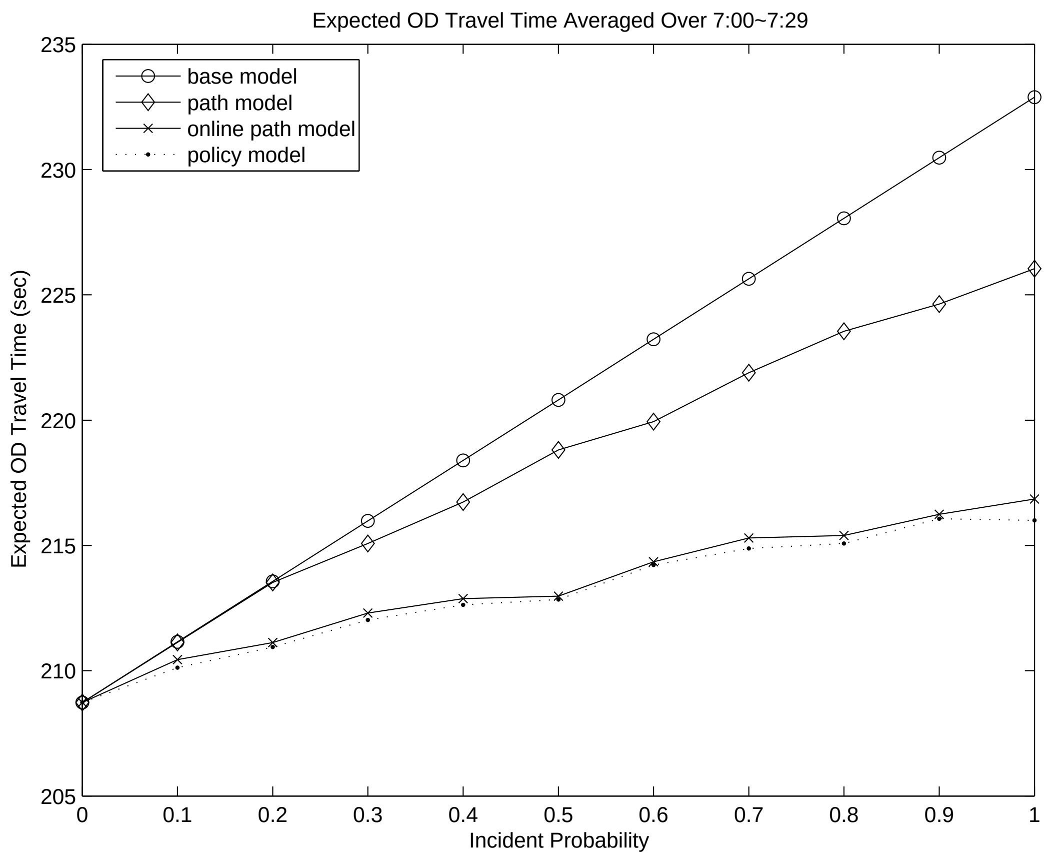


Figure 6-22: Average Expected OD Travel Times as Functions of Incident Probability

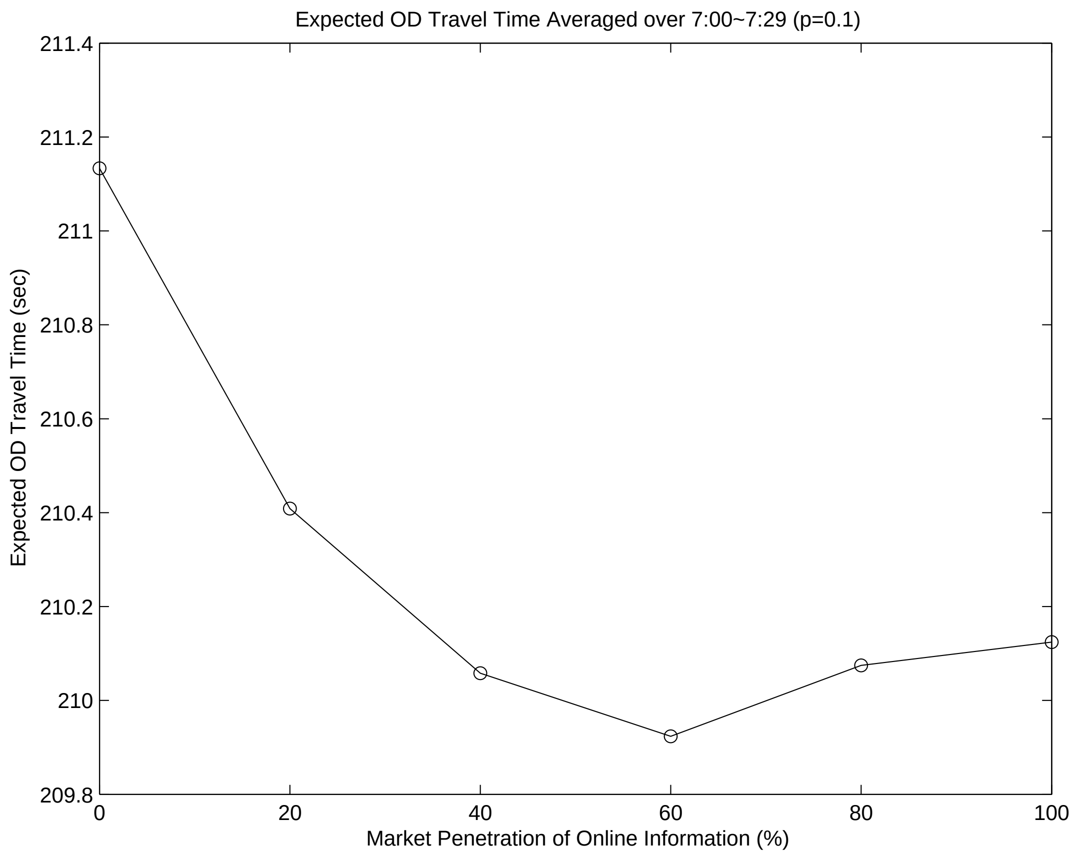


Figure 6-23: Average Expected OD Travel Times as Functions of Market Penetration ($p = 0.1$)

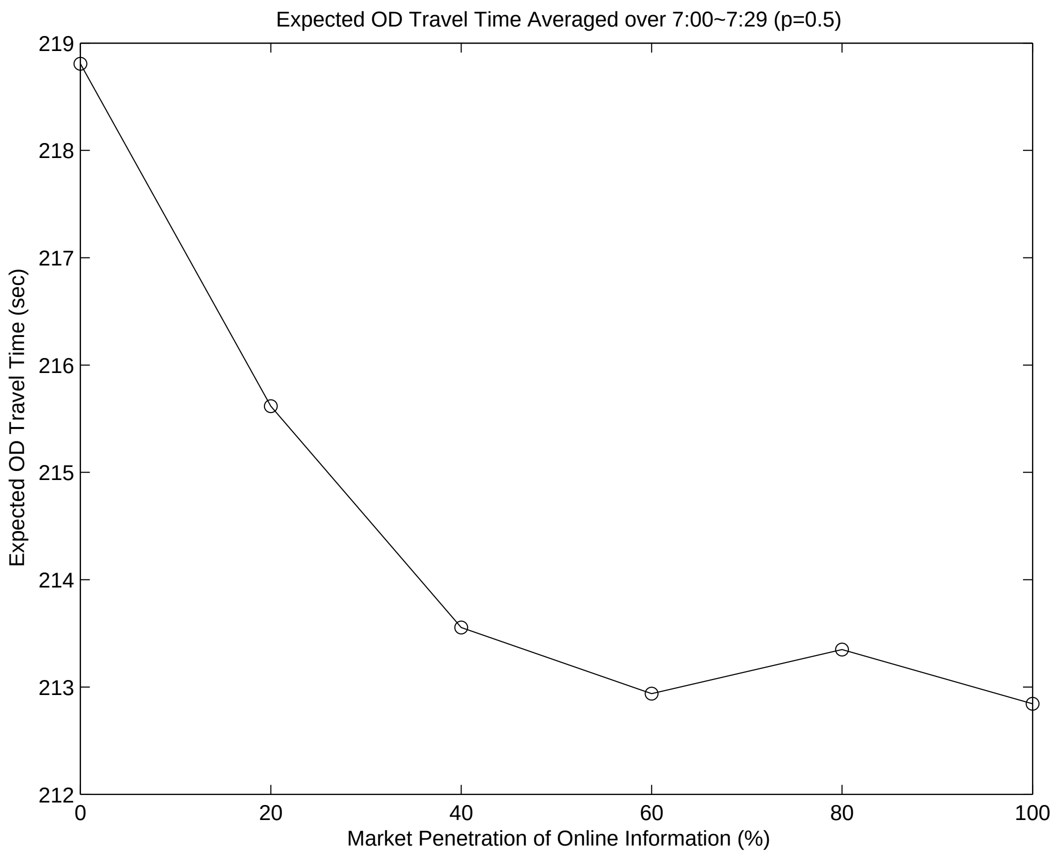


Figure 6-24: Average Expected OD Travel Times as Functions of Market Penetration ($p = 0.5$)

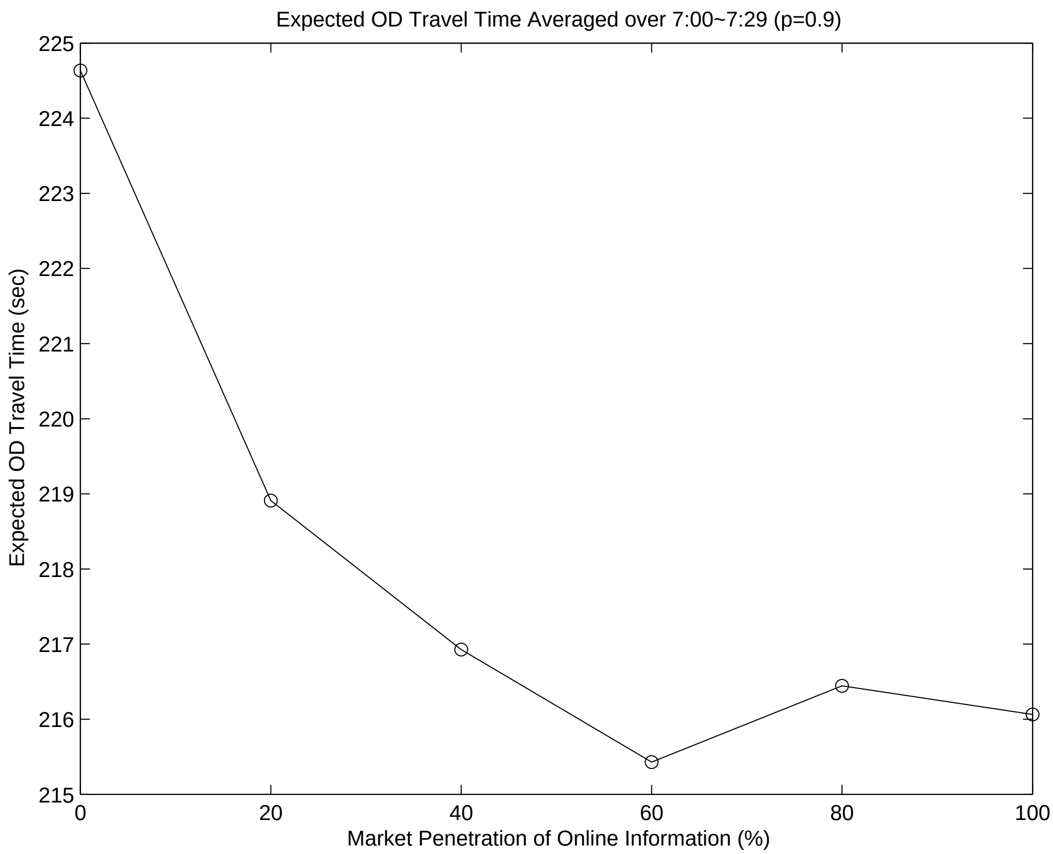


Figure 6-25: Average Expected OD Travel Times as Functions of Market Penetration ($p = 0.9$)

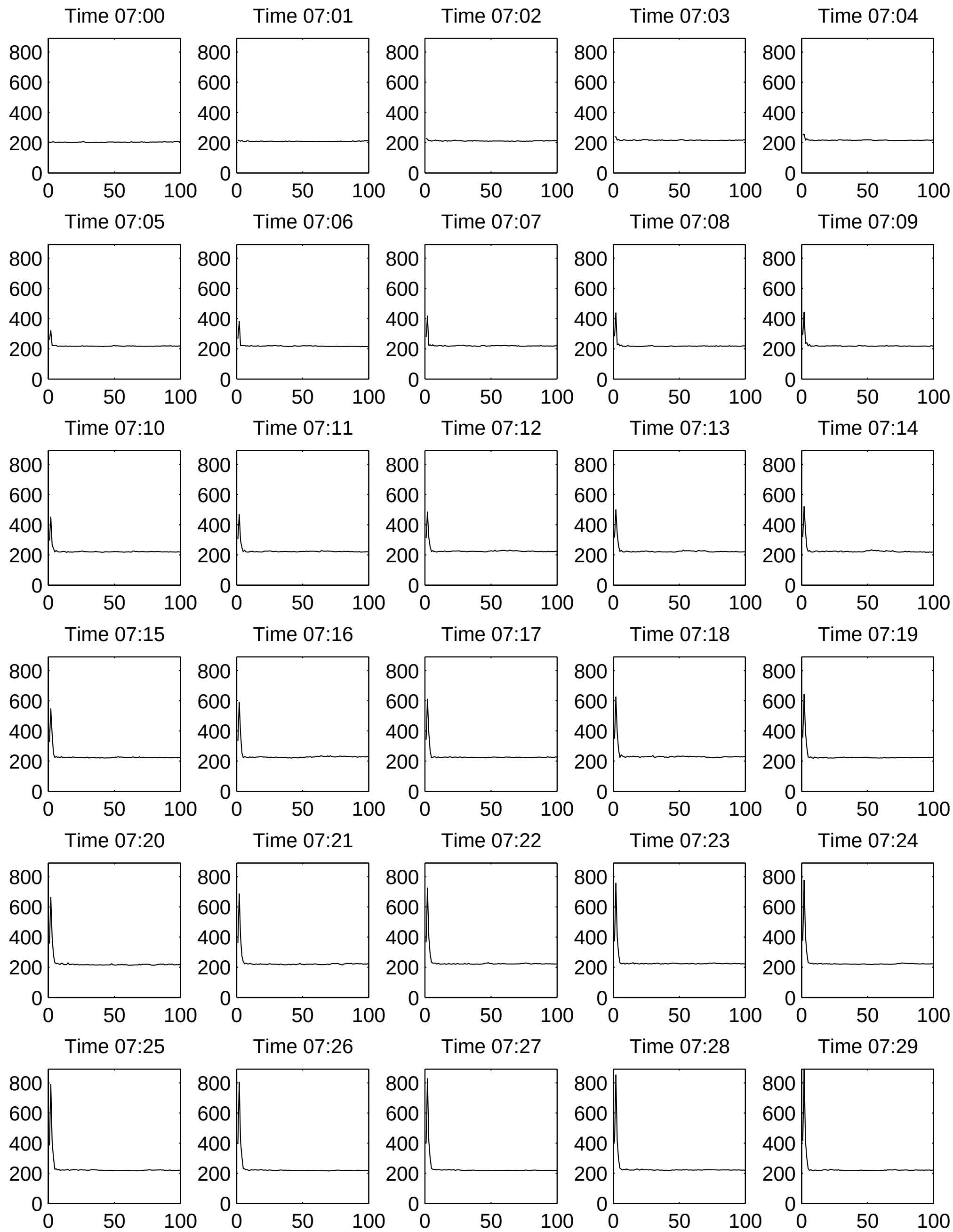


Figure 6-26: OD Travel Time Distribution of Policy Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); Random Demand)

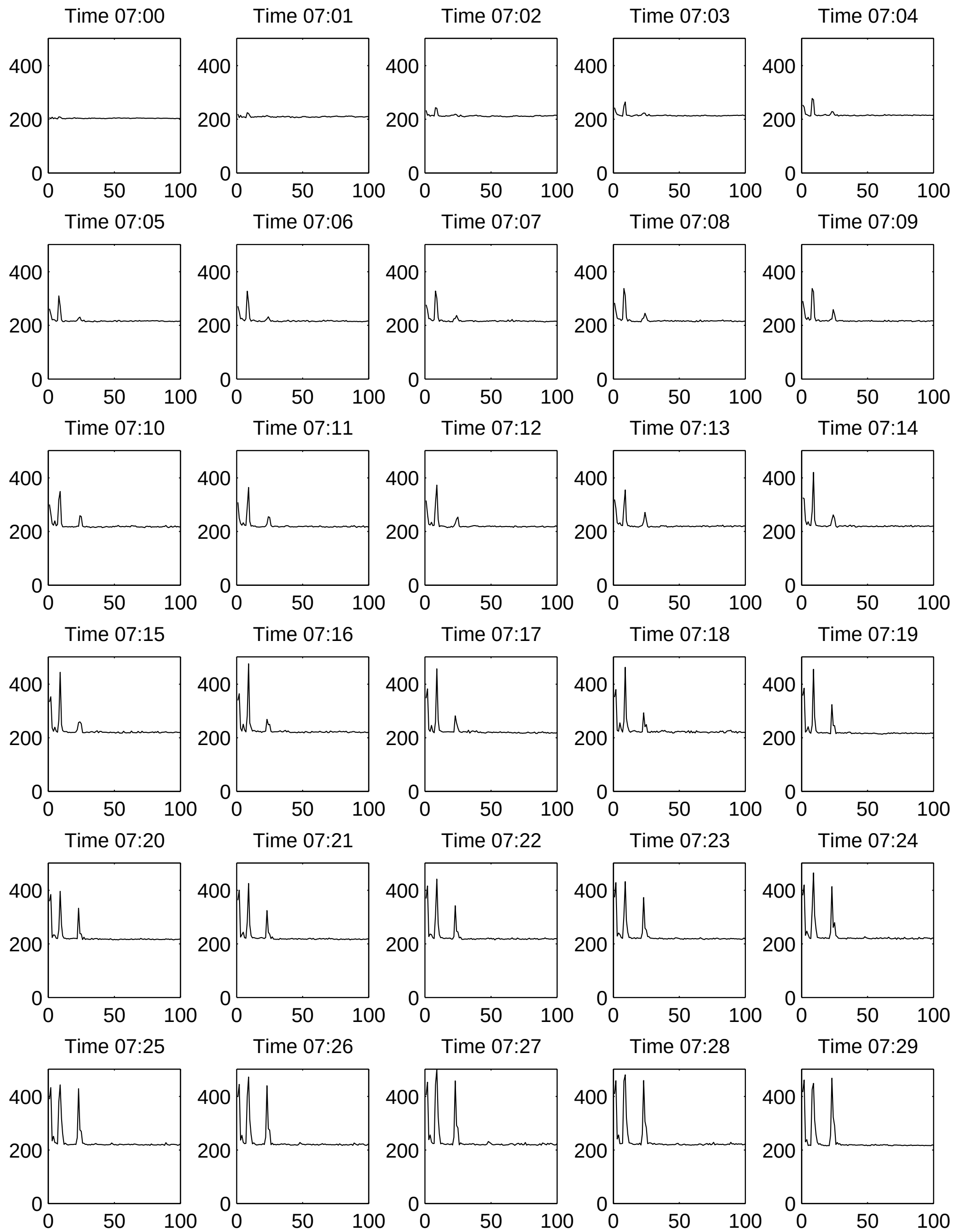


Figure 6-27: OD Travel Time Distribution of Online Path Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); Random Demand)

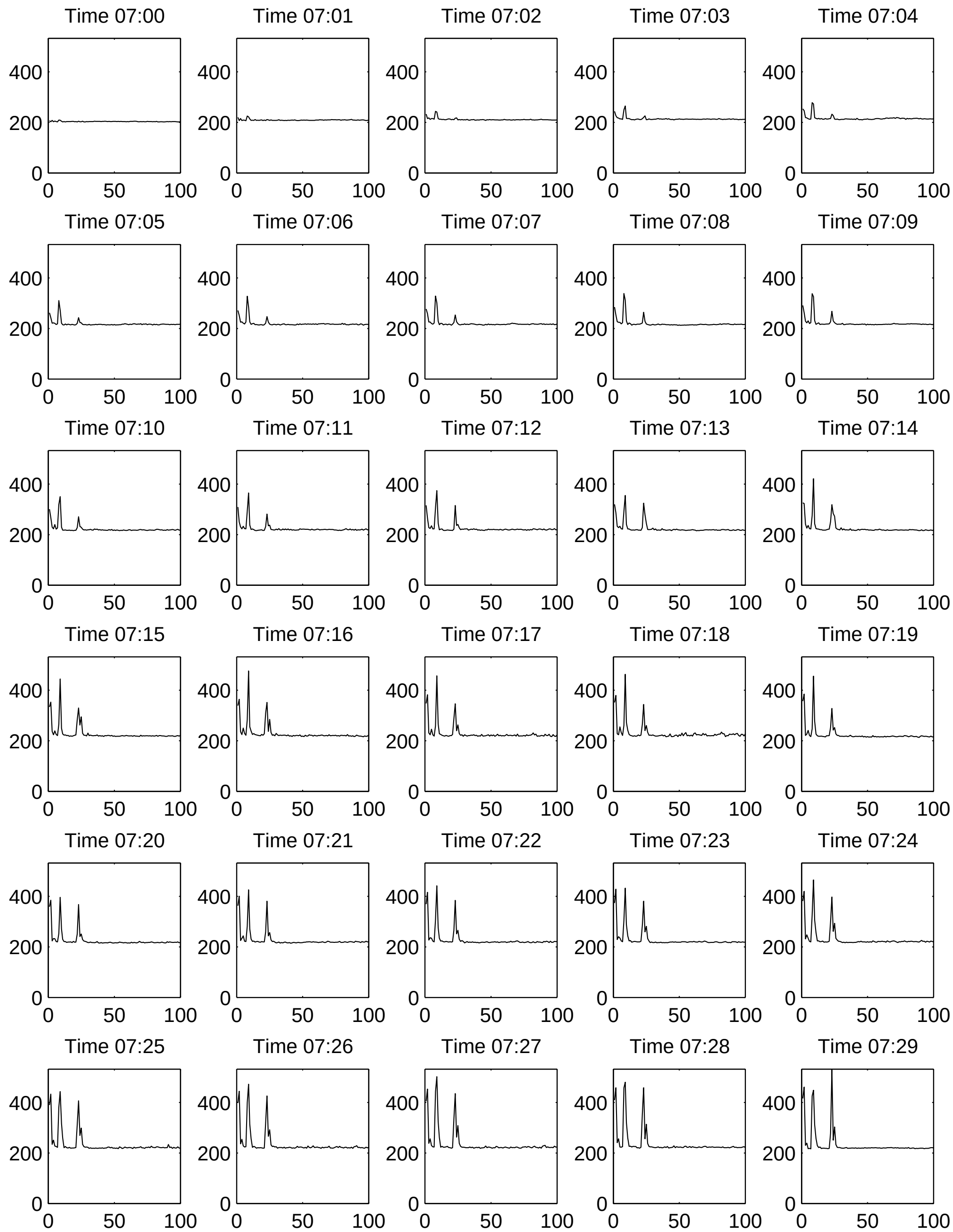


Figure 6-28: Convergence of Policy Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); Random Demand)

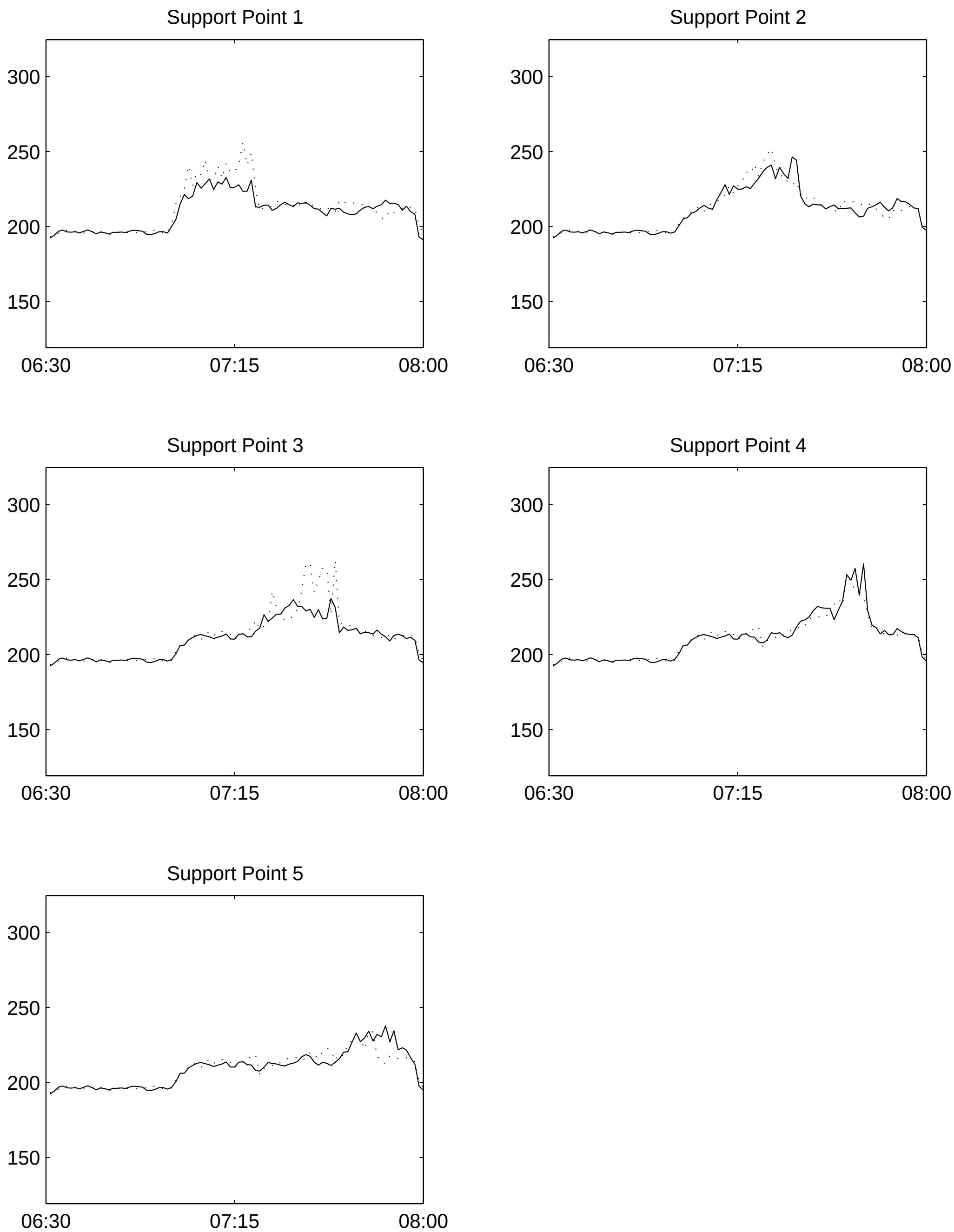


Figure 6-29: Convergence of Policy Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); Random Demand)

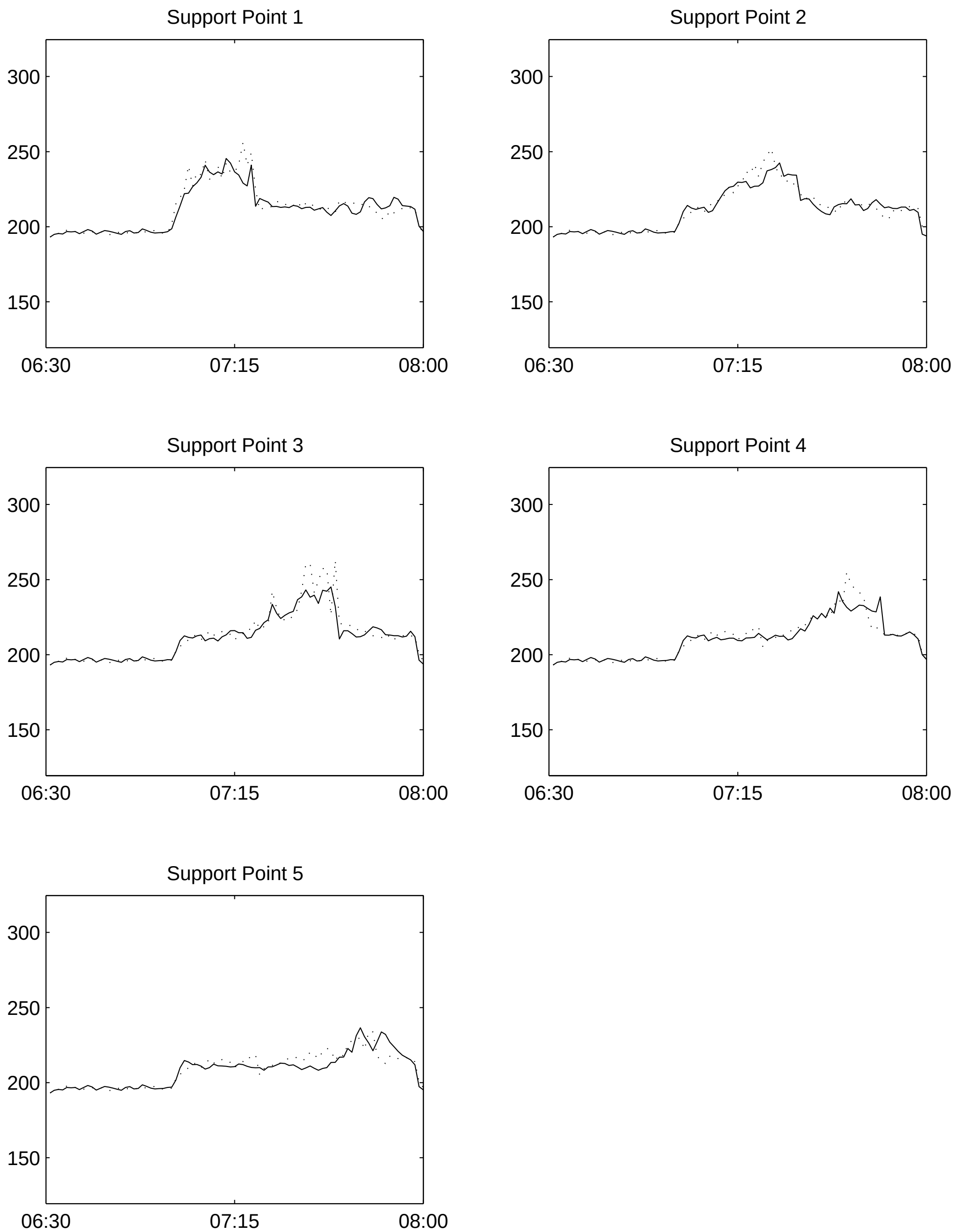


Figure 6-30: Convergence of Policy Model (X-Axis: Departure Time; Y-Axis: OD Travel Time (sec); Random Demand)

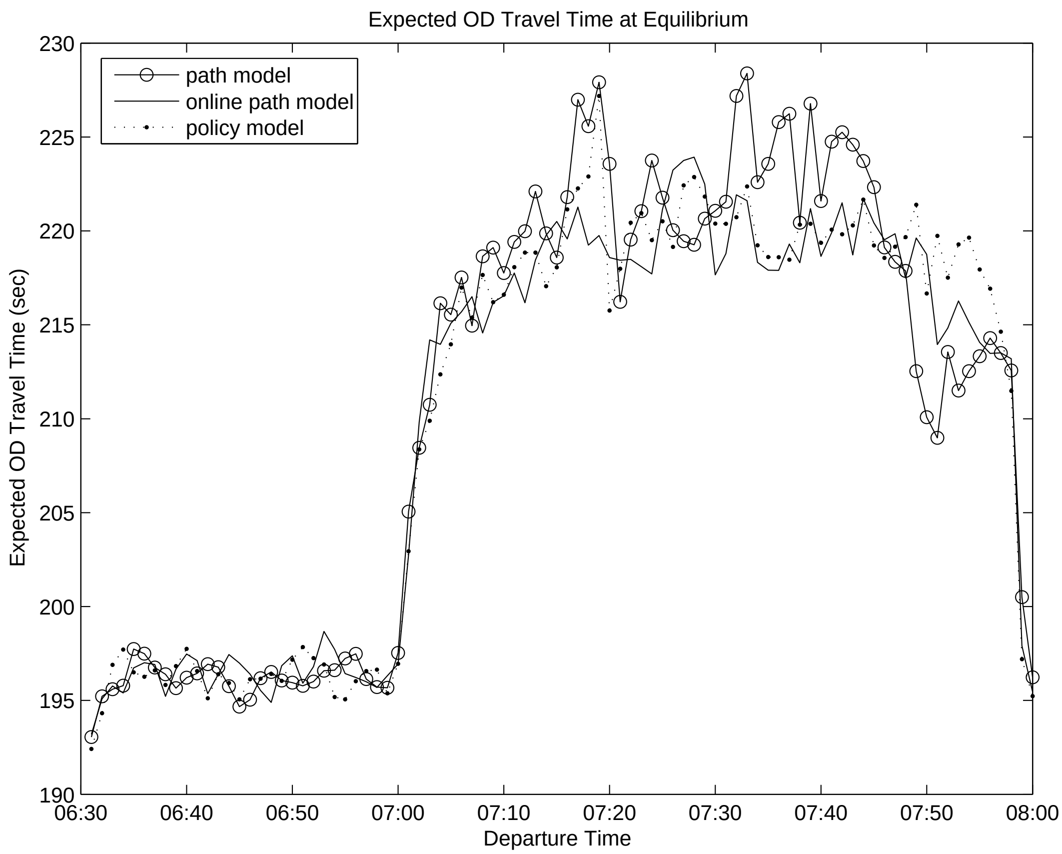


Figure 6-31: Expected OD Travel Time at Equilibrium of Three Models (Random Demand)

Chapter 7

Conclusions and Future Directions

7.1 Research Summary

In this chapter, a brief summary is presented on the thesis work and findings, and then some suggested future research directions. If only one key word is allowed for this thesis, it would be "adaptive". What is the optimal way of being adaptive in a stochastic network? What traffic conditions will emerge if travelers are adaptive to network conditions? The first question leads to the research on optimal routing policy (ORP) problems and algorithm design in stochastic time-dependent (STD) networks, and the second question leads to the research on a policy-based stochastic dynamic traffic assignment (DTA) model assuming that users take routing policies, where the first part is a base for the second part.

There are many variants of the ORP problem in STD networks, however they can be integrated in a framework. This thesis established such a framework, including a general description of the STD network, the decision process, the problem statement, and the optimality conditions. A comprehensive taxonomy of the ORP problem is provided, based on network link-wise and time-wise stochastic dependency and information access. These two factors determine the current information based on which routing decisions are made. Numerous variants exist according to the taxonomy, and this thesis provided insights into most of them, focusing on the specification of current information. This thesis then studied in detail a variant (termed the POI variant) which is particularly relevant in traffic networks. The POI variant takes into account the stochastic dependency among link travel times and the role of online information in routing decision making, which is a realistic depiction of traffic systems equipped with ATIS. An exact algorithm (Algorithm DOT-SPI) was designed and implemented for this variant.

The complexity analysis of Algorithm DOT-SPI revealed the need to design good approximations for the ORP problem. Four approximations were presented. Their properties were studied both theoretically and computationally. There is a trade-off between effectiveness and efficiency for all approximations, i.e. they could have satisfactory running times, but their results could be arbitrarily worse in absolute value than those obtained by running the exact algorithm. The computational tests studied the relationship between some parameters and the performance of approximations.

The criteria of optimality are extended from expected travel time to travel time variance and expected early/late schedule delays, which are three other important factors for travelers' routing choices in a stochastic network. Generic optimality conditions are presented, and algorithms are developed. It is found that Bellman's principle of optimality does not hold for the minimum variance policy problem, and thus no exact optimality conditions are available.

This thesis established the first policy-based dynamic traffic assignment model in a stochastic time-dependent network. The distinctive feature that differentiates the proposed model from a conventional DTA model is the ability to model travelers' adaptive routing choices. This DTA model takes as input a discrete distribution of demand/supply, and outputs equilibrium routing policy splits and an equilibrium discrete distribution of link travel times. The random supply includes incidents, work zones, breakdowns and bad weather, for example, and can be expressed through random reduction of link capacities.

The policy-based DTA model is composed of three interacting components: an optimal routing policy generation algorithm, a users' routing policy choice model, and a policy-based dynamic loading model, which are studied respectively in the thesis. The assignment problem is formulated as a fixed point problem and an MSA heuristic is proposed to solve it. Computational tests are carried out in a hypothetical network, where random incidents are the source of stochasticity. Four different DTA models in stochastic networks are studied, which differ in the information available and routing choices assumed for the users. The MSA heuristic with some modifications worked reasonably with the test network under the proposed test settings. As there is no proof of convergence for the heuristic, caution should be taken when generalizing the findings. It is found that the adaptiveness leads to travel time savings at equilibrium. As a byproduct, travel time reliability is also enhanced, as the travel time peak caused by a random incident was cut by intelligent diversion. Sensitivity analyses show that the value of online information is an increasing function of incident probability for the tested demand scale and network topology. It is also shown that online information penetration played an important role in travel time savings. The savings are high when market penetration is low. The relationship between travel time savings and market penetration is not monotonic, although any positive penetration leads to time savings compared to the no-online-information scenario. This suggests that when implementing a travelers' information system or route guidance system, one needs to choose carefully the penetration of information to maximize benefits.

7.2 Future Research Directions

The research of optimal routing policy problems in stochastic time-dependent networks and the policy-based stochastic dynamic traffic assignment problem is far from being mature. Yet it is of more and more interest, as it is very timely and relevant with the advent of advanced traveler information systems (ATIS). In the following paragraphs, future research directions are discussed.

Modeling Issues Traffic routing and assignment problems under the presence of ATIS or other information provision systems are much more complicated than conventional problems in static or dynamic deterministic networks with simplified assumptions of no information or full information. This thesis has managed to handle a variety of modeling issues arising in such a context, but there are still more issues for future research.

First of all, more realistic specifications of the current information I are needed to model all kinds of ATIS. In this thesis, the complete dependency perfect online information (POI) variant is studied in detail, depicting a traffic system with a very advanced ATIS that covers all links in the network at any time and with travelers equipped with in-vehicle communication systems that can retrieve and process the large amount of information available. However, as indicated in the taxonomy, there are many other variants that are relevant in a traffic network. The online information might be limited in the coverage of links and time periods for many existing and future ATIS. For example, the VMS location problem is of immediate interest for many transportation agencies. VMS technology is quite mature and thus enables many agencies to actually install VMS, but careful consideration should be paid in locating them, as the cost is still very high. The coverage of VMS is very limited both spatially and temporally. It is desired to have in-depth study of an ORP variant with a proper current information defined for VMS. Information provided by an VMS is about links downstream the VMS location and is only available when a traveler is passing the VMS location. Therefore, at any time and node, the online information could contain realized travel times on links that were reported by VMSs along the route the traveler has traveled so far. If the traveler has not passed any VMS, his/her online information is empty. This routing algorithm then can be used in a policy-based stochastic DTA model, and an evaluation of a specific VMS location design is available.

Another consideration is to include predictions about future travel times in the current information. Sometimes an ATIS provides predictions to travelers other than realized network conditions. This thesis focuses on online information about realized link travel times, however, there is no theoretical limitation on applying the same method to cases where predictions are included in the online information. As the current information is mathematically travel times of a set of links at a set of time periods, it could contain time periods beyond the current time. As an extreme example, the full information variant discussed in Section 2.2.2 has a current information that contain all links at all time periods, i.e. the future is known for sure.

Travelers could also differ in the *a priori* distribution of link travel times, and specifically a traveler's *a priori* distribution of link travel times could be different from the equilibrium distribution of all link travel times. If the difference is just in the number of random variables included (remember that each link travel time at each time period is a random variable), such as in the case that travelers have little knowledge about untraveled routes, the same model can be applicable by setting the untraveled link travel times to free flow travel times in the *a priori* distribution. Sometimes due to other factors, such as perception errors, measurement errors and others, there is a difference between the distribution of a true link travel time and that perceived by a traveler and utilized in making routing decisions. One measure to handle this situation is to use a random utility model for routing policy choice, such that random errors are added to true policy attributes, such as expected travel time or variance. Actually a random utility choice model is already in the policy-based stochastic DTA model studied in this thesis. This treatment avoids direct manipulation of "perceived" link travel time distributions. If in some cases, such a direct manipulation is needed, research efforts in this direction should be made in the future.

A routing policy in this thesis is defined as a mapping from a state to next node. However, it is possible to map a state to a set of next nodes, each with a certain probability. Is this a better depiction of travelers' routing choices in reality? Would this way of routing lead to lower expected travel time than a routing policy defined in this thesis? How to design algorithms to compute an optimal routing policy defined this way? Little is known at the time being and extensive future research is desired.

The developed DTA model deals with a wide range of random sources, but it makes a "large sample" assumption that policy splits are equal to policy choice probabilities. It is interesting to make the model completely stochastic by relaxing this assumption. As shown in the literature review of Section 5.1, there have been a number of papers that abandon the "large sample" assumption. Their work is carried out in static network with non-adaptive routing choices, but could still be very helpful in extending the work in this thesis.

Mathematical Issues This thesis focuses on modeling, algorithm design and computational tests. A number of mathematical issues are encountered but are not dealt with. However to better understand the routing policy problems, mathematical analysis is desired.

The policy-based stochastic DTA model studied in this thesis is formulated as a fixed point problem and solved by an MSA heuristic. However, no theoretical study on the existence, uniqueness, stability or other properties of a policy-based stochastic dynamic user equilibrium or the convergence of MSA is conducted. Instead, this thesis focuses on the development of a solution heuristic that works reasonably to provide the first insight into the newly established model. However, in order to make the study systematic and mature, mathematical analysis is indispensable. Earlier work on the mathematical analysis of deterministic dynamic user equilibrium, of stochastic static user equilibrium and of hyperpath-based static user equilibrium could be helpful. It is generally understood that there is a trade-off between the realistic assumptions of a model and the tractability of its mathematical properties, therefore it is conjectured that the mathematical study requires restricting assumptions, especially on link performance functions.

The other mathematical issue is the relationship between the hyperpath approach which is popular in transit network research and the routing policy approach in this thesis. As discussed in the literature review of DTA models in Section 5.1, the two approaches have quite a few similarities. Compared to the lack of research evidence in routing policy based traffic assignment in general road networks, hyperpath based transit assignment problems have been studied for over a decade. Therefore it is very interesting to compare and relate the two approaches, and potentially the hyperpath approach will help the development of the routing policy approach.

Algorithmic Issues The routing policy based algorithms and models are mostly likely to be applied in real time, and thus efficiency is an important issue. The efficiency can be obtained from two ways: algorithmic design and computer implementation. The algorithmic issues are discussed here. In Chapter 3, four approximation methods are designed and tested. In fact, there are many other ways of designing approximations. One possible approximation is the aggregate states approximation. It is also referred to as feature extraction in dynamic programming. In this thesis the network conditions are at a very disaggregate level, i.e. each possible link travel time realization can possibly change the state. Sometimes, however, aggregate states could be used to reduce the dimension of the state space while still give a satisfactory representation of the network conditions. One possible aggregate state in traffic applications is the level of service, A, B, C, D, E, or F.

Another issue is on the study and algorithm design of a chronological policy-based dynamic network loading (DNL) model. In this thesis, the iterative implementation of the policy-based DNL is used, where an existing non-adaptive loader can be used as a black box. The advantage of the iterative approach is that one can make use of the existing non-adaptive DNL conveniently. The disadvantage is the overhead in data exchange, since the non-adaptive loader is used as a black box and the only way for exchanging data is through files. It is then desirable to implement the chronological approach and study its performance against the iterative approach.

Implementation Issues The efficiency of the models also depends on how the algorithms are implemented in computers. The implementation of the MSA heuristic for the policy-based stochastic DTA model in this thesis is not optimized, as the focus of the computational tests is more on the convergence of the MSA method and differences between various models, given that the research is still at the very early stage for a policy-based DTA model. The efficiency of an algorithm, however, is very important if it is to be applied to real-world problems. High-performance computing implementations, such as parallel computing, should be considered in the future.

Computational Tests Extensions The computational tests on policy-based stochastic DTA model in this thesis provide valuable insights into the problem for the first time. However, since the research is still in a very early stage, many interesting tests are not performed and are desired for future research. Specifically, tests on a real-world network with actual data are of great interest, as they can provide researchers with an idea how the model and algorithms will perform in a realistic network.

In all of the computational tests, there is no sign of significant difference between the online path model and the policy model. This is largely because of the small size of the test network and thus not enough diversion chances. As shown in an illustrative example in Section 6.1, the difference does exist with carefully chosen network data even in a very small network. Would this difference be significant in reality? More tests on larger networks and complicated information accessibility situations are desired. Note that an efficient implementation of the solution heuristic is a pre-requisite for extended tests.

Reliability is another important criterion besides expected travel time, when travelers make routing decisions in a stochastic network. This thesis provides studies on reliability of a routing policy and develops an algorithm that optimizes a linear combination of expected travel time, expected early schedule delay and expected late schedule delay of a routing policy, in Chapter 4, but no computer implementations are realized. It would be interesting to implement it and study its performance computationally. The reliability issue then can be addressed in the policy-based stochastic DTA model by changing the functional form of the utility function to include reliability related attributes, and an optimal routing policy with minimum linear combination of attributes could be generated and used for updating the choice set.

Appendix A

An Illustrative Example for Algorithm DOT-SPI

We use an example to illustrate how Algorithm DOT-SPI works. The small network in Figure A-1 has three nodes, three links and the number of time periods is 3. The values of the travel time realizations are in Table A.1. Each of the eight realizations has a probability of 0.125. The network is designed to be very small to make the understanding of the algorithm easier. Note that travelers starting from node 2 or node 3 have no choice but to take node 3 as the next node. It is suggested that the reader pay attention to how routing decision at node 1 is affected by time and information.

Step 0: Construct $\mathbf{EV}(t), t = 0, ..., 2$

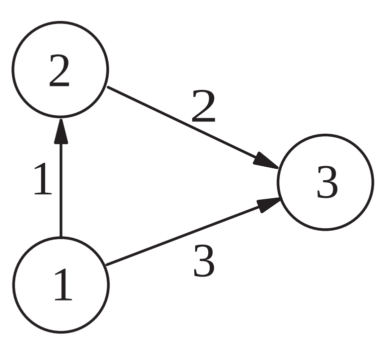


Figure A-1: Algorithm DOT-SPI: A Small Network

Time	Link	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
0	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1
	3	1	1	1	4	4	4	3	3
1	1	1	1	1	1	1	1	1	1
	2	2	2	1	2	2	1	2	1
	3	3	3	2	2	2	1	3	2
$t \geq 2$	1	1	1	2	1	1	1	2	2
	2	1	2	1	1	1	1	2	1
	3	3	2	2	3	4	3	5	2

Table A.1: Joint Realizations for the Small Network

Call Generate_Event_Collection

$D = \{\{v_1, ..., v_8\}\}$

$t = 0$

$(j, k) = 1$

$S = \{v_1, ..., v_8\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, ..., v_8\}\}$

$D = \{\{v_1, ..., v_8\}\}$

$(j, k) = 2$

$S = \{v_1, ..., v_8\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, ..., v_8\}\}$

$D = \{\{v_1, ..., v_8\}\}$

$(j, k) = 3$

$S = \{v_1, ..., v_8\}$

$w = 3$

$S'_1 = \{v_1, v_2, v_3\}, S'_2 = \{v_4, v_5, v_6\}, S'_3 = \{v_7, v_8\}$

$D' = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}\}$

$D = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}\}$

$\mathbf{EV}(0) = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}\}$

$t = 1$

$(j, k) = 1$

$S = \{v_1, v_2, v_3\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}\}$

$S = \{v_4, v_5, v_6\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}\}$

$S = \{v_7, v_8\}$

$w = 2$

$S'_1 = \{v_7\}, S'_2 = \{v_8\}$

$D' = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7\}, \{v_8\}\}$

$D = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7\}, \{v_8\}\}$

$(j, k) = 2$

$S = \{v_1, v_2, v_3\}$

$w = 2$

$S'_1 = \{v_1, v_2\}, S'_2 = \{v_3\}$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5, v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_4, v_5, v_6\}$

$w = 2$

$S'_1 = \{v_4, v_5\}, S'_2 = \{v_6\}$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_7\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_8\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$D = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$(j, k) = 3$

$S = \{v_1, v_2\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_3\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_4, v_5\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_6\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_7\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_8\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$D = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$\mathbf{EV}(1) = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$t = 2$

$(j, k) = 1$

$S = \{v_1, v_2\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_3\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_4, v_5\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_6\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_7\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_8\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$D = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$(j, k) = 2$

$S = \{v_1, v_2\}$

$w = 2$

$S_1' = \{v_1\}, S2' = \{v_2\}$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_3\}$

$w = 1$

$S_1' = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_4, v_5\}$

$w = 1$

$S_1' = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_6\}$

$w = 1$

$S_1' = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_7\}$

$w = 1$

$S_1' = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_8\}$

$w = 1$

$S_1' = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$D = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$(j, k) = 3$

$S = \{v_1\}$

$w = 1$

$S_1' = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_2\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_3\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_4, v_5\}$

$w = 2$

$S'_1 = \{v_4\}, S2' = \{v_5\}$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_6\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_7\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$S = \{v_8\}$

$w = 1$

$S'_1 = S$

$D' = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$D = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$\mathbf{EV}(2) = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

A summary of the results of constructing event collections is as follows.

$\mathbf{EV}(0) = \{\{v_1, v_2, v_3\}, \{v_4, v_5, v_6\}, \{v_7, v_8\}\}$

$\mathbf{EV}(1) = \{\{v_1, v_2\}, \{v_3\}, \{v_4, v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

$\mathbf{EV}(2) = \{\{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \{v_6\}, \{v_7\}, \{v_8\}\}$

Step 1: (Initialization)

1.1 Compute $e(j,2,EV), \forall j \in N, \forall EV \in \mathbf{EV}(2)$

This step involves solving deterministic static shortest path problems with each single joint realization $v_r, r = 1,..,8$. Any classical shortest path algorithm can be used. In our small network, this can be done by observation. The results are listed in Table A.2. In each result cell, the minimum expected travel time is given and the corresponding next node is in the parenthesis. We use "$n3$" to denote Node 3 and this rule of notation applies to all other nodes.

	$\{v_1\}$	$\{v_2\}$	$\{v_3\}$	$\{v_4\}$	$\{v_5\}$	$\{v_6\}$	$\{v_7\}$	$\{v_8\}$
$t=3$	0 (n3)	0 (n3)	0 (n3)	0 (n3)	0 (n3)	0 (n3)	0 (n3)	0 (n3)
$t=2$	1 (n3)	2 (n3)	1 (n3)	1 (n3)	1 (n3)	1 (n3)	2 (n3)	1 (n3)
$t=1$	2 (n2)	2 (n3)	2 (n3)	2 (n2)	2 (n2)	2 (n2)	4 (n2)	2 (n3)

Table A.2: Results in the Static Deterministic Period

1.2 $e(j,t,EV) \leftarrow +\infty, \forall j \in N \backslash \{d\}$,

$e(d,t,EV) \leftarrow 0$,

$\forall t < K-1, \forall EV \in \mathbf{EV}(t)$

Step 2: (Main Loop)

$t = 1$

$EV = \{v_1, v_2\}$

$EV_1' = \{v_1\}, Pr(EV_1'|EV) = 0.5$

$EV_2' = \{v_2\}, Pr(EV_2'|EV) = 0.5$

$(j,k) = (1,2)$

$$temp = 1 + e(2, 1+1, EV_1')Pr(EV_1'|EV)$$
$$+e(2, 1+1, EV_2')Pr(EV_2'|EV)$$

$$= 1 + 1 \times 0.5 + 2 \times 0.5 = 2.5 < +\infty$$

$$e(1, 1, \{v_1, v_2\}) = 2.5, \mu^*(1, 1, \{v_1, v_2\}) = n2$$

$$(j, k) = (2, 3)$$

$$temp = 2 + e(3, 1 + 2, EV_1')Pr(EV_1'|EV)$$

$$+e(3, 1 + 2, EV_2')Pr(EV_2'|EV)$$

$$= 2 + 0 \times 0.5 + 0 \times 0.5 = 2 < +\infty$$

$$e(2, 1, \{v_1, v_2\}) = 2, \mu^*(2, 1, \{v_1, v_2\}) = n3$$

$$(j, k) = (1, 3)$$

$$temp = 3 + e(3, 1 + 3, EV_1')Pr(EV_1'|EV)$$

$$+e(3, 1 + 3, EV_2')Pr(EV_2'|EV)$$

$$= 3 + 0 \times 0.5 + 0 \times 0.5 = 3 > 2.5 = e(1, 1, \{v_1, v_2\})$$

$$EV = \{v_3\}$$

$$EV_1' = \{v_3\}, Pr(EV_1'|EV) = 1$$

$$(j, k) = (1, 2)$$

$$temp = 1 + e(2, 1 + 1, EV_1')Pr(EV_1'|EV)$$

$$= 1 + 1 \times 1 = 2 < +\infty$$

$$e(1, 1, \{v_3\}) = 2, \mu^*(1, 1, \{v_3\}) = n2$$

$$(j, k) = (2, 3)$$

$$temp = 1 + e(3, 1 + 1, EV_1')Pr(EV_1'|EV)$$

$$= 1 + 0 \times 1 = 1 < +\infty$$

$$e(2, 1, \{v_3\}) = 1, \mu^*(2, 1, \{v_3\}) = n3$$

$$(j, k) = (1, 3)$$

$$temp = 2 + e(3, 1 + 2, EV_1')Pr(EV_1'|EV)$$

$$= 2 + 0 \times 1 = 2 = e(1, 1, \{v_3\})$$

$$EV = \{v_4, v_5\}$$

$$EV_1' = \{v_4\}, Pr(EV_1'|EV) = 0.5$$

$$EV_2' = \{v_5\}, Pr(EV_2'|EV) = 0.5$$

$(j, k) = (1, 2)$

$$temp = 1 + e(2, 1 + 1, EV_1')Pr(EV_1'|EV)$$
$$+e(2, 1 + 1, EV_2')Pr(EV_2'|EV)$$
$$= 1 + 1 \times 0.5 + 1 \times 0.5 = 2 < +\infty$$
$$e(1, 1, \{v_4, v_5\}) = 2, \mu^*(1, 1, \{v_4, v_5\}) = n2$$

$(j, k) = (2, 3)$

$$temp = 2 + e(3, 1 + 2, EV_1')Pr(EV_1'|EV)$$
$$+e(3, 1 + 2, EV_2')Pr(EV_2'|EV)$$
$$= 2 + 0 \times 0.5 + 0 \times 0.5 = 2 < +\infty$$
$$e(2, 1, \{v_4, v_5\}) = 2, \mu^*(2, 1, \{v_4, v_5\}) = n3$$

$(j, k) = (1, 3)$

$$temp = 2 + e(3, 1 + 2, EV_1')Pr(EV_1'|EV)$$
$$+e(3, 1 + 2, EV_2')Pr(EV_2'|EV)$$
$$= 2 + 0 \times 0.5 + 0 \times 0.5 = 2 = e(1, 1, \{v_4, v_5\})$$

$EV = \{v_6\}$

$EV_1' = \{v_6\}, Pr(EV_1'|EV) = 1$

$(j, k) = (1, 2)$

$$temp = 1 + e(2, 1 + 1, EV_1')Pr(EV_1'|EV)$$
$$= 1 + 1 \times 1 = 2 < +\infty$$
$$e(1, 1, \{v_6\}) = 2, \mu^*(1, 1, \{v_6\}) = n2$$

$(j, k) = (2, 3)$

$$temp = 1 + e(3, 1 + 1, EV_1')Pr(EV_1'|EV)$$
$$= 1 + 0 \times 1 = 1 < +\infty$$
$$e(2, 1, \{v_6\}) = 1, \mu^*(2, 1, \{v_6\}) = n3$$

$(j, k) = (1, 3)$

$$temp = 1 + e(3, 1 + 1, EV_1')Pr(EV_1'|EV)$$
$$= 1 + 0 \times 1 = 1 < 2 = e(1, 1, \{v_6\})$$

$$e(1,1,\{v_6\}) = 1, \mu^*(1,1,\{v_6\}) = n3$$

$$EV = \{v_7\}$$

$$EV_1' = \{v_7\}, Pr(EV_1'|EV) = 1$$

$$(j,k) = (1,2)$$

$$temp = 1 + e(2, 1+1, EV_1')Pr(EV_1'|EV)$$

$$= 1 + 2 \times 1 = 3 < +\infty$$

$$e(1,1,\{v_7\}) = 3, \mu^*(1,1,\{v_7\}) = n2$$

$$(j,k) = (2,3)$$

$$temp = 2 + e(3, 1+2, EV_1')Pr(EV_1'|EV)$$

$$= 2 + 0 \times 1 = 2 < +\infty$$

$$e(2,1,\{v_7\}) = 2, \mu^*(2,1,\{v_7\}) = n3$$

$$(j,k) = (1,3)$$

$$temp = 3 + e(3, 1+3, EV_1')Pr(EV_1'|EV)$$

$$= 3 + 0 \times 1 = 3 = e(1,1,\{v_7\})$$

$$EV = \{v_8\}$$

$$EV_1' = \{v_8\}, Pr(EV_1'|EV) = 1$$

$$(j,k) = (1,2)$$

$$temp = 1 + e(2, 1+1, EV_1')Pr(EV_1'|EV)$$

$$= 1 + 1 \times 1 = 2 < +\infty$$

$$e(1,1,\{v_8\}) = 2, \mu^*(1,1,\{v_8\}) = n2$$

$$(j,k) = (2,3)$$

$$temp = 1 + e(3, 1+1, EV_1')Pr(EV_1'|EV)$$

$$= 1 + 0 \times 1 = 1 < +\infty$$

$$e(2,1,\{v_8\}) = 1, \mu^*(2,1,\{v_8\}) = n3$$

$$(j,k) = (1,3)$$

$$EV_1' = \{v_8\}, Pr(EV_1'|EV) = 1$$

$$temp = 2 + e(3, 1+2, EV_1')Pr(EV_1'|EV)$$
$$= 2 + 0 \times 1 = 2 = e(1, 1, \{v_8\})$$

The summary of results at time 1 is in Table A.3.

	$\{v_1, v_2\}$	$\{v_3\}$	$\{v_4, v_5\}$	$\{v_6\}$	$\{v_7\}$	$\{v_8\}$
$t = 3$	0(n3)	0(n3)	0(n3)	0(n3)	0(n3)	0(n3)
$t = 2$	2(n3)	1(n3)	2(n3)	1(n3)	2(n3)	1(n3)
$t = 1$	2.5(n2)	2(n2)	2(n2)	1(n3)	3(n2)	2(n2)

Table A.3: Results at Time 1

$t = 0$

$EV = \{v_1, v_2, v_3\}$

$EV_1' = \{v_1, v_2\}, Pr(EV_1'|EV) = 2/3$

$EV_2' = \{v_3\}, Pr(EV_2'|EV) = 1/3$

$(j, k) = (1, 2)$

$$temp = 1 + e(2, 0+1, EV_1')Pr(EV_1'|EV)$$
$$+ e(2, 0+1, EV_2')Pr(EV_2'|EV)$$
$$= 1 + 2 \times 2/3 + 1 \times 1/3 = 8/3 < +\infty$$

$e(1, 0, \{v_1, v_2, v_3\}) = 8/3, \mu^*(1, 0, \{v_1, v_2, v_3\}) = n2$

$(j, k) = (2, 3)$

$$temp = 1 + e(3, 0+1, EV_1')Pr(EV_1'|EV)$$
$$+ e(3, 0+1, EV_2')Pr(EV_2'|EV)$$
$$= 1 + 0 \times 2/3 + 0 \times 1/3 = 1 < +\infty$$

$e(2, 0, \{v_1, v_2, v_3\}) = 1, \mu^*(2, 0, \{v_1, v_2, v_3\}) = n3$

$(j, k) = (1, 3)$

$$temp = 1 + e(3, 0+1, EV_1')Pr(EV_1'|EV)$$
$$+ e(3, 0+1, EV_2')Pr(EV_2'|EV)$$
$$= 1 + 0 \times 2/3 + 0 \times 1/3 = 1 < 8/3 = e(1, 0, \{v_1, v_2, v_3\})$$

$$e(1, 0, \{v_1, v_2, v_3\}) = 1, \mu^*(1, 0, \{v_1, v_2, v_3\}) = n3$$

$$EV = \{v_4, v_5, v_6\}$$

$$EV'_1 = \{v_4, v_5\}, Pr(EV'_1|EV) = 2/3$$

$$EV'_2 = \{v_6\}, Pr(EV'_2|EV) = 1/3$$

$$(j, k) = (1, 2)$$

$$temp = 1 + e(2, 0 + 1, EV'_1)Pr(EV'_1|EV)$$

$$+e(2, 0 + 1, EV'_2)Pr(EV'_2|EV)$$

$$= 1 + 2 \times 2/3 + 1 \times 1/3 = 8/3 < +\infty$$

$$e(1, 0, \{v_4, v_5, v_6\}) = 8/3, \mu^*(1, 0, \{v_4, v_5, v_6\}) = n2$$

$$(j, k) = (2, 3)$$

$$temp = 1 + e(3, 0 + 1, EV'_1)Pr(EV'_1|EV)$$

$$+e(3, 0 + 1, EV'_2)Pr(EV'_2|EV)$$

$$= 1 + 0 \times 2/3 + 0 \times 1/3 = 1 < +\infty$$

$$e(2, 0, \{v_4, v_5, v_6\}) = 1, \mu^*(2, 0, \{v_4, v_5, v_6\}) = n3$$

$$(j, k) = (1, 3)$$

$$temp = 4 + e(3, 0 + 4, EV'_1)Pr(EV'_1|EV)$$

$$+e(3, 0 + 4, EV'_2)Pr(EV'_2|EV)$$

$$= 4 + 0 \times 2/3 + 0 \times 1/3 = 4 > 8/3 = e(1, 0, \{v_4, v_5, v_6\})$$

$$EV = \{v_7, v_8\}$$

$$EV'_1 = \{v_7\}, Pr(EV'_1|EV) = 0.5$$

$$EV'_2 = \{v_8\}, Pr(EV'_2|EV) = 0.5$$

$$(j, k) = (1, 2)$$

$$temp = 1 + e(2, 0 + 1, EV'_1)Pr(EV'_1|EV)$$

$$+e(2, 0 + 1, EV'_2)Pr(EV'_2|EV)$$

$$= 1 + 2 \times 0.5 + 1 \times 0.5 = 2.5 < +\infty$$

$$e(1, 0, \{v_7, v_8\}) = 2.5, \mu^*(1, 0, \{v_7, v_8\}) = n2$$

$$(j,k) = (2,3)$$

$$temp = 1 + e(3, 0+1, EV_1')Pr(EV_1'|EV)$$

$$+e(2, 0+1, EV_2')Pr(EV_2'|EV)$$

$$= 1 + 0 \times 0.5 + 0 \times 0.5 = 1 < +\infty$$

$$e(2, 0, \{v_7, v_8\}) = 1, \mu^*(2, 0, \{v_7, v_8\}) = n3$$

$$(j,k) = (1,3)$$

$$temp = 3 + e(3, 0+3, EV_1')Pr(EV_1'|EV)$$

$$+e(3, 0+3, EV_2')Pr(EV_2'|EV)$$

$$= 3 + 0 \times 0.5 + 0 \times 0.5 = 3 > 2.5 = e(1, 0, \{v_7, v_8\})$$

The summary of results at time 0 is in Table A.4.

	$\{v_1, v_2, v_3\}$	$\{v_4, v_5, v_6\}$	$\{v_7, v_8\}$
$t = 3$	0(n3)	0(n3)	0(n3)
$t = 2$	1(n3)	1(n3)	1(n3)
$t = 1$	1(n3)	8/3(n2)	2.5(n2)

Table A.4: Results at Time 2

Bibliography

[1] M. A. Abdel-Aty, R. Kitamura, and P. P. Jovanis. Exploring route choice behavior using geographic information system-based alternative routes and hypothetical travel time information input. *Transportation Research Record 1493*, pages 74–80, 1995.

[2] M. A. Abdel-Aty, R. Kitamura, and P. P. Jovanis. Investigating the effect of travel time variability on route choice using repeated-measurement stated preference data. *Transportation Research Record 1493*, pages 39–45, 1995.

[3] M. D. Abkowitz. *The impact of service reliability on work travel behavior*. PhD thesis, Massachusetts Institute of Technology, April 1980.

[4] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows*. Prentice-Hall, New Jersey, 1993.

[5] G. Andreatta and L. Romeo. Stochastic shortest paths with recourse. *Networks*, 18:193–204, 1988.

[6] R. Arnott, A. de Palma, and R. Lindsey. Departure time and route choice for the morning commute. *Transportation Research Part B*, 24(3):209–228, 1990.

[7] R. Arnott, A. de Palma, and R. Lindsey. Does providing information to drivers reduce traffic congestion? *Transportation Research Part A*, 25(5):309–318, 1991.

[8] J. Bander and C. C. White III. A heuristic search approach for a nonstationary stochastic shortest path problem with terminal cost. *Transportation Science*, 36(2):218–230, 2002.

[9] M. Ben-Akiva, A. de Palma, and P. Kanaroglou. Dynamic model of peak period traffic congestion with elastic arrival rates. *Transportation Science*, 20(2):164–181, 1986.

[10] M. E. Ben-Akiva, M. Bierlaire, J. Bottom, H. Koutsopoulos, and R. Mishalani. Development of a route guidance generation system for real-time application. In M. Papageorgiou and A. Pouliezos, editors, *Proceedings of the 8th IFAC Symposium on Transportation Systems*, pages 433–438, Chania, Greece, June 1997.

[11] M. E. Ben-Akiva and S. Ramming. Lecture notes: Discrete choice models of traveler behaivor in networks. Prepared for Advanced Methods for Planning and Management of Transportation Networks. Capri, Italy, May 1998.

[12] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, Belmont, Massachusetts, 2nd edition, 2000.

[13] J. A. Bottom. *Consistent Anticipatory Route Guidance*. PhD thesis, Massachusetts Institute of Technology, September 2000.

[14] G. E. Cantarella. A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. *Transportation Science*, 31(2):107–128, May 1997.

[15] G. E. Cantarella and E. Cascetta. Dynamic processes and equilibrium in transportation networks: Toward a unifying theory. *Transportation Science*, 29(4):305–329, 1995.

[16] E. Cascetta. A stochastic process approach to the analysis of temporal dynamics in transportation networks. *Transportation Research Part B*, 23(1):1–17, 1989.

[17] I. Chabini. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Record*, 1645:170–175, 1999.

[18] I. Chabini. Minimum expected travel times in stochastic time-dependent networks revisited. Internal Report. MIT, Cambridge, MA, 2000.

[19] R. K. Cheung. Iterative methods for dynamic stochastic shortest path problems. *Naval Research Logistics*, 45:769–789, 1998.

[20] J. Croucher. A note on the stochastic shortest-route problem. *Naval Research Logistics Quarterly*, 25:729–732, 1978.

[21] C. F. Daganzo and Y. Sheffi. On stochastic models of traffic assignment. *Transportation Science*, 11:253–274, 1977.

[22] G. A. Davis and N. L. Nihan. Large population approximations of a general stochastic traffic assignment model. *Operations Research*, 41(1):169–178, 1993.

[23] A. de Palma, P. Hansen, and M. Labbe. Commuters' paths with penalties for early or late arrival time. *Transportation Science*, 24(4):276–286, 1990.

[24] R. Emmerink, E. Verhoef, P. Nijkamp, and P. Rietveld. Information policy in road transport with elastic demand: Some welfare economic considerations. *European Economic Review*, 42(1):71–95, 1998.

[25] L. Fu. An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B*, 35:749–765, 2001.

[26] L. Fu and L. R. Rilett. Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B*, 32(7):35–52, 1998.

[27] S. Gao and I. Chabini. Best routing policy problems in stochastic time-dependent networks. *Transportation Research Record 1783*, pages 188–196, 2002.

[28] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B (accepted)*, 2004.

[29] R. W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation Science*, 20(3):91–101, 1986.

[30] M. Hazelton. Some remarks on stochastic user equilibrium. *Transportation Research Part B*, 32(2):101–108, 1998.

[31] M. L. Hazelton, S. Lee, and J. W. Polak. Stationary states in stochastic process models of traffic assignment: A Markov Chain Monte Carlo approach. In J-B. Lesort, editor, *Proceedings of the 13th symposium on transportation and traffic theory*, pages 341–359, Oxford, 1996. Pergamon.

[32] H. X. Liu, X. Ban, B. Ran, and P. Mirchandani. Analytical dynamic traffic assignment model with probabilistic travel times and perceptions. *Transportation Research Record 1783*, pages 125–133, 2002.

[33] H. Lo and Y. K. Tung. Network with degradable links: Capacity analysis and design. *Transportation Research Part B*, 37(4):345–363, 2003.

[34] P. Marcotte and S. Nguyen, editors. *Equilibrium and Advanced Transportation Modeling*, chapter 9. Kluwer Academic Publishers, 1998.

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

[36] E.D. Miller-Hooks. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52, 2001.

[37] P. Mirchandani and H. Soroush. Generalized traffic equilibrium with probabilistic travel times and perceptions. *Transportation Science*, 21(3), 1987.

[38] S. Nguyen and S. Pallottino. Equilibrium traffic assignment for large scale transit network. *European Journal of Operational Research*, 37(2):176–186, 1988.

[39] R. B. Noland and K. A. Small. Travel-time uncertainty, departure time choice, and the cost of morning commutes. *Transportation Research Record 1493*, pages 150–158, 1995.

[40] G. Polychronopoulos. *Stochastic and Dynamic Shortest Distance Problems*. PhD thesis, Massachusetts Institute of Technology, May 1992.

[41] G. Polychronopoulos and J. N. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27:133–143, 1996.

[42] D. Pretolani. A directed hyperpath model for random time dependent shortest paths. *European Journal of Operational Research*, 123:315–324, 2000.

[43] J. S. Provan. A polynomial-time algorithm to find shortest paths with recourse. *Networks*, 41(2):115–125, 2003.

[44] H. N. Psaraftis and J. N. Tsitsiklis. Dynamic shortest paths in acyclic networks with markovian arc costs. *Operations Research*, 41(1):91–101, 1993.

[45] D. Schrank and T. Lomax. The 2003 annual urban mobility report. Technical report, Texas Transportation Institute (*http : //mobility.tamu.edu/*), 2003.

[46] Y. Sheffi and W. B. Powell. An algorithm for the equilibrium assignment problem with random link times. *Networks*, 12:191–207, 1982.

[47] D. Watling. A second order stochastic network equilibrium model, I: Theoretical foundation. *Transportation Science*, 36(2):149–166, May 2002.

[48] D. Watling. A second order stochastic network equilibrium model, II: Solution method and numerical experiments. *Transportation Science*, 36(2):167–183, May 2002.

[49] B. Yang and E. Miller-Hooks. Adaptive routing considering delays due to signal operations. *Transportation Research Part B*, 38:385–413, 2004.