Development and Test of Dynamic Congestion Pricing Model

by

Shunan Xu

Bachelor of Civil Engineering

Tsinghua University, Beijing, P.R. China (2006)

Submitted to the Department of Civil and Environmental Engineering

In partial fulfillment of the requirements for the degree of

Master of Science in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2009

Author ....

Department of Civil and Environmental Engineering
January 16th, 2009

Certified by ....

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

Accepted by...

Daniele Veneziano
Chairman, Departmental Committee for Graduate Students

Development and Test of Dynamic Congestion Pricing Model

by

Shunan Xu

Submitted to the Department of Civil and Environmental Engineering
on January 16th, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science in Transportation

Abstract

Dynamic congestion pricing is an approach to control the traffic flow on the network by setting variable tolls that are adjusted with time based on the traffic condition. Different models have been developed and tested in the past. However, most of these models are based on deterministic network equilibrium rather than stochastic choices of travelers, and case studies on complex networks are rare. These disadvantages limit the use of existing models.

In this research, a new dynamic congestion pricing model is developed based on a discrete choice framework to capture users' personal choices. Several solution algorithms are examined and tested in a synthetic network for solving this model. Among these algorithms, SPSA is found to be the best, and is applied successfully to a real case study in Lower Westchester County in New York State. The usefulness and effectiveness of dynamic congestion pricing is also examined and discussed. The results show that dynamic congestion pricing has the potential to improve network performance.

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

Acknowledgements

It is my great pleasure to have memorable experience in MIT, the top engineering school in the world. The experience enlarges my knowledge, and even more, matures my attitude and mentality towards life and the world. I would like to thank my research advisor Professor Moshe Ben-Akiva for his guidance of my research and study, as well as inspirational presence throughout the course of my work at MIT. I would also like to express my gratitude toward my academic advisor Professor Joseph Sussman for his kind help with my life and courses during my master period, especially in my first year. I am especially thankful my lab-mate Yang Wen for spending so much time to discuss with me the algorithms and the programs.

The Intelligent Transportation System (ITS) lab in MIT is a professional but relaxable place for research and life. Senior members in the lab including Yang Wen, Maya Abou Zeid and Charisma Choudhury are great sources for acquiring helps in courses and research. Vikrant Vaze, Vaibhav Rathi, Varun Ramanujam, Eric Huang, Zheng Wei, Lang Yang, Ying Zhu, Sarvee Diwan and Samiul Hasan are all great co-workers to work with.

I would also like to thank my closest friends Yaoqi Li, Shiyin Hu and Wilson Tam for their encouragements and helps during my master period, as well as all of my friends who made the life in MIT a fantastic experience.

Most importantly I would like to thank my parents, for their understanding and supports to my life from the other end of the world. Finally, thanks to God for providing me such a great journey in MIT.

Contents

List of Figures

List of Tables

1. Introduction

Traffic congestion in urban transportation system is an increasing serious problem in major cities all over the world, causing tremendous economic and social problems, such as increased fiscal expenditure, air pollution, increased fuel consumption, and health related issues. Between 1980 and 1997, the aggregate US vehicle-miles traveled in urban transportation systems increased 67 percent, while the lane-miles only increased by 4 percent. As a result of the additional vehicle-mile traveled, the average annual hours of delay per vehicle increased from 100 percent to 300 percent (Skinner, 2000). Studies conducted in 437 US urban areas show that in the twelve months between 2004 and 2005, gas usage increased by 140 million gallons equaling about $5 billion, and aggregate travel time increased by 220 million hours. In 2005, estimated congestion costs amassed to approximately $78 billion as a result of wasted time and fuel (Schrank and Lomax, 2007).

Despite the worsening situation, there are ways to alleviate congestion from both the supply side and the demand side. From the supply point of view, congestion alleviation strategies are generally focused on building more facilities such as roads. However, urban road construction is very difficult and expensive.

From the demand point of view, congestion pricing is often considered one of the most efficient solutions for urban traffic congestion. In 1844, Jules Dupuit established the concept of "road pricing" by introducing a toll for the use of a certain bridge (Hager, 2004). Since the 20th century, it has been well established that people should be charged for road use (Knight, 1924; Pigou, 1912). While certain practical issues are still being discussed, congestion pricing is generally accepted as a good strategy for traffic alleviation and has already been applied or proposed in many cities around the world.

1.1 Overview of congestion pricing

Congestion pricing aims to relieve network congestion or to maximize revenue. While the latter is more often used by private road operators, the former is the focus of public concern. The aim of congestion pricing is to ensure a more rational use of road resources. This is accomplished by charging fees for the use of certain roads in order to reduce traffic demand or distribute traffic demand more evenly among the traffic network through the day. A toll is designed to relieve congestion by changing travelers' behavior. Different classes of users may respond differently to congestion pricing.

Congestion pricing is usually implemented in two main forms, pricing for roads or pricing for parking. Based on different toll strategies, road pricing is classified into three categories:

- Pricing at a cordon area: Drivers are charged a toll for road use in a specific area. The toll may be charged at the gantry when users enter the area or charged at a later time. The latter would require the use of electronic recording devices such as cameras to record and verify the presence of the user. This strategy is usually applied in metropolitan areas, and has been successfully implemented in Singapore and London.
- Pricing on urban roads: This strategy is popular in the US. The toll is collected manually or electronically for the use of a specific road. The tolls, which are collected on the road, may maintain a flat rate throughout the day, or vary based on the time of day.
- Pricing on single facilities: Drivers are charged for the use of specific transportation facilities. One example would be High Occupancy Toll (HOT) lanes where single drivers pay a fee for using HOT lanes, while drivers of high occupancy vehicles are exempted.

Road pricing is either link-based or path-based. Link-based tolls do not depend on a user's travel history because all users traveling through a certain link at the same time are charged the same fee. On the contrary, path-based tolls are dependent on the previous links of a trip. The tolls are based on a path rather than a link. Path-based tolls are difficult to implement in a real network because it is hard to specify each user's travel path. Therefore, for the purpose of this thesis, only link-based tolls will be discussed.

Parking pricing is another component of congestion pricing that is aimed to control traffic by setting varying parking fees based on location, time, and traffic conditions. Parking pricing is considered an important part of congestion pricing and is applied almost all over the world.

Parking pricing is quite different from road pricing strategies in terms of principle and application. While road pricing is generally focused on relieving traffic congestion, parking pricing is usually implemented as a compensation for property maintenance, land use, and environmental pollution. Studies show that the elasticity of parking pricing is smaller than that of road pricing (Alberta and Mahalel, 2006). In addition, parking pricing may elicit diverse and unpredictable responses from users. For example, if a parking lot charges too much, a driver may continue to drive around in an attempt to find cheaper parking. This response to high parking fees may increase traffic congestion. The response to road pricing is often more homogenous and predictable, leading most researchers to recognize road pricing to be technically more efficient than parking pricing.

However, a stated preferences (SP) survey recently conducted as part of a parking pricing study shows that people tend to be more accepting of parking pricing than road pricing (Verhoef, 1995). People usually consider parking pricing as a regular fee for car ownership, similar to fuel tax and maintenance costs. In contrast, road pricing is regarded as a supplemental fee. Thus, from a political perspective, parking policies may receive broader acceptance.

Since this thesis focuses on the technical aspects of congestion pricing, the discussion will concentrate on road pricing. In the following chapters, unless otherwise specified, the term "congestion pricing" will refer to road pricing.

1.2 Response to congestion pricing

The use of congestion pricing to control network flow is based on the principle that travelers will respond to congestion pricing by changing route, switching departure time, changing travel mode, or canceling trip to avoid the toll. Responses to congestion pricing vary between individuals based on their socio-economic and demographic characteristics (income, gender, age, occupation, etc.). In addition, perceptions and attitudes of travelers also affect their decisions. Generally, responses to congestion pricing can be categorized as one of the five types listed in the following:

- Route choice: People may choose alternative routes to avoid a toll. Route choice depends on route attributes (e.g., toll, travel time) and the traveler's characteristics (e.g., familiarity with the network, value of time).
- Departure time choice: If the toll fluctuates throughout the day, travelers may change their departure time to avoid peak charges. Compared to route choice models, departure time choice models are usually much more complicated. Departure time models not only depend on the traveler's characteristics and travel times, but are also affected by penalties for early or late arrival which depend on factors like work schedule and flexibility. Departure time choice is a major focus in current congestion pricing research.
- Canceling trip: If the trip is unnecessary, the traveler may decide to cancel the trip in order to avoid the congestion toll.
- Mode choice: The traveler may also switch from private car use to other modes such as public transportation. Mode choice and the decision to cancel trips are sometimes considered together in congestion pricing research.
- Destination choice: For non-commuter trip travelers, they may change their destination to avoid the toll. This usually happens when people have many alternative destinations, and can choose one of them freely, e.g., shopping. There is no destination choice for commuter trips.

1.3 Static and dynamic congestion pricing

There are two strategies in congestion pricing: static congestion pricing and dynamic congestion pricing. In a static congestion pricing scheme, the toll typically stays the same, or varies but is predetermined for a long periods of time. The toll level is easy to determine and control which is why the static congestion pricing scheme is widely used in practice. Oslo in Norway is a good example of the static congestion pricing scheme. A recent research study indicated that after a static toll was introduced in Oslo, Norway, traffic was reduced by 5% within the first year of operation (Ieromonachou, 2006).

In a dynamic congestion pricing scheme, the toll changes during the day based on traffic conditions on the tolled road. Although a pricing ceiling may established, travelers will not know the exact cost for their trips, but can only estimate based on the time of day, traffic conditions, and historical toll records. Dynamic congestion pricing schemes have demonstrated notable traffic alleviation in real world applications. In 2004, dynamic congestion pricing schemes were introduced in a traffic network in Santiago de Chile, at four different points in the network. The tolls were programmed to have three toll prices and to change from one pricing to the next based on the distance traveled, time of day, and speed of the traffic flow. Since 2004, drivers have reported a significant decrease in travel time. Road safety has also increased as a result of decreased accident rates (Road Charging Scheme, 2004). In Orange County, California, a variable toll that adjusts according to traffic flow and time of day was implemented on road SR 91 Express Lanes. Since the toll was implemented, the congestion on Highway 91 has dropped to a 15-year low (Yildirim, 2001).

A key concept behind congestion pricing is the economic principle that the toll should equal the marginal cost of the network users. Static congestion pricing schemes are unable to capture this principle, making it less effective than dynamic congestion pricing schemes which are able to capture this idea. Consequently, dynamic congestion pricing schemes have received much more attention from the research community.

For the most part, dynamic congestion pricing is still in the theoretical research stage. Although it has been applied in some cities such as Orange County in California (Yildirim, 2001), the toll prices are usually determined based on the traffic conditions of a single road instead of the entire network. There are various limitations associated with such an implementation. First, the network is an inter-connected system. The toll set on one link will affect the whole network. Therefore, to determine toll prices based only on a single road is unrealistic. Second, if the toll is only changed based on the tolled road, the traffic may alternate among several roads. This means that by relieving the congestion of one road, another road may become congested. Therefore, the current implementations of dynamic congestion pricing schemes are still immature.

1.4 Motivation

The dynamic congestion pricing models and algorithms are regarded as some of the most efficient ways to relieve traffic congestion, and are the subject of numerous research studies. However, past research studies have presented several drawbacks. First, some models only consider the route choice behavior, while some others only focus on the departure time choice. Although both aspects are significant, the combination of route choice and departure time choice is usually more important. Second, while most of the models have been tested using real case studies, they are based on network equilibrium rather than on users' choice behavior. Although the network equilibrium approach simplifies the simulation process, it does not adequately show the effects of congestion pricing on users' behavior. Third, almost all of the previous research efforts only tested models on small networks. To test the true validity and efficiency of the models, it is necessary to conduct a real case study using complex networks. These shortcomings motivate the need for further research in dynamic congestion pricing models.

DynaMIT, the software developed by the Intelligent Transportation System (ITS) lab at the Massachusetts Institute of Technology, provides a good resource for dynamic congestion pricing research (Balakrishna and Sundaram, 2005). As a mesoscopic traffic simulation software, DynaMIT generates passenger simulations based on OD flow, people's characteristics, and the personal choice behavior of each user by using discrete choice models. DynaMIT then aggregates the passengers to calculate the final flow on each link. This process is repeated over time to simulate traffic on a specific network. In DynaMIT, each simulated user will be considered individually and his/her choice will be simulated based on current travel time and travel cost. Therefore, DynaMIT provides an excellent tool to test dynamic congestion pricing models based on users' personal choices.

Real data are also available. In collaboration with the New York State Department of Transportation (NYSDOT), real data can be obtained from the Lower Westchester County (LWC) network in New York State, including the network topology, OD flows and sensor data. The validity and efficiency of dynamic congestion pricing models can be analyzed and tested by performing a case study using the real LWC network.

This thesis focuses on dynamic congestion pricing modeling based on discrete choice models, and includes a case study that implements a newly developed dynamic congestion pricing model in the LWC network.

1.5 Implemented framework

The first task of this thesis is to develop a new dynamic congestion pricing model based on previous research. The model's objective is to find a toll strategy that minimizes the aggregate travel time of all network users. The following relationships are considered as constraints:

Network topology: paths, links and lane connections in the network;

Supply parameters: the relationship between link travel time and link flow;

Demand parameters: origin-destination (OD) flows;

Behavior parameters: route choice and departure time choice, which are analyzed using a discrete choice framework.

An experiment to examine the model on a test network will be required. In order to consider users' personal choice, this model will be implemented in DynaMIT. The chosen test network for this experiment has been used for testing in DynaMIT for several years. Some parameters will be calibrated, while others may be cited from other studies.

A real case study will be conducted using the LWC network in New York State. With hundreds of links and nodes, this network is complicated enough for model testing. In addition, the LWC network has been featured in previous research (Vaze, 2007; Rathi, 2007). The calibration and simulation results from previous research studies can be cited and compared to the current congestion pricing research.

1.6 Contribution

This thesis focuses on the development and testing of the dynamic congestion pricing model. Compared with previous studies, this model has several advantages.

First, the model is based on travelers' choice behavior. Compared with the deterministic network equilibrium approach used in previous models, this method with stochastic behavior choice can explain travelers' uncertainty of response to congestion pricing more realistically.

Second, by considering route choice and departure time choice together, the model is able to provide a more comprehensive view of the different effects of congestion pricing, making the results more credible and persuasive.

Third, after initial examination on a test network, the model is tested in a case study on the LWC network. The case study provides a more reliable estimation of the validity and efficiency of the model as well as dynamic congestion pricing.

1.7 Outline

This thesis is organized as follows. Chapter 2 provides a literature review on models, algorithms and tests on congestion pricing, especially for dynamic congestion pricing. Chapter 3 presents a newly developed model that incorporates users' choice behavior using a discrete choice framework. The route choice and departure time choice are presented in this regard. Chapter 4 discusses the methods and algorithms to solve the model presented in Chapter 3. Chapter 5 presents the evaluation of different algorithms based on the experiments conducted using a test network. A modified version of Finite Difference Stochastic Approximation (FDSA) algorithm (FDSA-M) is selected as the best method. Chapter 6 presents a case study where the model from Chapter 3 is analyzed and tested on the LWC network in New York State. Finally, the research conclusions are provided in Chapter 7.

2. Literature Review on Dynamic Congestion Pricing

Congestion pricing has been studied worldwide for several decades. Previous congestion pricing studies usually concentrated on the static aspect, in which the congestion tolls were either flat or pre-set along the day. Although static congestion pricing neglects several significant issues, the models and results in route choice and departure time choice are still quite valuable to be applied in the dynamic congestion pricing model. Recently, dynamic congestion pricing receives more attention than before. Newer models for both analysis and simulation have been developed.

The literature review is structured in the following manner: First, the materials on route choice model and departure time choice model are summarized and analyzed. Next, the studies on the general structure of congestion pricing, especially dynamic congestion pricing, are discussed.

2.1 Route choice models

Route choice is crucial for dynamic congestion pricing. Every commuter trip faces route choice. Although travelers usually have a habitual route for a commuter trip, they may switch to another path to avoid congestion or a toll. Thus, route choice should be considered significantly in every congestion pricing model.

In route choice models, one of the most important issues is how to simulate the uncertainty of travelers' choices. Two main methods are discussed in this regard, fuzzy logic and discrete choice model. The potential application of previous research in this thesis is also discussed.

2.1.1 Fuzzy logic

Fuzzy logic is derived from fuzzy set theory which deals with reasoning by approximation rather than precision. Fuzzy logic allows the user to set values ranging between 0 and 1. Its linguistic form expresses the concepts similar to "slightly", "quite" and "very", known as degrees of the truth, which denotes the extent to which a proposition is true. Degree of truth should not be confused with a probability. For a detailed introduction of fuzzy set and fuzzy logic, see Zadeh (1965).

Fuzzy logic is shown to be a very promising mathematical approach to modeling traffic and transportation processes. It has been applied in various traffic and transportation problems, especially in route choice (Teodorovic and Kikuchi, 1990; Akiyama and Yamanishi, 1993; Lotan and Koutsopoulos, 1993; Akiyama and Tsuboi, 1996). Teodorovic (1999) has provided a good summary on the fuzzy logic research in route choice modeling.

The first application of fuzzy logic in route choice can be found in Teodorovic and Kikuchi (1990). In the paper, an algorithm based on fuzzy logic is developed to solve a two route problem. In this example, route choice simply depends on the travel time. Denote the travel time on route A and B as TA and TB; the fuzzy sets of TA are thus represented as: MLTB – much less than TB; LTB – less than TB; GTB – greater than TB; and MGTB – much greater than TB.

Two fuzzy indices P_A and P_B are used to represent users' preferences for the respective routes. Thus, they are two positive values that sum up to 1. P_A and P_B are highly correlated to the fuzzy sets. For example, if TA = MLTB, P_A is a large number; whereas if TA = MGTB, it is very small. The exact value of P_A can be calculated based on the travel time relationship between the two routes.

When multiple travelers are generated on the network, an algorithm is developed to determine the number of users in each route based on the previous discussion. The users are generated randomly depending on the network characteristics, e.g., the distribution of travel time. The number of users is simply obtained by adding all P_A and P_B up.

This algorithm, although simple, provides a general structure of fuzzy logic application in route choice model.

Akiyama and Yamanishi (1993) have developed another structure of route choice model using fuzzy logic. The authors first conduct a survey to determine the allowable ranges between the informed travel time and real driving time; a triangular fuzzy number (TFN) structure is obtained from the results of the survey. The authors then improve the TFN structure by three methods. The first method (Method I) is generally used to express the similarity which is related to fuzzy truth values. The second method (Method II) can be explained by equality relation between usual numbers with the extension principle. And the third method (Method III) considers the distributed area of TFN.

A test is then conducted to examine the three methods. Method III is found to provide the largest informed travel time, followed by Method II and Method I. Although their results are not as stable as Method I, Methods II and III are still recommended as better approaches after a comparison of their advantages and disadvantages.

Fuzzy logic has been successfully used in traveler's route choice model. However, there are several limitations that confine the further implementations. First, it is lack of theoretical basis to simulate traveler's behavior choice by fuzzy logic. Although the simulation results seem reasonable in many case studies, it is hard to explain traveler's behavior choice from the results. Second, there is still no implementation of fuzzy logic in complex networks. Without the test in real network, the usefulness and effectiveness of fuzzy logic model in route choice area is in doubt.

2.1.2 Discrete choice model

Discrete choice models deal with the problems that involve choices among two or more discrete alternatives. The theory of discrete choice is now well developed and the application is widespread in numerous areas, such as, marketing, shopping and transportation.

In transportation, discrete choice models have been successfully applied to mode choice, traffic assignment, airline management, etc. For detailed descriptions about the theory of discrete choice model and its application, see Ben-Akiva and Lerman (1985).

Discrete choice model has been successfully used in route choice by many researchers. Ramming (2002) has shown the advantages to apply discrete choice model in route choice. These advantages include, excellent simulation results, the ability to explain traveler's behavior and easy to implement.

Ben-Akiva and Lerman (1985) describe a famous example which is often cited by other researchers. This example is usually used to prove the powerfulness of nested logit (NL) model over simple multinomial (MNL) logit in specific problems, including route choice. In this example, two paths are under consideration as shown in Figure 2.1. Near the destination path 2 gets divided into two sections.

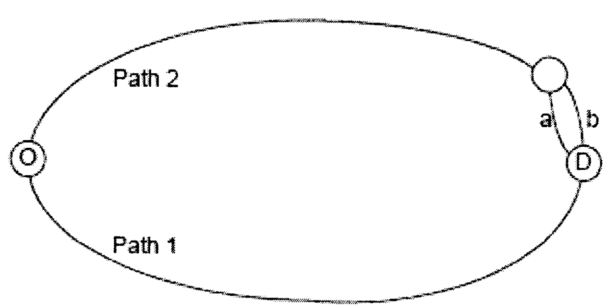


Figure 2.1: Path choice problem

Only consider travel time in the utility function, and assume the travel time in the two paths are the same. Thus, without considering a and b, the system part of the utility for path 1 and path 2 are exactly the same, leading to the probability for choosing each of them to be 50%.

Now consider the small sections a and b as well. There are three paths in the network now: path 1, path 2a and path 2b. Based on the multinomial logit (MNL) model theory, since the travel time among the three paths are the same, the probability to choose each of them is 1/3.

However, this result is unreasonable, because most users cannot identify the difference between path 2a and path 2b, and the probability to choose path 1 and 2 is therefore the same as the previous example, which is 1/2.

The reason for this paradox is that the independent and identically distribution (i.i.d.) property does not hold for path 2a and path 2b, since they overlap each other for most of the part. The paradox can be solved by using a nested logit (NL) model in which the correlation between ε_{2a} and ε_{2b} is considered by a nested structure, where path 2a and path 2b are in the same branch, and path 1 is in another separate branch. The result of the nested logit model is correct.

Although NL model is more powerful than MNL model in route choice, there are still several problems. The most serious problem is, the overlap among different paths is hard to consider in practice. In the previous example, one can simply conclude that the overlap of path 2a and path 2b is very high, so that the nest can be built by assigning them to the same branch. But in the real world, if the overlap of two paths is not so high, it is hard to decide how to build the nest structure. In order to improve it, several other models are developed.

Koppelman and Wen (2000) have developed a paired combinatorial logit (PCL) structure for the route choice problem. In the NL model, covariances are assumed to be equal among all alternatives in a common nest, and zero otherwise. PCL model relaxes the restrictions by allowing for different covariances for each pair of alternatives. The advantages of PCL model are that different competitive relationships can be estimated between each pair of alternatives; and the closed form of the PCL model is computationally less intensive when compared to other logit models.

The PCL model is a member of the generalized extreme value (GEV) family of models (McFadden, 1978). PCL and NL are both general cases of the MNL model; but none of them is a restricted case of the other. In route choice, PCL model can deal with the problems that can not be formulated by NL model. For example, a three path cases in which path 2 is overlapping with path 1 in one section, and is overlapping with path 3 in another section.

Cascetta (1996) has developed a structure for considering the overlapping of the routes, known as C-logit model, which is a modified MNL model. Although some overlapping can be explained by NL and PCL model, when network expands, the nested structure is very hard to build. C-logit uses a different modification from the MNL model by introducing a term called "commonality factor", which can be interpreted as the degree of overlapping of one path with other paths. The commonality factor is added in the utility function of a path to decrease its utility based on the level of overlapping.

A case study was performed based on a sample of 1471 paths chosen from truck drivers on the Italian inter-city road network. The results show significant improvements by the C-logit model to the MNL model specification.

Bekhor et al. (2002) have developed another structure to incorporate behavioral theory in the C-logit adjustment process, known as Path-Size logit. Path-Size logit adds a "size" variable the utility of alternative routes:

$$P(i \mid C_n) = \frac{\exp \mu(V_{in} + \ln S_{in})}{\sum_{j \in C_n} \exp \mu(V_{jn} + \ln S_{jn})} \tag{2.1}$$

where C_n is the available path set for traveler n; V_{in} and V_{jn} are the utilities for the path i and j for traveler n, respectively; μ represents a scale parameter; and S_{in} and S_{jn} are the size variables for path i and j for user n, respectively. The size S_{in} is defined by:

$$S_{in} = \sum_{a \in \Gamma_i} \left(\frac{l_a}{L_i} \right) \frac{1}{\sum_{j \in C_n} \delta_{aj} \frac{L^*_{C_n}}{L_j}} \qquad (2.2)$$

where Γ_i is the set of links in path i; l_a and L_i are the length of link a and path i, respectively; δ_{aj} is the link-path incidence variable that is one if link a is on path j and 0 otherwise; and $L^*_{C_n}$ is the length of the shortest path in C_n.

Ramming (2002) has performed a comparison among many different models in route choice including MNL, NL, C-logit and Path-Size logit, and recommended Path-Size logit as the best. In his thesis, Ramming (2002) also performed a lot of experiments to demonstrate the usefulness and effectiveness of discrete choice model.

Unlike fuzzy logit, discrete choice model has a solid theory base, and can explain the personal behavior choice clearly. Based on all these advantages, it is selected as the model used in this thesis.

2.2 Departure time choice models

Departure time choice is another important part in congestion pricing model. In response to congestion tolls, users may change their departure time similar to changing a route on the network. Departure time choice is more complex than route choice, because while users can choose their routes freely, they cannot depart at any time they want. There is an additional penalty for deviating from the habitual departure time. On one hand, people should arrive on time for commuter trips, so they cannot delay their departure too much; whereas on the other hand, most people do not want to depart too early in the morning either. Thus, the penalty for departure time choice is more complex than the one for route choice.

Departure time choice has been attracting a lot of attention recently in research. Various structures have been developed to formulate it. However, none of them are well accepted.

In a recent paper, Borjesson (2008) developed a mixed logit model to analyze departure time choice. In trip timing modeling, scheduling disutility is formulated as a function of departure time shifts from the most preferred departure time (*PDT*). The shifts are defined as schedule delay early (*SDE*) and schedule delay late (*SDL*):

$$SDE_{in} = \max[PDT_n - DT_{in}, 0] \tag{2.3}$$

$$SDL_{in} = \max[DT_{in} - PDT_n, 0] \tag{2.4}$$

where *DT* is the departure time, *i* refers to alternative and *n* represents different individuals.

The author considers three alternatives for stated preferences (SP) data, two for cars and one for public transport. The utility functions are defined as a combination of travel time, travel cost, SDE, SDL, and the variability of travel time. While combining with the revealed preferences (RP) data, cost is not available. Thus, cost is not considered for the RP data.

For each individual, consider a SP alternatives sequence $i_{SP_n} = \{i_1, ..., i_K\}$, and a RP alternative $i_{RP_n} = \{i_{K+1}\}$. Assuming that the random parameters are independent between RP and SP choices, the conditional probabilities for each sequence can thus be presented as a logit function. Then the unconditional probabilities can be simply calculated by an integral of the conditional probabilities.

The model is then tested by RP-SP data. The reliability ratio is estimated to be 0.74, which is inline with an earlier study (Black and Towriss, 1993). However, this paper does not provide a comparison between the results with other methods. Therefore, the efficiency of the model is in doubt.

Antoniou (1997) has developed the departure time choice structure based on discrete choice model. In his master thesis, Antoniou introduces a general structure of the demand simulation tools for Dynamic Traffic Assignment (DTA), in which the departure time choice is determined within a more general trip decision structure. A time discretization is applied in the departure time choice model, in which the departure time is split into five intervals: depart two intervals earlier; depart one interval earlier; depart on time; depart one interval later; depart two intervals later. In this structure, interval length is critical, and is pre-set by experience and experiments. The five-interval structure can be expanded to more time intervals.

In this structure, departure time is considered together with route choice, where a combination of one departure time and one route is considered as one alternative. A MNL model is applied to determine the right departure time and route. In this process, some weighting are applied in each time interval to capture the passengers' willingness to change departure time, as well as the penalty for early and late arrivals. Weighting parameters are determined by calibration.

Ozbay and Tuzel (2008) have used another approach to consider the departure time choice. In their paper, the authors develop a mathematical formulation at the beginning, in which maximum available early arrival and maximum available late arrival are considered. By applying the Lagrangian multipliers, several important values are obtained, including marginal utility of additional income and marginal utility of additional available time. The authors then analyze the mathematical formulation stepwise to get a utility function, in which the utility is formulated by a complex relationship of travel time, travel cost, deviation from desired arrival time, the time difference between departure and desired arrival time.

The authors then apply this utility in a NL model in a case study in New Jersey Turnpike (NJTPK) to calculate the value of time (VOT) on it. The mean and standard deviation for VOT range from $15/h to $20/h and $2/h to $3/h, respectively, which are consistent with previous research.

Jou et al. (2008) have developed a departure time choice model for commuter trip in the morning. In this model, five different time variables are used: departure time, earliest acceptable arrival time, preferred arrival time, actual arrival time and work starting time. Based on the five times, the time line is divided into four sections, known as early-side loss (segment I), early-side gain (segment II), late-side gain (segment III) and late-side loss (segment IV). The utility of these four sections are all composed of a deterministic part and a random section.

Assuming the random errors are normally distributed, the authors calculate the probability of each segment. The model is then solved by maximum likelihood method. All of the estimators can be estimated accordingly.

The model is tested by a survey data conducted in Taiwan in 2002. It is found that most (52%) of the arrivals are within 5 minutes of their preferred arrival time, and 75% are within 10 minutes. The gain and loss of a late-side are observed to be greater than those of early-side, which is consistent with previous research.

Departure time choice is still an immature area, in which none of the model is well accepted. Among all these models, Antoniou's model based on discrete choice structure is suitable for the dynamic congestion pricing model in this thesis, and is proved to be useful in several real time case studies (Vaze, 2007; Rathi, 2007). Although some modifications are necessary, this model provides us a good approach to consider the departure time choice in congestion pricing model.

2.3 General congestion pricing models

In this section, some general congestion pricing structures are discussed. While the paper in previous sections usually focus on one aspect of congestion pricing, the models in this section all concentrate on the whole picture of congestion pricing. Generally, these methods and algorithms are classified in two types: analysis model and simulation model.

2.3.1 Analysis model

In analysis model, the research usually concentrates on the correctness and completeness of the model, rather than its application. The model is usually quite complex, and requires complete information about the network and users. While providing some theoretical perspectives, the model cannot be applied in practice.

Wie and Tobin (1998) is a good example of analysis model. In the paper, the authors develop two types of dynamic congestion pricing model based on the marginal cost pricing theory. In the first model, the arc capacities and travel demands are stable from day to day; while in the second model, both of the capacity and travel demands fluctuate significantly. The authors show that two type questions can both be solved by a convex control formulation of the dynamic system optimal traffic assignment problem with several different origins and destinations. The equilibrium of the models is proven at last.

In this paper, a general structure of dynamic congestion pricing is developed and proven. One can examine several significant conclusions theoretically by the model. However, in the application part, there are still many works left. First, the homogeneous user assumption is irrational. The authors fail to consider any users' behavior, e.g., route choice and departure time choice in their model. Second, it is almost impossible to charge different users different tolls at the same link based on their destination. Generally, a standardized toll will be charged for all the users (or based on users' type, e.g., cars and trucks) on a link.

In addition, in almost all analysis models, too much information is required to determine the optimal toll. It is too difficult to implement them in practice. This is a common limitation of all analysis models. This limitation confines the further development of analysis models.

2.3.2 Simulation model

In simulation model, researchers focus on the application of the algorithm. The model is usually less complex but more applicable than analysis model. In addition, the authors often provide an example to test the algorithm. Although simple, the example explains some characteristics of congestion pricing. Because the purpose of congestion pricing lies upon its application, the simulation model is regarded as the focus of the congestion pricing research, e.g., Yan and Lam (1996), Verhoef et al. (1996), Joksimovic et al. (2005) and Verhoef (2002).

Verhoef et al. (1996) have used a two-link network simulation model to examine the effects of various demands and cost parameters. Several important topics are discussed and tested in the simple network. Some basic welfare economic properties are discussed for the network with one route tolled and the other untolled. The factors determining the relative performance of one-route tolling are examined. It is found that the lower the two cost parameters (one for free-flow cost and the other for congestion cost) on the tolled route, the more attractive the tolled route becomes. With identical routes, the attractiveness of the tolled route increases with the elasticity of the demand. It is also found that with fixed cost or regulation, the strategy of tolling one route is more efficient than tolling both routes, especially if the cost parameters for the untolled route exceed that for tolled one. The paper also examines the toll strategy by maximizing revenue. This paper, while too simple by considering only a two-route network, provides several useful concepts for congestion pricing.

Yan and Lam (1996) have developed a bi-level structure for determining the optimal toll. A bi-level programming problem has been applied in traffic assignment successfully before (Yang and Yagar, 1994; Yang, 1995). The authors introduce this method into this paper to solve the optimal toll problem. Although the authors only consider the static congestion pricing based on route choice, this bi-level structure is applied broadly in the future dynamic congestion pricing research.

The general structure of a bi-level programming is:

$$\min \ F(u, v(u)) \tag{2.5}$$

$$\text{s.t.} \ \ G(u, v(u)) \leq 0 \tag{2.6}$$

where *v(u)* is implicitly defined by

$$\min_{v} f(u, v) \tag{2.7}$$

$$\text{s.t.} \ \ g(u, v) \leq 0 \tag{2.8}$$

where *F* is the objective function of upper-level decision maker (in this case, system manager); *u* is the decision vector of upper-level decision maker (in this case, road toll); *G* is constraint set of upper-level decision vector; *f* is the objective function of lower-level decision maker (in this case, network users); *v* is the decision vector of lower level decision maker (in this case, network flow); and *g* is a constraint set of lower-level decision vector.

In the bi-level structure, the system manager chooses toll values *u* to optimize his objective function *F*. The network users, after obtaining the information about the system manager's decision, make route choice to minimize their travel cost. The network flow *v(u)* is thus determined by aggregating their decisions. Moreover, for any given toll pattern *u*, it is assumed that there is a unique flow distribution *v(u)*, which is obtained from network equilibrium at the lower-level problem. *v(u)* is also called the response or reaction function. The optimal toll level *u* greatly depends on the evaluation of the reaction function *v(u)*. In other words, users' route decision corresponds to toll charges.

The algorithm is performed by iteration. First, an initial toll pattern *u* is selected. Second, the network flow *v(u)* is calculated by network equilibrium in the lower-level problem. Third, the upper-level objective function is formulated by a local approximation, and a new toll is determined by solving the approximation problem. If the new toll is close to the original one, stop; otherwise, go back to the lower-level again.

The authors then develop two algorithms based on this general structure, applying different optimization method and different strategies to determine the new toll. Two examples are used to test the algorithms. While the examples are quite simple, the results reveal some interesting properties of congestion pricing, and demonstrate the efficiency of the algorithms.

Verhoef (2002) has described an algorithm to calculate the second-best optimal toll levels and toll points in general networks, and test the algorithm in a ten-link network. The algorithm in this paper, although only based on a static research, expands the scope of congestion pricing model, because it can be used for studies of various archetype pricing schemes, including cordon toll, single major highway pricing and parking policies.

The author first develops a model for the fixed location toll. The algorithm determines network equilibrium based on the economic equilibrium principle in which marginal benefits equal marginal private costs, which is known as the standard Wardrop condition (Wardrop, 1952). The optimal toll level is determined by maximizing social welfare, which is defined as total benefits minus total costs. The algorithm is therefore described as two steps. First, for a given network equilibrium with user level, compute the solution to the optimization problem. Second, implement the tolls in the network to get new network equilibrium. The two steps are repeated until convergence.

A numerical example in which some other toll strategies are also implemented in the model is conducted as follows. The network has ten links, three origins and two "real" destinations (W and Y&Z). Since the second destination has different parking strategies, it splits into two different parking nodes, Y for private parking and Z for public parking. Two virtual links are added connecting to Y and Z. In addition, a link with lane charge restriction also splits into two parallel links, one free and the other tolled. The test shows that most of the toll levels converge for an accuracy limit of 1% by five iterations or less. However, an exception exists, for that lane the toll converges for no less than 250 iterations.

The author examines the toll points in the next part. The first single toll point is determined by an indicator I_x, which predicts the welfare gain from implementing a toll on link x from no-toll equilibrium. The algorithm performs very well in the first toll point searching, in which the correction between true and predicted welfare gains is 0.9987. In multiple toll-points case, three attempts have been tried: selecting the toll-points based on the several highest scores; selecting toll-points one-by-one, taking previous toll points given; or, selecting the set of toll-points simultaneously. Among them, strategy two is observed as the best one by considering the accuracy and computation load together. Some combination methods of the three algorithms are also discussed without simulation test.

This paper provides several new ideas, e.g., parking pricing implementation and toll-points determination. However, the author only considers network equilibrium to calculate the network flow. Users' behavior is excluded, and thus cannot be reflected by the model.

Joksimovic et al. (2005) have designed an optimization structure based on the bi-level programming by Yan and Lam (1996). On the upper level, a Mathematical Problem with Equilibrium Constraints (MPEC) formulation is applied to determine the optimal toll by considering the dynamic nature of traffic flows; on the lower level, users' behavior will be considered in route choice and departure time choice by discrete choice model.

The model is composed of three major sections, that is, dynamic network loading (DNL), users' behavior and the road-pricing model. While the road-pricing model is on the upper level, dynamic network loading and users' behavior model constitute the lower level, known as dynamic traffic assignment (DTA) model, which can be converted into an equivalent variational inequality (VI) problem. We describe the two parts of the model separately.

The lower level DTA model contains DNL and users' behavior model, where the latter is the focus of this part. In this part, users' behavior is characterized by a combination of route choice and departure time choice. An appropriate random utility discrete choice model will be used for this purpose. In the users' utility function, both dynamic travel time element and dynamic road-pricing toll element are contained. The departure time choice is considered by introducing penalties for deviating from the preferred arrival time (PAT) and preferred departure time (PDT), a similar approach using in several research (De Palma and Rochat, 1995; De Palma and Fontan, 2001; Gabuthy et al., 2006).

Based on the calculation of the lower level, the road pricing model can be expressed as an optimization problem in the upper level. Mathematical problem with equilibrium constraints (MPEC) is a special case of a bi-level structure where the upper level is an optimization problem and the lower level is an equilibrium problem, refer to Luo et al. (1996) and Lawphongpanich and Hearn (2004).

The bi-level problem is solved iteratively until it converges or reaches certain loops. During the process, the tolls are updated by a simple grid-search procedure each iteration. Some experiments are performed on a simple network that comprises three links.

This paper leads to a good approach to consider the dynamic congestion pricing. The bi-level structure is improved in this paper, where discrete choice model is implemented in the lower level to formulate users' behavior in both route choice and departure time choice, and an optimization problem is developed in the upper level to determine the optimal toll.

This model presents a complete structure of congestion pricing model, in which both route choice and departure time choice are considered. However, there are still some problems in the structure. First, the model considers network equilibrium instead of travelers' behavior choice; second, the optimization algorithm in this paper, a simple random search, is not efficient for big networks; last but not least, just as all previous research, no real case study is performed.

Simulation model has several advantages: first, although simplifying a lot, many models still have good theoretical basis; second, the models are usually easy to apply in the real network, and a case study, although simply, is usually followed after the theoretical discussion; third, the simulation model is easy to absorb good results from other research, e.g., early and late penalty for departure time choice. Therefore, simulation model attracts more attention in the dynamic congestion pricing model research.

Nevertheless, several limitations still exist in simulation model. First, the model always uses deterministic network equilibrium, rather than users' stochastic choices, to determine the network flow. Second, the optimization algorithms are usually inefficient. An appropriate optimization method should be suggested. Third, the test network is too small to examine the validity and efficiency of the algorithms. A real case study is necessary to test the model and algorithm, but is rare in the congestion pricing research today.

2.4 Summary

Current dynamic congestion pricing research is classified into two types: first, focus on the users' behavioral responses to congestion pricing; second, develop general frameworks of congestion pricing model.

Within the users' behavior model research, route choice model and departure time choice model attract the most attention. Two methods are summarized in the route choice model: fuzzy logic and discrete choice model. While fuzzy logic is successfully used in some research, the uncertainty of users' behavior in route choice is explained better by discrete choice model. Departure time choice is more complex than route choice. Penalty is difficult to formulate. Various studies are conducted in this topic, but none of them are well accepted.

Research concentrated on general structure of dynamic congestion pricing often focuses on two aspects of this topic: analysis model or simulation model. Although the analysis model sheds light on some theoretical viewpoints, it is difficult to implement in the real network. Therefore, the simulation model attracts more attention. The effect of dynamic congestion pricing is observed in a lot of research. However, current research on the simulation model has several drawbacks. First, the network flow is determined by network equilibrium rather than users' choice behaviors in most research. Without users' choice behavior, the characteristics of behavior response cannot be reflected in deep. Second, most studies only focus on some aspects of the behavior model. The full set of the behavior model, i.e., route choice, departure time choice, mode choice and canceling trip choice, is rarely considered. Third, the test is only performed in a small network in all the research. However, a case study in real network is necessary to examine the validity and efficiency of dynamic congestion pricing.

3. Model Development

In the previous chapters, the existing literature on dynamic congestion pricing has been summarized, and the common limitations in previous works have been highlighted. The major limitations include ignoring travelers' personal choices in the simulation process, failing to choose efficient optimization algorithms, and the lack of case studies on large networks. In this thesis, these limitations will be addressed by a new model. In this chapter, a new dynamic congestion pricing model based on a discrete choice framework is presented. Instead of considering network equilibrium, this model simulates travelers' choice behaviors on their routes and departure time. The structure of the model is implemented in DynaMIT, a mesoscopic transportation simulation software developed by MIT ITS lab. In Chapter 4, optimization algorithms to operationalize the model are discussed and summarized. Based on previous research, appropriate algorithms are selected and tested on a small network in Chapter 5. A real case study is performed on a network from New York state in Chapter 6.

This chapter is organized as follows: first, the mathematical formulation of the proposed model is developed; second, the values of the important parameters in the model are estimated and calculated.

3.1 Mathematical formulation

In congestion pricing problems, the mathematical formulation is critical. It not only provides a good tool for theoretical analysis, but also presents a clear explanation for the basic concepts of the problem.

In this section, the dynamic congestion pricing model is developed as an optimization problem, where the objective is to minimize the total travel time of all passengers in the network, and the constraints are governed by the network topology, demand, supply and choice behavior of the travelers. All these concepts are discussed in detail below. The simulation conducted in the following chapters is based on this formulation.

3.1.1 Optimization formulation structure

Depending on different purposes, the objectives of congestion pricing can be separated into network optimization and revenue maximization. While the latter is sometimes used in some private companies, the former is the focus of congestion pricing research. Several different quantities can be used to measure the network condition, e.g., total travel time, social welfare and consumer surplus. While the latter two are feasible for both elastic and inelastic demand, the total travel time is only feasible for inelastic demand. But compared to the social welfare and consumer surplus, the total travel time is the more intuitive reflection of the effect of congestion pricing. Since only route choice and departure time choice are considered in this thesis, the demand can be assumed to be inelastic, and the objective function is chosen to minimize the total travel time among all travelers. The optimization problem can therefore be written as:

$$\underset{x}{\text{Min}}\ T(x, \mathcal{S}, \mathcal{D}, \mathcal{N}, \mathcal{C}) \tag{3.1}$$

$s.t.$

$$f(x, \mathcal{S}, \mathcal{D}, \mathcal{N}, \mathcal{C}) = 0 \tag{3.2}$$

$$l_x \le x \le u_x \tag{3.3}$$

where T is the total travel time experienced by all drivers, x denotes the costs (tolls) to be optimized, and $\mathcal{S}$, $\mathcal{D}$, $\mathcal{N}$ and $\mathcal{C}$ represent supply parameters, demand parameters, network topology and travelers' choices, respectively. x are the only unknown parameters in this optimization problem, and are bounded from below by l_x and from above by u_x.

The constraints of the optimization problems are formulated by four modules: network topology ($\mathcal{N}$), supply ($\mathcal{S}$), demand ($\mathcal{D}$) and travelers' choices ($\mathcal{C}$). Notice that this optimization problem can be viewed as a bi-level problem, in which the minimization in (3.1) is the upper level problem, and the evaluation of constraints (3.2) is the lower level problem. In the lower level, the network condition is calculated based on a given toll x. Then in the upper level, for the toll value x, the total travel time is calculated based on the current traffic condition.

One of the main differences between this thesis and previous research is that travelers' personal choices are considered in the lower level problem. The choices are simulated in the traffic simulation software DynaMIT (see Section 5.2.1 and Antoniou 1997 for details of DynaMIT). Since travelers' behavior choices are critical, the following sections focus on the modeling of travelers' behavior choices.

3.1.2 Formula development

In this section, the general formulation developed in the previous section is expanded, and the formulas are discussed in detail. These formulas are all based on discrete choice models, mainly involving route choice and departure time choice. The following notation is used to formulate the dynamic congestion pricing model:

n: Travelers, $n \in 1, 2, 3, \ldots, N$;

nk: Path k of traveler n, $k \in 1, 2, 3, \ldots, K_n$;

nh: Habitual path of traveler n, $nh \in nk$, $k \in 1, 2, 3, \ldots, K_n$;

i: Links, $i \in 1, 2, 3, \ldots, I$;

$x_i(t)$: Toll for link i at time t;

$x_{nk}(t)$: Toll for path nk at time t;

δ_{ink}: Dummy $\begin{cases} =1 & \text{if link } i \in \text{path } nk \\ =0 & \text{otherwise} \end{cases}$;

t_n^d: Departure time for traveler n, $d \in -2, -1, 0, 1, 2$;

$t_i(t)$: Travel time for link i at time t;

$t_{nk}(t)$: Travel time for path nk at time t;

$t_n^{early}(t)$: Early arrival time for traveler departing at time t;

$t_n^{late}(t)$: Late arrival time for traveler departing at time t.

For departure time t_n^d, five different departure intervals will be considered for each traveler:

t_n^0: Habitual departure time;

t_n^{-1}: Departure time for one interval earlier;

t_n^{-2}: Departure time for two intervals earlier;

t_n^1: Departure time for one interval later;

t_n^2: Departure time for two intervals later.

The departure time structure can be expanded to include more departure time intervals in order to simulate travelers' choices if necessary. However, this also makes the model more complex. Thus, the selection of the number of departure time intervals and the interval length need a lot of experiments, and may differ for different networks and OD flows. It is a good topic for further research.

The dynamic congestion pricing model is developed in two steps: First, a departure time is chosen for each individual; second, after a departure time is selected, a route choice model follows based on the chosen time (see Figure 3.1). This is a sequential MNL model.

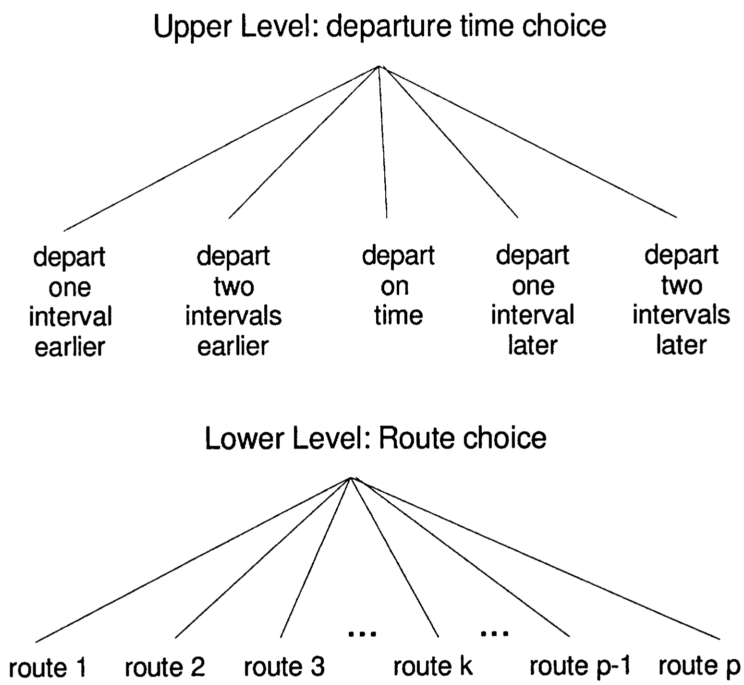


Figure 3.1: Structure of dynamic congestion pricing model

There are several different structures to build the dynamic congestion pricing framework. Two main structures are considered: sequential MNL model and NL model. The sequential MNL model simulates a traveler's decision process as follows: First, the traveler chooses a departure time based on his or her experience of the travel time and the cost on the habitual path (assuming that the habitual path information is available for all travelers before simulation). Next, after choosing a departure time, the traveler will select a path. The NL model simulates the traveler's decision by a different process. In the departure time choice model, the traveler makes the decision not only based on the habitual path, but also on all other available paths by a "logsum" formula. Both structures are reasonable. Experiments are needed to test which one is better. However, the parameters of NL model are correlated to each other, whereas the parameters between two MNL models in a sequential MNL model are independent. In the NL approach, these parameters are also difficult to estimate because of the high correlation. Thus, the sequential MNL model is selected. In this thesis, the data are not sufficient to estimate and calibrate all parameters. Some parameters are asserted from the literature (see Section 3.2).

Departure time choice

Departure time choice is considered in the upper level of the sequential MNL models. Based on discrete choice theory, the probability for traveler n with habitual departure time t to depart in interval d is:

$$P_n^d(t) = \frac{\exp(V_n^d(t))}{\sum_{d'=-2,-1,0,1,2} \exp(V_n^{d'}(t))}, d \in -2,-1,0,1,2 \tag{3.4}$$

where $V_n^d(t)$ is the utility for departing at interval d when the habitual departure time is t. Notice that the corresponding probability $P_n^d(t)$ satisfy the following relationship:

$$\sum_{d=-2,-1,0,1,2} P_n^d(t) = 1 \tag{3.5}$$

There are several different utility function forms for the departure time choice. In this thesis, based on the literature review, four important variables are selected: the cost for the habitual path, the travel time for the habitual path, the early arrival time length and the late arrival time length. A constant is also added to capture the differences among these departure times. The utility function is:

$$\begin{aligned} &V_n^d(t) = \alpha^d + \beta_1^{'} \ln x_{nh}(t) + \beta_2 t_{nh}(t) + \beta_3 t_n^{early}(t) + \beta_4 t_n^{late}(t); \\ &d = -2,-1,0,1,2 \end{aligned} \tag{3.6}$$

where βs are parameters for these variables, and α^d is the constant for departure time choice.

In the utility functions, several issues need to be noticed.

First, α^d ($d \in$ -2, -1, 0, 1, 2) are the constants to capture the difference among different departure time interval options. Generally, the utility function maintains four constant parameters after normalizing one of the five to 0. These parameters can be simplified further using some assumptions. For example, we can assume that the constants for all other departure time intervals except departing on time are all the same. The constant for departure on time is used to capture the inertia of the habitual departure time. In fact, this is the assumption adopted in this thesis. We discuss this point further in Section 3.2.4.

Second, the effect of cost is represented as a "log" function rather than a "linear" function. The difference between linear relationship and log relationship is depicted in Figure 3.2. From the figure, one can notice that the choice probability of the log relationship is much smoother than that of the linear relationship. Since the choice probability should not be expected to increase exponentially with the cost, the log relationship is selected to reflect reality.

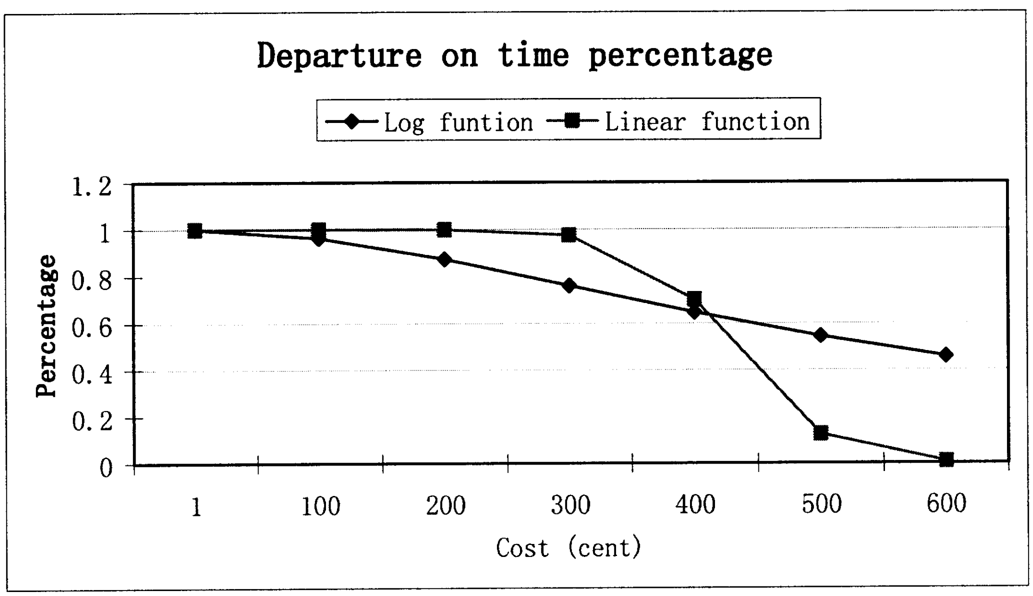


Figure 3.2: Relationship between departure on time probability and cost by different function forms

Third, for $t_n^{early}(t)$ and $t_n^{late}(t)$, at least one of them should be 0. Assume that the traveler arrives just on time if he or she departs at the habitual time; the two variables can be calculated as:

$$t_n^{early}(t) = \max(0, t_n^{d,0} - t + t_{nh}(t_n^{d,0}) - t_{nh}(t))$$
$$t_n^{late}(t) = \max(0, t - t_n^{d,0} + t_{nh}(t) - t_{nh}(t_n^{d,0})) \tag{3.7}$$

If the traveler departs at the habitual time, both $t_n^{early}(t)$ and $t_n^{late}(t)$ are 0.

Route choice

After departure time choice, the traveler chooses his or her route. From a literature review, Path-Size logit is selected for modeling route choice (Ben-Akiva and Bierlaire, 1999, Ramming, 2002), in which a "size" variable is implemented for considering the overlapping of the routes. Similar to departure time choice, the probability for traveler n to choose path nk at time t is:

$$P_{nk}(t) = \frac{\exp(V_{nk}(t))}{\sum_{k=1}^{K_n} \exp(V_{nk}(t))}, n = 1, ..., N \tag{3.8}$$

where $V_{nk}(t)$ is the utility for path nk at time t.

In the dynamic congestion pricing model, costs and times are the only key variables which vary over time. In contrast, other variables, including the "size", remain constant during the optimization process. Therefore, the utility $V_{nk}(t)$ for path nk at time t is written as:

$$V_{nk}(t) = \beta_1 x_{nk}(t) + \beta_2 t_{nk}(t) + C_{nk} \tag{3.9}$$

where β_1 is the parameter for path toll, β_2 is the parameter for path travel time and C_{nk} represents all other variables including the "size".

Unlike departure time choice, a linear relationship is applied in the route choice model between the cost and the utility function. The selection of different function forms is based on previous research as well as a priori hypotheses. While commuter travelers can usually choose routes freely, their departure time is often not very flexible. Thus, it is reasonable to consider route choice more elastic than departure time choice. In this sense, the log function is chosen for departure time choice, and the linear function for route choice.

Objective function

Based on the previous discussion, the objective function can be expressed as a function of the travel time and the travel cost. Recall from equation (3.1) that the objective is to minimize the total travel time T, which is the sum of the expected travel time of all the network users, stated as:

$$T = \sum_{n=1}^{N} \overline{t_n} \tag{3.10}$$

In this expression, the key is $\bar{t}_n$, the expected travel time of user n, which is:

$$\overline{t_n} = \sum_{d=-2,-1,0,1,2} P_n^d(t) \cdot [\sum_{k=1}^{K_n} P_{nk}(t) \cdot t_{nk}(t)] \tag{3.11}$$

where t_n^0 is the default departure time for user n, $P_n^d(t)$ is the probability for user n to choose departure time t, $P_{nk}(t)$ is the probability for user n to choose path nk, and $t_{nk}(t)$ is the travel time for user n on path nk at time t.

In sum, the objective function is:

$$\text{Min} \sum_{n=1}^{N} \sum_{d=-2,-1,0,1,2} P_n^d(t) \cdot [\sum_{k=1}^{K_n} P_{nk}(t) \cdot t_{nk}(t)] \tag{3.12}$$

Constraints

In this model, the cost variables x are unknown. However, since the relationship between x and t is very complex, the constraints among them are quite difficult to express explicitly. By choosing different x, the utility function V changes, this amends the vehicle distribution on the network, and requires recalculation of the travel time t. This complex relationship among x, t and V is expressed as an implicit formula:

$$f(t,x,V)=0 \tag{3.13}$$

This formula does not need to be specified, because the relationship among these variables is simulated in the DynaMIT software automatically.

3.1.3 Summary of problem formulation

In this section, the expression of the optimization problem will be summarized. All the expressions have been discussed in previous sections.

$$\underset{x}{Min} \sum_{n=1}^{N} \sum_{d=-2,-1,0,1,2} P_n^d(t) \cdot [\sum_{k=1}^{K_n} P_{nk}(t) \cdot t_{nk}(t)]$$

$s.t.$

$$f(t,x,V)=0$$

where

$$
\begin{aligned}
&P_n^d(t) = \frac{\exp(V_n^d(t))}{\sum_{d'=-2,-1,0,1,2} \exp(V_n^{d'}(t))} \\
&V_n^d(t) = \alpha^d + \beta_1^{'} \ln x_{nh}(t) + \beta_2 t_{nh}(t) + \beta_3 t_n^{early}(t) + \beta_4 t_n^{late}(t) \\
&P_{nk}(t) = \frac{\exp(V_{nk}(t))}{\sum_{k=1}^{K_n} \exp(V_{nk}(t))} \\
&V_{nk}(t) = \beta_1 x_{nk}(t) + \beta_2 t_{nk}(t) + C_{nk} \\
&t_{nk}(t) \geq 0, \forall k \in K_n \\
&x_{nk}(t) \geq 0, \forall k \in K_n \\
&d = -2, -1, 0, 1, 2 \\
&n = 1, \ldots, N
\end{aligned}
\tag{3.14}
$$

3.2 Parameters implementation

In the previous sections, the optimization formulation has been developed. However, the simulation of the algorithm also requires the availability of all the parameters. The best method to obtain these parameters is estimation and calibration. Estimation is usually applied through the maximum likelihood method to obtain the best parameters fitting the real data. The purpose of calibration is to get parameters with which the results in DynaMIT fit the observed data the best. However, current data cannot support either calibration or estimation for all the parameters, especially the parameters for congestion pricing. For those parameters which cannot be calibrated, the data are used by referring to the literature. All the parameters which are discussed here are required by DynaMIT.

Parameters obtained by calibration

The current data can support calibration for the following parameters:

- Supply model parameters:
  - Segment speed-density parameters (free flow speed v_{max}, jam speed v_{min}, minimum density k_{min} and jam density k_{jam});

- Segment capacities on freeway and arterial segments;
- Capacity factors that determine capacities on segments affected by incidents;
- Parameters of time in the route choice utility function β_2.

- Demand model parameters:
  - The historical OD flows;
  - The variance-covariance matrix associated with the indirect measurement errors;
  - The variance-covariance matrix associated with the direct measurement errors;
  - The matrices of autoregressive factors.

For a detail discussion of parameter calibration, see Balakrishna (2006).

Parameters obtained from the literature

All other parameters in the model cannot be calibrated, and must be asserted from the literature. These parameters include:

- β_1 in route choice utility: parameter for the cost in route choice.
- α^d ($d \in$ -2, -1, 0, 1, 2) in departure time utility: constants for departure time choice.
- β_1' in departure time utility: parameter for the cost in departure time choice.
- β_3 in departure time utility: parameter for the early arrival in departure time choice.
- β_4 in departure time utility: parameter for the late arrival in departure time choice.

We will discuss each of them in this section.

3.2.1 Parameter for the cost in route choice

The parameter for the cost in route choice β_1 is represented in formula (3.9). In this formula, β_2/β_1 is known as the value of time (VOT). Because β_2 can be obtained from calibration, if VOT is available, β_1 can be calculated by the following relationship:

$$\beta_1 = \beta_2 / VOT \tag{3.19}$$

VOT is an important area of transportation research. There are many previous studies on this topic. Table 3.1 is a summary of some of the previous research on VOT.

Table 3.1: VOT estimation based on discrete choice models

Study	Area	Model	VOT
Leurent (1998)	Marseilles, France	RP, binary logit	\$12/h
Algers et al. (1998)	Sweden	SP, mixed logit	\$7.96/h
Hensher (1996)	Austrilia	SP, heteroscedastic logit	\$6.34-\$10.2/h
Calfee and Winston (1998)	Michigan	SP, multinomial logit	\$4/h
Ghosh (2000)	I-15, San Diego	RP, conditional logit	\$22/h (morning)
Sullivan (2000)	SR 91, California	RP, multinomial logit	\$8-\$16/h
Small and Sullivan (2001)	SR 91, California	RP, multinomial logit	\$13-\$16/h
Hultkrantz and Mortazavi (2001)	Sweden	SP, probit	\$6.43/h

Data Source: Ozbay and Tuzel (2008)

From previous research, the VOTs in revealed preferences (RP) data are usually much greater than the ones in stated preferences (SP) data. Compared to RP data, SP data usually contain a lot of biases. For example, people who dislike the congestion pricing strategy may claim much lower value of their time than their actual perception. Thus, RP data are usually regarded as more reasonable and stable. Considering the VOT values in all the research, in this thesis, we use:

$VOT = \$15 / h$.

3.2.2 Parameters for early and late arrival penalty

Parameters for early and late arrival penalty are represented by β_3 and β_4. Among different algorithms in departure time choice research, the concept of early and late arrival penalty is used by many scholars. The penalty for early and late arrival length is usually compared with travel time. The proportion between the parameter for early/late arrival and the parameter for travel time is computed by several studies. Similar as Section 3.2.1, because β_2 can be calibrated, if the proportion β_3/β_2 and β_4/β_2 are available, the parameters β_3 and β_4 can be calculated. The results for the studies on the proportion are summarized as Table 3.2.

Table 3.2: Estimation of the early and late arrival penalty

Date Source	β_3/β_2	β_4/β_2
De Plama and Rochat (1995)	0.3	2.7
De Plama and Fontan (2001)	0.5	2.5
Gabuthy et al. (2006)	0.3	2.5

From previous research, the cost of travel time is usually greater than the cost of early arrival, but is much smaller than the cost of late arrival. Since the estimations for these penalties are similar, we just refer to the most recent results in this thesis, that is:

$$\beta_3 / \beta_2 = 0.3$$

$$\beta_4 / \beta_2 = 2.5$$

3.2.3 Parameters for departure time choice

Parameters for departure time choice include α^d $(d \in -2, -1, 0, 1, 2)$ and β_1'. Since it is very difficult to estimate five parameters together, they need to simplify further. Consider the constants for different departure times. Except the habitual departure time, travelers may not differentiate the difference among other departure times. Thus, we assume that there is a constant (α^0) for habitual departure time to capture the inertia of the traveler's departure time choice, and all other constants are 0, that is, $\alpha^{-2} = \alpha^{-1} = \alpha^1 = \alpha^2 = 0$.

Because of the log relationship between cost and utility function in departure time choice, the parameters for departure time choice α^0 and β_1' are hard to estimate separately. In Jou et al. (2008), the authors report the result that 20% of commuters are likely to switch their departure time. In this thesis, α^0 and β_1' are adjusted to reflect the departure time percentage corresponding to this number.

It should point out that all these parameters need to be checked in DynaMIT, that is to say, the estimation results in DynaMIT should be reasonable under these parameters.

3.3 Summary

A dynamic congestion pricing model is developed and discussed in this chapter to address some of the gaps identified through the literature review in Chapter 2. The model is constructed as a non-linear optimization problem with the objective to minimize the total travel time among all network users. In order to perform simulation, the model is implemented in DynaMIT, a mesoscopic transportation software developed in MIT ITS lab. The parameters in the model, which are necessary for simulation, are also discussed in this chapter. Although calibration and estimation are known as the best methods to get the parameters, some parameters are obtained by referring to other materials because of the lack of data.

4. Solution Algorithm

In Chapter 3, the dynamic congestion pricing model has been developed as a non-linear optimization problem. The implementation of the model has also been discussed. This chapter focuses on the solution algorithm for the model. The characteristics of the dynamic congestion pricing model are analyzed; then the optimization methods for the non-linear problem are overviewed and summarized. Based on the characteristics of the model and the results of previous research, some appropriate algorithms are selected and tested in the following sections.

4.1 Characteristics of the dynamic congestion pricing model

As formulated in Section 3.1.3, the dynamic congestion pricing problem is modeled as follows:

$$\underset{x}{Min} \sum_{n=1}^{N} \sum_{d=-2,-1,0,1,2} P_n^d(t) \cdot [\sum_{k=1}^{K_n} P_{nk}(t) \cdot t_{nk}(t)]$$

$$s.t. \qquad (4.1)$$

$$f(t, x, V) = 0$$

The formulation possesses several characteristics that affect the effectiveness of the solution algorithms: large scale, non-linearity and implicit expressions.

4.1.1 Large scale

Large scale is mainly caused by the dynamic characteristics of the problem. Because the tolls are changed by time, there can be many different toll values in the whole simulation period. For example, in a 10 toll plaza test scenario from 6:00 am to 9:00 am, if we assume the toll is changed every 15 minutes, the total number of unknown parameters is $10\times12=120$, which is a relatively large number for non-linear optimization problems.

4.1.2 Non-linearity

As observed from Equation (4.1), the optimization formulation is highly non-linear. The non-linearity mainly comes from the probability calculation (departure time choice probabilities and route choice probabilities). In MNL model, the probability for an alternative is calculated by an exponential function, where all parameters, including the unknown parameter costs, are contained in the exponent.

There are no universally accepted algorithms for non-linear problem, especially for the large-scale, highly non-linear problems. Because the objective function may contain many local minima, analyzing the shape of the objective function is usually impractical. Numerous algorithms have been developed, but none of them can guarantee the global minima.

4.1.3 Implicit relationship

The implicit relationship mainly comes from the complex relationship among travel time, travel cost and the utility function, which is similar as a black box. Every time it receives an input value, it will return an output. But the explicit relationship is very difficult to obtain. The implicit relationship also affects the objective function, in which travel time and travel cost are interconnected to each other.

Based on the characteristics of the problem, the most time consuming step is the function evaluation. For large network, each function evaluation may take for more than one hour. Thus, the algorithms with less function evaluations are considered more feasible for this problem. The following discussions are all based on this basic point.

4.2 Overview of optimization algorithms

Due to the characteristics of the dynamic congestion pricing model, appropriate algorithms should be selected carefully. In this section, several optimization algorithms are reviewed. Based on the previous research, such as Balakrishna (2006), some algorithms are chosen as candidates, and discussed in detail.

Any optimization problem can be converted to a minimization problem with an objective function to be minimized subject to a set of constraints. If both the objective function and constraints are linear with respect to the unknown parameters, the problem is known as a linear optimization problem, which has been already solved perfectly by simplex algorithm (Dantzig, 1963) and interior point methods (Karmarkar, 1984).

If either the objective function or the constraints are non-linear, the problem becomes a non-linear programming problem. The classical optimization algorithms for the non-linear problem include Newton's Method (Ypma, 1995), Conjugate Gradient Method (Hestenes and Stiefel, 1952), etc. In these methods, a gradient vector is critical, which provides useful information about the direction to improve the objective function. However, because of the characteristics of the dynamic congestion pricing problem, these classical methods are not fit for it. The classical methods can only converge to the local minima rather than the global optimum. Thus, for highly non-linear optimization problem, the results of these methods can be sub-optimal. Moreover, in dynamic congestion pricing model, because of the complex relationship among several variables in the objective function and the constraints, the gradient vector is impossible to obtain directly. Therefore, more appropriate algorithms are necessary.

In this thesis, the optimization algorithms for large-scale, non-linear problems with implicit expressions are summarized into three categories: path search methods, pattern search methods, and random search methods. The categories are derived from Balakrishna (2006).

4.2.1 Path search methods

In path search methods, an initial solution vector is used as the starting point, from which the solution moves certain distance forwarding in a direction so as to improve the objective function value. Usually, the direction is determined by the gradient of the function. If the gradient is not available, it is estimated before moving. Two methods are reviewed in this category: response surface methodology and stochastic approximation.

4.2.1.1 Response surface methodology

Response surface methodology (RSM) (Kleijnen, 1987) is widely applied in highly non-linear optimization. It starts from an initial point, and keeps moving from point to point in the search space. The movement is based on the gradient of the objective function, which is evaluated in the vicinity of the current parameter vector in a local polynomial approximation. The evaluated points are determined systematically, such as from an experimental design. Typically, the response surface is chosen as linear or quadratic.

RSM can be implemented in two ways: sequential search or meta-models. In sequential search method, linear response surfaces are applied repeatedly until the objective function does not improve further. Then a quadratic surface is used until the gradient estimate converges to zero. Sometimes the higher order approximations are also applied, depending on the accuracy requirements of the problem. In meta-models methods, the function values of a set of points will be used to fit a response curve or meta-model. Then the meta-model's gradient is generated, and the parameters and points are updated by deterministic optimization method. The process will be repeated until convergence.

SNOBFIT (Huyer and Neumaier, 2004) algorithm is developed recently as an extension of the RSM type algorithms. Rather than searching for a path to the optimum, SNOBFIT fits a quadratic response surface at multiple points in each iteration. In SNOBFIT, a set of points around the local quadratic surfaces will be generated and maintained. In each iteration of the optimization process, a set of five types of points are recommended: Type 1: a single point to minimize the quadratic approximation; Type 2: a single point generated by solving an identical problem with a certain radius; Type 3: points chosen from the minima of all remaining points; Type 4: points from unexplored regions; and Type 5: points that are generated randomly to maintain a certain number of required points.

At the end of each iteration, a new set of points containing these five types of points is recommended based on minimizations of the quadratic approximations. The number of points in different types is chosen to reflect the balance between global and local searching. The function value is re-evaluated for the next iteration until convergence.

4.2.1.2 Stochastic approximation

Stochastic approximation (SA) is one of the most important methods in the family of path search algorithms. SA, as well as its derived algorithms, traces a sequence of points in the search space that converges to the point with zero gradient of the objective function. Generally, the parameter vector θ_{i+1} at the beginning of iteration $i+1$ is generalized as:

$$\theta_{i+1} = \theta_i - a_i \hat{g}(\theta_i) \tag{4.2}$$

where θ_i represents the parameter vector at the beginning of iteration i; $\hat{g}(\theta_i)$ is the current estimation of the gradient; and a_i is the step size of iteration i. There are several different versions of SA algorithm based on the different gradient functions $\hat{g}(\theta_i)$.

Robbins-Monro Stochastic Approximation

Robbins-Monro Stochastic Approximation (RMSA) is the original version in the family of stochastic approximation technique. In RMSA, the gradient is calculated most intuitively as:

$$\theta_{i+1} = \theta_i - a_i Y(\theta_i) \tag{4.3}$$

where Y is an instance of the stochastic gradient of the objective function f. The expected value of Y is corresponding to the direct derivative of the objective function, which is:

$$E(Y(\theta_i)) = \frac{\partial f}{\partial \theta}(\theta_i) \tag{4.4}$$

RMSA is the most intuitive algorithm in the family of stochastic approximation methods. Although it provides the theoretical basis and is applicable in various problems, there are several drawbacks in RMSA.

First, there is no guarantee that the new point θ_{i+1} will satisfy all the constraints. To expand RMSA by fulfilling the constraints, Kushner and Yin (1998) introduced the projection operator Π_c that would confine the solution inside the feasible space. The θ_{i+1} is thus calculated as:

$$\theta_{i+1} = \Pi_c[\theta_i - a_i Y(\theta_i)] \tag{4.5}$$

Second, RMSA applies gradient measurements directly, which is not available in many complex objective functions. In fact, in the dynamic congestion pricing model in this thesis, the direct derivative of the objective function is impossible to obtain.

To improve RMSA's feasibility in the complicated objective functions where the gradient measurements are not available, two other algorithms are developed, Finite Difference Stochastic Approximation (FDSA) and Simultaneous Perturbation Stochastic Approximation (SPSA).

Finite Difference Stochastic Approximation

FDSA (Kiefer and Wolfowitz, 1952) aims to calculate the gradient measurements by evaluating the objective functions instead of calculating the direct derivatives. Each component of gradient vector is calculated by differencing the objective function, then dividing by the corresponding difference in θ_i values, written as:

$$\hat{g}_{ij}(\theta_i) = \frac{f(\theta_i + c_i e_j) - f(\theta_i - c_i e_j)}{2c_i} \tag{4.6}$$

where e_j denotes the vector with component j equal to 1 and all other components equal to 0. Notice that

$$\lim_{c_i \to 0} \hat{g}_{ij}(\theta_i) = \lim_{c_i \to 0} \frac{f(\theta_i + c_i e_j) - f(\theta_i - c_i e_j)}{2c_i} = \frac{\partial f(\theta_i)}{\partial \theta_i} \tag{4.7}$$

corresponding to the gradients definition in formula (4.4).

FDSA algorithm is performed by the following steps:

1. Initialize at step $i = 0$ by $\theta_i = \theta_0$, the priori values of the initial vector. Choose non-negative coefficients a, A, c, α and γ according to the characteristics of the problem.
2. Move to the next iteration $i = i + 1$; recalculate the step size a_i and c_i as

$$a_i = \frac{a}{(A+i)^\alpha} \quad \text{and} \quad c_i = \frac{c}{i^\gamma}$$

3. The component j of the gradient vector is obtained by formula (4.6). The gradient vector $\hat{g}_i(\theta_i)$ is consisted by J values $\hat{g}_{ij}(\theta_i), j = 1,2,\ldots,J$.
4. Update the point by:

$$\theta_{i+1} = \Pi_c[\theta_i - a_i \hat{g}_i(\theta_i)] \tag{4.8}$$

   where Π_c is the constraints to confine the new vector in the feasible space.
5. Go back to step 2 until convergence or reaching the maximal iterations.

Simultaneous Perturbation Stochastic Approximation

Different from FDSA, in SPSA (Spall, 1992, 1994a, b, 1998a, b, 1999), all the components in the gradient vector are estimated simultaneously by just two measurements of objective function, written as:

$$\hat{g}_{ij}(\theta_i) = \frac{f(\theta_i + c_i\Delta_i) - f(\theta_i - c_i\Delta_i)}{2c_i\Delta_{ij}} \tag{4.9}$$

where Δ_i is a J-dimensional perturbation vector consisting of component-wise perturbations Δ_{ij}.

SPSA algorithm is performed by the following steps:

1. Initialize at step $i = 0$ by $\theta_i = \theta_0$, the priori values of the initial vector. Choose non-negative coefficients a, A, c, α and γ according to the characteristics of the problem.
2. The number of gradient calculations *grad_rep* is selected, which is the average gradient estimate at θ_i.
3. Move to the next iteration $i = i + 1$; recalculate the step size a_i and c_i as

   $$a_i = \frac{a}{(A+i)^\alpha} \quad \text{and} \quad c_i = \frac{c}{i^\gamma}$$

4. A J-dimensional vector Δ_i is generated. Δ_i is the independent random perturbations in

which each element Δ_{ij} ($j = 1,2,\ldots,J$) is chosen from a probability distribution satisfies that:

1) it is symmetrically distributed about zero;

2) both $|\Delta_{ij}|$ and $E|\Delta_{ij}^{-1}|$ are bounded above by constants.

Bernoulli distribution with $\Delta_{ij} = \pm 1$ with equal probability is recommended by literature.

5. Two evaluation points are obtained by $\theta_i^+ = \theta_i + c_i\Delta_i$ and $\theta_i^- = \theta_i - c_i\Delta_i$. The objective function is calculated at these two points, and confined by a pre-determined constraint, e.g., by lower and upper bound.
6. The component j in the gradient vector is approximated as

$$\hat{g}_{ij}(\theta_i) = \frac{f(\theta_i^+) - f(\theta_i^-)}{2c_i\Delta_{ij}} \tag{4.10}$$

7. Repeat step 4 to 6 *grad_rep* times using independent Δ_i, and average the gradient vector for θ_i.
8. Update the point by formula (4.8), where Π_c is the constraint to confine the new vector in the feasible space, and $\hat{g}_i(\theta_i)$ is consisted by J components $\hat{g}_{ij}(\theta_i), j = 1,2,\ldots,J$.
9. Go back to step 3 until convergence or reaching the maximal iterations.

4.2.1.3 Comparison of different algorithms in path search method

From the review of the optimization algorithms and the characteristics of the dynamic congestion pricing model, three algorithms are chosen as the potential solution methods for the problem: FDSA, SPSA and SNOBFIT. Several studies have been already performed in the comparison of these algorithms. In this section, the comparison is introduced by referring to the literature.

Comparison between FDSA and SPSA

The convergence characteristics of FDSA and SPSA depend on the choice of gain sequences $\{a_i\}$ and $\{c_i\}$, but is not guaranteed (Spall, 1999). The best sequences are found to approach zero at appropriate rates neither too high nor too low. The performance of these algorithms thus largely depends on the selection of the parameters a, A, c, α, γ and *grad_rep*.

The effectiveness of the SA algorithms is usually measured by the number of evaluations of the objective function, because it is the most time consuming step in the optimization process in complex non-linear problems. In FDSA, the number of evaluations in each step equals two times the dimension (J) of the gradient vector, plus an additional evaluation in the updating step. Thus, in each iteration,

$$n_i^{FDSA} = 2J + 1 \tag{4.11}$$

In SPSA, there are only two evaluations in each gradient calculations step. Considering the number of gradient calculations *grad_rep* and the additional evaluation at the updating point, in each iteration,

$$n_i^{SPSA} = 2grad\_rep + 1 \tag{4.12}$$

The total number of evaluations in each algorithm is thus:

$$N^{FDSA} = n^{FDSA}(2J + 1) \tag{4.13}$$

$$N^{SPSA} = n^{SPSA}(2grad\_rep + 1) \tag{4.14}$$

where n^{FDSA} and n^{SPSA} are the number of total iterations in FDSA and SPSA, respectively.

Spall (1998b) has performed a detailed discussion of the comparison of efficiency between FDSA and SPSA in the overview of the SA algorithms. In that paper, Spall claimed that in most practical problems, SPSA performed better than FDSA. If the $\{a_i\}$ and $\{c_i\}$ are chosen as recommended in this paper, then by equaling the asymptotic mean squared error $E(\|\theta_i - \theta^*\|^2)$

in SPSA and FDSA,

$$\frac{\text{No. of measurements of the objective function in SPSA}}{\text{No. of measurements of the objective function in FDSA}} \rightarrow \frac{1}{p} \tag{4.15}$$

where p is the number of variables in the objective function.

Spall also performed an experiment of the convergence of a 2-variable function (p=2) in SPSA and FDSA, and showed that the two algorithms reach the solution in same number of iterations (Figure 4.1).

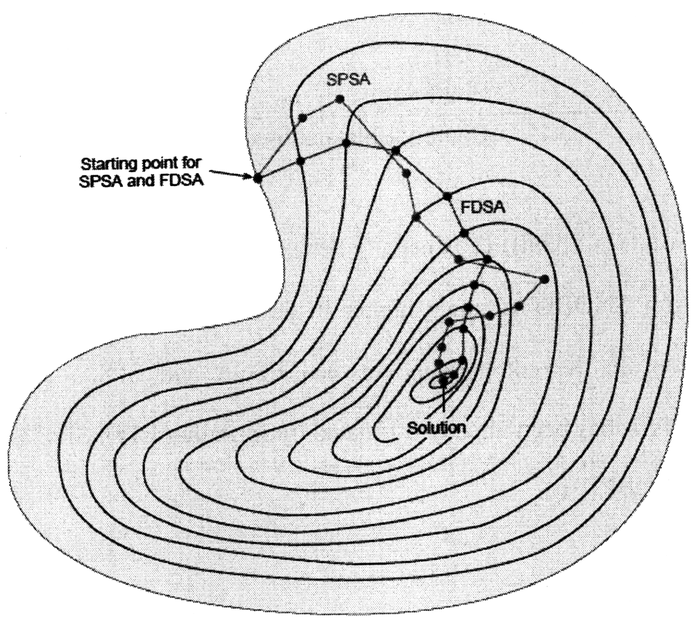


Figure 4.1: Comparison between the convergence of FDSA and SPSA*

However, sometimes in noise-free environment, FDSA is found to perform better if the number of iterations is small (Chin, 1997).

Comparison between SPSA and SNOBFIT

The comparison between SPSA and SNOBFIT can be found in Balakrishna (2006). In his

* Source: Spall (1998b).

thesis, Balakrishna aims to get the best estimation of the demand and supply parameters of DynaMIT. The calibration is performed by minimizing the error measurements root mean square error (RMSE) and root mean square normalized error (RMSN) between the estimated value and the true value of some observed variables, e.g., sensor counts of vehicles. The test is performed in an 8-node synthetic network. The results show that both SNOBFIT and SPSA improve the quality of estimation (Table 4.1).

Table 4.1: Estimation results of SNOBFIT and SPSA

Algorithms	Calibration data by counts	
	RMSE by counts	RMSE by speed
Original	17.19	3.85
SNOBFIT	15.87	3.02
SPSA	15.90	3.17

Source: Balakrishna (2006)

The estimation result of SNOBFIT is usually better than SPSA in the long run. However, the running burden for SNOBIT is much higher in term of the number of evaluations of the objective function. In the comparison between FDSA and SPSA, the number of total evaluations for SPSA has been already estimated by formula (4.14). Similarly, the number of evaluations for SNOBFIT is:

$$N^{SNOBFIT} = n^{SNOBFIT}(J+6) \tag{4.16}$$

where J denotes the number of unknown variables in the objective function, and $n^{SNOBFIT}$ is the number of iterations required for SNOBFIT to converge. Notice that in SNOBFIT, at least $(J+6)$ points are needed in each iteration.

Because the running burden for SNOBFIT is much higher than SPSA, the convergence speed for SNOBFIT in terms of the number of iterations is much less (Figure 4.2). Thus, although SNOBFIT can obtain a better result in long run, it is not fit for the dynamic congestion pricing model, in which the evaluation of the objective function is very time consuming.

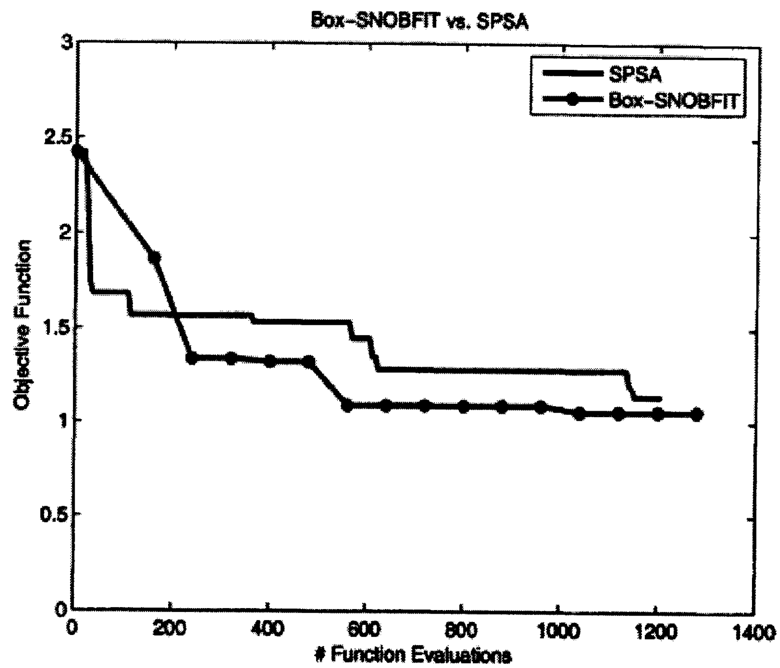


Figure 4.2: Comparison of estimation results for SNOBFIT and SPSA*

4.2.2 Pattern search methods

In pattern search methods, the best direction of movement is determined by the comparison of function values instead of the derivative calculations. Thus, it is often listed under direct search methods.

4.2.2.1 Hooke and Jeeves method

Hooke and Jeeves (1961) developed an algorithm for the non-linear optimization problem. Starting from the initial point θ_0, the algorithm begins to search at the direction of the first component of θ_0 for $j=1$. Two evaluations of the objective function are performed on both sides of θ_0 by maintaining the remaining parameter dimensions $j=2,3,\ldots,J$ constant. If an

* Source: Balakrishna (2006).

improved point is observed on either side, the step size will be reduced. The search will be repeated until a local descent direction is identified for $j=1$.

Then an intermediate point is generated at the minima of the first iteration. The algorithm continues to perform similar steps for $j=2$ at the new intermediate point. This process will be repeated for each $j=3,4,\ldots,J$ respectively. The result is updated after each estimation.

This algorithm is not appropriate for the complex, large-scale and highly non-linear problems. First, this algorithm does not perform well in the balance of local minima and global optimum. Its movement gets usually confined to a local area in the first several steps, and then moves very slowly in the following searches. Second, a new pattern requires at least $J+1$ evaluations of the objective function, which is very time consuming, and thus does not fit the requirements of the large-scale problems.

4.2.2.2 Nelder-Mead (Simplex) method

Nelder-Mead method (Nelder and Mead, 1965) maintains a "simplex" made of $J+1$ points. The initial points are generated randomly confined by the constraints. The evaluations of all these points are performed before the algorithm starts running. In each iteration, the worst point is replaced by a new point at the center of the remaining points in the simplex. The new point is also evaluated. The process is repeated until very little improvement can be achieved by replacing the worst point.

While the Simplex method is useful in practice, the convergence is hard to prove, especially for high dimensional cases with $J>2$ (Lagarias et al., 1998).

4.2.2.3 Box-Complex method

Box (1965) has developed an extension of the Nelder-Mead approach, known as

Box-Complex method. In Box-Complex method, at least $J+2$ points are required to maintain instead of $J+1$ points in Simplex method. Box (1965) suggested $2J$ points as a practical setting.

At the beginning of the algorithm, a "complex" of $S(>J+1)$ points is generated randomly. The objective functions are evaluated in all the points. In each iteration, the worst point is replaced by a new point, which is set to be equal to the centroid of the remainder of the complex, as well as the replaced point. The dimension of complex is expanded in this process. If the new point repeats as the worst point, it moves half the distance towards the centroid of the remaining points. This process is repeated until some pre-determined criteria are satisfied.

Box-Complex algorithm has better ability to achieve the global optimization. However, it cannot guarantee the improvement from the best point at each iteration. In addition, if the worst point repeats, Box-Complex algorithm will perform multiple evaluations on the objective function. In such case, the convergence speed becomes very slow in terms of the number of function evaluations. This is not unusual in the optimization problem with complex objective function.

4.2.3 Random search methods

In random search methods, the search is usually performed in the whole feasible space rather than along a single search path. The parameter vectors are updated by probabilistic mechanisms with the hope of improving towards optimality. Derivatives are not applied in the updating process of the parameters. In this kind of methods, the optimal solution is never guaranteed, and the convergence is not provable theoretically. The effectiveness of the random search methods is mainly based on experiments and tests. In this section, two important random search methods are reviewed, simulated annealing and genetic algorithms.

4.2.3.1 Simulated annealing

Simulated annealing (Metropolis et al., 1953; Corana et al., 1987) is an optimization algorithm simulating the physical process of cooling. A high temperature is chosen at the beginning of the algorithm, and the temperature is reduced in each successive iteration. The aim of simulated annealing is to reach the lowest possible temperature. A "learning" progress is applied in the optimization process by traveling downhill generally with a decreasing probability of traveling uphill. A big probability allows the process to jump out the local optima, but also slows down the speed to convergence. The selection of the probability controls the balance between local minima and global optimum.

The primary advantage of simulated annealing is the ability to jump out the local optimum. It is found to be effective in combinatorial optimization problems. However, the performance of simulated annealing in continuous variables problems and large problems is unattractive. Goffe et al. (1994) analyzed a 2-variable test case with 3789 evaluations of the objective function, which is too heavy for a small example. In addition, because of the lack of theoretical studies on simulated annealing, the convergence cannot be guaranteed.

4.2.3.2 Genetic algorithms

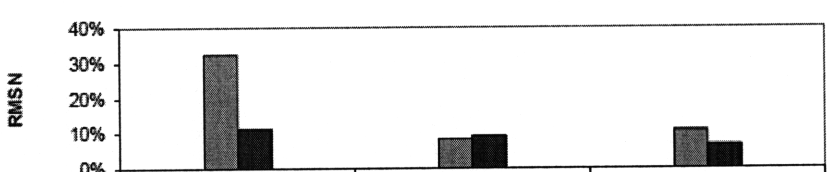
Genetic algorithms (GA) (Holland, 1975; Goldberg, 1989) belong to a class of evolutionary algorithms which simulate the optimization process as the evolutionary progress including inheritance, mutation, selection, and crossover. In GA, the solution point will be represented by an abstract individual, known as chromosome, which is traditionally expressed by strings of 0s and 1s, while other encodings are also possible. At the beginning, a population of a number of chromosomes is generated. In each iteration, some new chromosomes are created; while some existing chromosomes are killed, similar to the biological process of "birth" and "death". The new chromosomes are generated based on mutation and crossover, which also simulates the natural process "inheritance".

These major operations are explained as follows:

- Mutation: the process to generate new chromosomes. Two existing chromosomes, called "parents", switch a part of their elements, getting two new individuals, named "children". When children are born, parents are still alive.
- Crossover: the process to generate new chromosomes. From one existing chromosome, the "father", changes part of its components by some random elements, obtaining a new individual, the "son".
- Selection: at the end of each iteration, because mutation and crossover are performed, various new chromosomes are generated. In selection operation, some chromosomes are killed, either existing individual or new members, to maintain the balance of the population. The selection is based of the fitness of the chromosome. Better chromosomes (e.g., the chromosome with low objective function value in the minimization problem) have lower probability to be killed, and thus are more likely to survive. The selection process maintains a fixed number of chromosomes in each iteration, and helps prevent premature convergence by a probability rule.

Comparison between GA and SPSA

Vaze (2007) has performed a comparison between the performance of GA and SPSA in a synthetic network. In the test, demand and supply parameters of DynaMIT are calibrated, and the objective function is the RMSN between the observed data and the estimated data. The results are shown in Figure 4.3.



	Starting Values	SPSA	GA
Counts	32.49%	8.30%	10.77%
TT	11.29%	9.09%	6.97%

Method

Counts TT

Figure 4.3: Comparison of estimation results for SPSA and GA*

* Source: Vaze (2007).

From the comparison, Vaze concluded that the final results of GA and SPSA are similar. However, the number of evaluations of the objective function in GA is much greater than that in SPSA. Thus, SPSA is considered a better method for complex problem where the function evaluation is very time consuming.

4.3 Summary

In this chapter, various optimization methods have been reviewed. The optimization methods are classified into three categories: path search methods, pattern search methods and random search methods. Based on the characteristics of the dynamic congestion pricing model, several candidates are selected and discussed, e.g., FDSA, SPSA, SNOBFIT and GA. From the previous research, SPSA is regarded as the best performing method in large-scale, non-linear optimization problems with implicit objective functions which are time consuming to evaluate. Thus, SPSA is selected as the first candidate in this thesis, and other methods are also considered if needed.

5. Assessment of algorithms in synthetic network

In this chapter, the dynamic congestion pricing model developed in Chapter 3 is tested on a synthetic network using DynaMIT. Some candidate optimization algorithms are tested in this regard. From the discussion in Chapter 4, SPSA is tested first; a modified version of FDSA, which is named as FDSA_M, is also examined. A conclusion is drawn at the end of this section.

5.1 Case study description

The case study is performed on a small network used in the study by Vaze (2007) with some modifications of the network topology and the demand to fit the congestion pricing purpose better. All components of the experiment will be discussed. Some parameters refer to the results from Vaze (2007). This section is organized in the following manner: First, the network topology and the historical data (such as demand, travel time and toll) are provided; then, the parameters related to the experiment are calculated.

5.1.1 Experiment design

The synthetic network has been used by Vaze (2007) to study the performance of different algorithms in parameter calibration of DynaMIT. Based on the purpose of this thesis, several modifications are made from the previous network in both the network topology and the demand to fit the congestion pricing test better. The network and the corresponding demand are described in detail in this section.

The synthetic network consists of 14 nodes and 16 links. All the links are directed with only one segment, i.e., the cross sectional characteristics do not change in the entire length of the link. Each link contains two lanes, connecting directly to each other at the intersections. The research period is one hour and a half, from 7:30 am to 9:00 am, in which the period from 8:00 am to 8:30 am is the peak hour. The simulation period is half an hour longer than that (7:30 am to 9:30 am) to let all vehicles in the estimation period to finish their trips so that the demand is inelastic and the results are comparable. The estimation length is 15 minutes, which is also the interval length. The simulation is updated every minute. Three toll plazas are set on link 3, link 4 and link 7. The network is depicted in Figure 5.1.

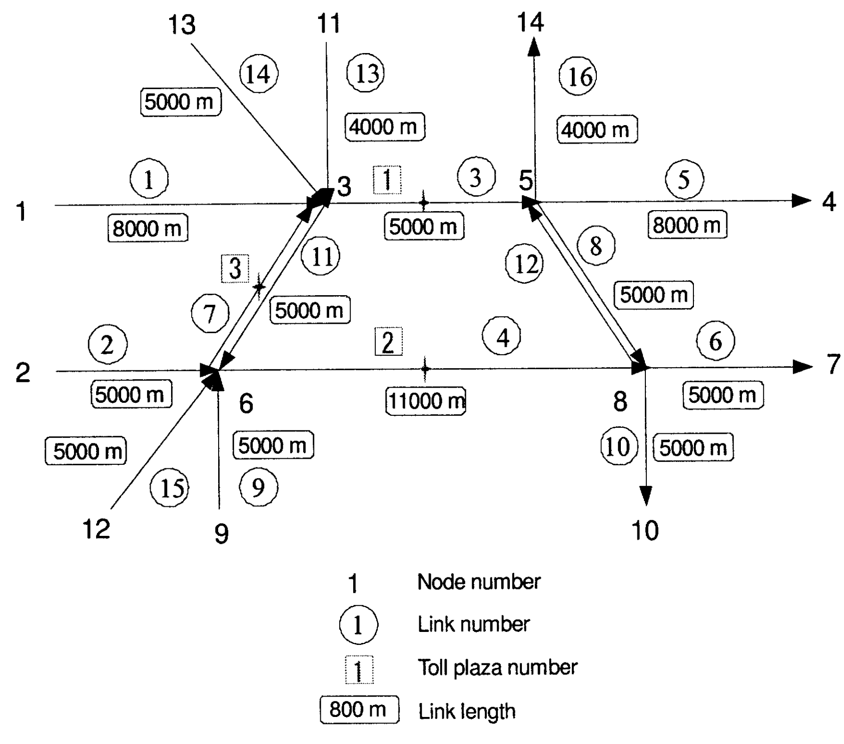


Figure 5.1: Synthetic network topology

16 origin-destination pairs are assigned on the network, among which the ones from 1 and 2 are the major source of the OD flow. The peak hour appears at 8:00 am to 8:30 am, where the heaviest demands are assigned. A reasonable set of OD flows has been selected such that the traffic burden on the network is neither too heavy nor too light. Among all the travelers, it is assumed that half of them are available to receive the real time toll information, which means,

only half of them can react to different toll strategies by changing departure time or changing route.

Three toll plazas are set on the important links of the network (refer to Figure 5.1). Because there is no historical toll data available, a simple toll strategy is used as the historical data, in which the toll is set to $2.00 for peak hours (8:00 am to 8:30 am), and 0 for the rest of time. The historical travel time without toll is simulated by DynaMIT.

5.1.2 Parameter calculation

The methods for calculating parameters in DynaMIT have been discussed in Section 3.2. Based on the discussion in that section, the best approaches to obtain the parameters are estimation and calibration. If neither of them can be supported by the data, the parameters are referring to the literature. In this section, the methods are applied to acquire the value of parameters in the synthetic network.

5.1.2.1 Parameters by calibration

In the synthetic network, the calibration for the parameters has been performed in Vaze (2007) (see Section 3.2). Based on the modification of the network, several parameters are calibrated again. One of most important parameters is the one for travel time in the route choice utility function β_2, which is useful for calculating several other parameters. In this synthetic network, the result is cited from Vaze (2007), which is:

$$\beta_2 = -0.008 \tag{5.1}$$

5.1.2.2 Parameters by referring literature

From Section 3.2, the parameters gotten by referring to literature are: β_1 (the parameter for the cost in route choice), β_3 (the parameter for the early arrival in departure time choice), β_4 (the parameter for the early arrival in departure time choice), α^0 (the constant for habitual departure time in departure time choice), and β_1' (the parameter for the cost in departure time choice).

β_1, β_3 and β_4 can be simply obtained from the relationship of β_2.

$$\beta_1 = \beta_2 / VOT = \frac{-0.008}{15 \times \frac{100}{3600}} = -0.0192 \tag{5.2}$$

$$\beta_3 = \beta_2 \cdot (\beta_3 / \beta_2) = -0.008 \times 0.3 = -0.0024 \tag{5.3}$$

$$\beta_4 = \beta_2 \cdot (\beta_4 / \beta_2) = -0.008 \times 2.5 = -0.020 \tag{5.4}$$

Notice that in DynaMIT, time is in seconds and cost is in cents; thus, VOT needs to convert to cent/second from dollar/hour.

The parameters α^0 and β_1' are obtained based on tests. The tests simulate the commuter trip in the morning from 7:30 am to 9:00 am, 8:00 am to 8:30 being the peak hour, the same as the case study in the synthetic network. The toll is assumed to be charged only in the peak hour, and the habitual travel times are the same for all the departure intervals.

α^0 and β_1' are adjusted together to get a reasonable estimation of the percentage of people who want to change their departure times. From the tests, it is found that α^0=10 and β_1'=-4.4 is a good estimation, in which the results are depicted in Figure 5.2. From the figure, three features can be observed: First, people are more likely to depart earlier than later; this is reasonable because the penalty for late arrival in commuter trip in the morning is much greater than the peak hour toll. Second, the percentage departing one interval earlier is much greater than that for two intervals, which is reasonable because of people's inertia to depart close to the habitual time. Third, under the costs less than $3.00, the changing departure time percentage is about 20%, corresponding to the previous research.

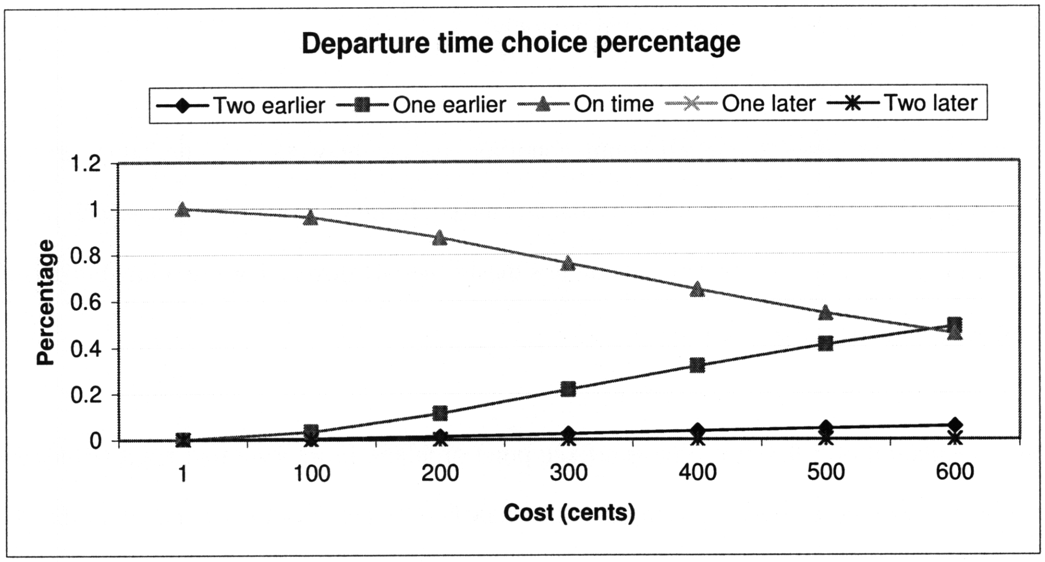


Figure 5.2: Relationship between costs and departure time choices

5.2 Network equilibrium test

When using DynaMIT as the simulation tool, the selection of time convergence algorithms for network equilibrium is critical. The simulation process of DynaMIT is a fixed point problem, in which the equilibrium is said to be reached when the network condition and the travelers' choices corresponding to each other. In this section, a brief review of DynaMIT is provided. Different fixed point algorithms are discussed. Based on previous research and the tests of the different algorithms, Method of Successive Average (MSA) is selected as the best algorithm in this thesis.

5.2.1 Introduction to DynaMIT

The software Dynamic Network Assignment for the Management of Information to Travelers, abbreviated as DynaMIT (Ben-Akiva et al., 1997), is a state-of-the-art dynamic traffic assignment (DTA) system with both real-time and planning applications. In this thesis, only the planning modes are required, which is known as DynaMIT-P, a mesoscopic traffic simulation tool for equilibrium calculation. For simplicity, DynaMIT always refers to

DynaMIT-P in this thesis unless pointed otherwise.

DynaMIT_P estimates the equilibrium condition of the network by simulating drivers' behaviors. DynaMIT_P can be used to calculate the equilibrium link travel time as well as the origin-destination (OD) flow demands. In this thesis, the OD flows are given. Only link travel time is needed to be estimated.

The calculation of link travel time is a fixed point problem. From a starting point (either the free flow travel time or a good estimation), DynaMIT_P generates travelers based on the OD flow demands, and simulates their behavior choices on the network. When the travelers are loaded on the network, the link travel time is calculated again based on the network condition, and is updated according to the current simulation results and the previous estimations. This process will be terminated after convergence or reaching the maximum iteration.

5.2.2 Fixed point problems

In the simulation process of DynaMIT_P, one of the most important steps is the update of link travel time, which is a fixed point problem in which the link travel time should be corresponding to the traffic condition.

In a function, a fixed point is the point which is mapped to itself by the function. That is to say, x is said to be a fixed point of $f(x)$ if

$$x = f(x) \tag{5.5}$$

The estimation of link travel time in DynaMIT is a fixed point problem. In DynaMIT, the simulation of travelers' behavior choices and the aggregation of travelers should be based on a given link travel time. After simulation and aggregation, a new link travel time will be calculated based on the traffic condition. The new time should be corresponding to the previous one; otherwise the guidance is not trustable to the travelers. The estimation of the fixed point for link travel time is a crucial step in the simulation of DynaMIT.

5.2.3 Fixed point algorithms for network equilibrium

There are several algorithms for the fixed point problem. In this section, three of the algorithms are reviewed. For more information about the fixed point algorithms, refer to Bottom (2000).

5.2.3.1 Function iteration algorithm

Function iteration is a well-known method for computing fixed point of a contractive mapping. From a starting point x^0, function iteration algorithm updates the state every step by mapping it to a new point. To specify, for x^k in step k, the state update the result in $k+1$ as:

$$x^{k+1} = f(x^k) \tag{5.6}$$

where f is the mapping function.

A mapping f is said to be contractive, if there exist an $0 < \alpha < 1$ such that $\|f(x^1) - f(x^2)\| \le \alpha \|x^1 - x^2\|$, where x^1 and x^2 are any points in the choice set, and $\|\bullet\|$ is a norm. The contraction mapping theorem claims that if f is contractive on a closed set $X \subseteq R^n$, then f has a unique fixed point in X. Function iteration converges geometrically to the fixed point in the contraction mapping.

If the mapping is not contractive, there is no guarantee that function iteration will converge to a fixed point (if exist one). Under such a case, a common modification for function iteration is using smaller steps when update the states from k to $k+1$. To specify, in step k, the new points is calculated as:

$$x^{k+1} = x^k + \alpha^k (f(x^k) - x^k) \tag{5.7}$$

where $0 < \alpha^k < 1$. Generally, α^k is fixed for each iteration. Obviously, if α^k is close to 0, the point is fixed to itself; if α^k is approaching to 1, it is just the traditional function iteration algorithm. Generally, the algorithm will try different α^k, and chooses the largest one under which the problem does not fluctuate frequently.

However, the modified algorithm does not guarantee a fixed point as well. Its effectiveness and efficiency needs tests for different problems.

5.2.3.2 Method of successive averages (MSA) algorithm

Method of successive averages (Sheffi and Powell, 1982) is a recursive averaging algorithm for the fixed point problems. Under MSA, the point is updated similar as function iteration by

$$x^{k+1} = x^k + \frac{1}{k}\left[f(x^k) - x^k\right] \tag{5.8}$$

Observe that this is just gotten by setting $\alpha^k = 1 / k$ in Formula (5.7) in the function iteration step. This formula means that the effect of the function mapping diminishes with the iteration growth. It is observed that in step k, the new point is updated by the simple average of the mapping of all previous points. This is why we name it "method of successive average".

MSA guarantees a fixed point when the running iteration increases. However, its convergence speed is sometimes too slow to apply. Cascetta and Postorino (1998) reported that in MSA, the later iterations, which are closer to the solution, receives smaller weights when computing a new estimate. Several modifications of MSA are generated to improve the convergence rate, but none are dominated. For detail of MSA, refer to Bottom (2000).

5.2.3.3 Polyak iterate averaging algorithm

Polyak iterate averaging algorithm updates the new points by a different manner. Polyak and Juditsky (1992) showed that in the state update step in Formula (5.7), if α^k approaches to 0 slower than $1 / k$ in a suitably-defined sense, then the sequence $\sum_{i=1}^{k} x^i / k$ converges to the fixed point at an optimum rate. This algorithm is known as Polyak iterate averaging algorithm.

As mentioned before, Polyak algorithm has two steps. The first step is a MSA-like step that:

$$x^{k+1} = x^k + \beta i^{-\gamma}(f(x^k) - x^k) \tag{5.9}$$

where ½ < γ< 1. In this thesis, β and γ are chosen by a common choice which are used in Bottom (2000), β= 1 and γ= 2/3.

In the second step, the point is updated by a simple average of the results in the first step, which is:

$$\tilde{x}^{k+1} = \frac{k}{k+1}\tilde{x}^k + \frac{1}{k+1}x^{k+1} \tag{5.10}$$

where $\tilde{x}^0 = 0$. Bottom (2000) reported that in some stochastic problems, Polyak algorithm converges faster than MSA.

5.2.4 Tests of fixed point algorithms

In the previous sections, several different fixed point algorithms have been discussed. All of them are feasible for some problems. In order to choose the most efficient one, some tests are necessary. In this section, the measurement rules are discussed first, and the tests are performed for three algorithms, which are, function iteration, MSA and Polyak iterate averaging. MSA is selected as the best algorithm in this thesis.

5.2.4.1 Measurement rules

It is necessary to discuss the measurement rules first, that is, how to measure the goodness of different test algorithm. Based on the tests of Bottom (2000), a 2-norm measurement is chosen in this thesis, which is:

$$IC = \left[\sum_l \sum_t [x_{lt} - f(x_{lt})]^2\right]^{\frac{1}{2}} \tag{5.11}$$

where l is for all links, and t is for all time periods.

Bottom (2000) has also tested some other measurement rules. The first one is root mean square (RMS), which is obtained by dividing IC by the total number of time periods and the number of links. RMS is rejected in this problem, because there are a lot of "empty" links with free flow link time. In this case, there will be a lot of zeros which do not contribute to the total norm value. Thus, RMS misleads the result by dividing the 2-norm by the total number of links.

Another measurement is the weighted norm in which each link and time period's discrepancy is multiplied by the corresponding traffic volume. This measurement is also rejected, because the link volumes may change between replications, causing impossible a meaningful comparison of norm values.

5.2.4.2 Test results for the base case without toll

In this section, the tests are performed in the basic case, in which the link travel time is chosen by free flow travel time, and the toll is set to 0 in all toll plazas and all time periods. Five results of three different algorithms are tests: function iteration with current iteration weight as 0.3 (FI3), function iteration with current iteration weight as 0.2 (FI2), MSA, the MSA-like step in Polyak iterate averaging (Polyak1), the average step in Polyak iterate averaging (Polyak2). The total running iteration is set by 300 for each. The results are depicted in Figure 5.3.

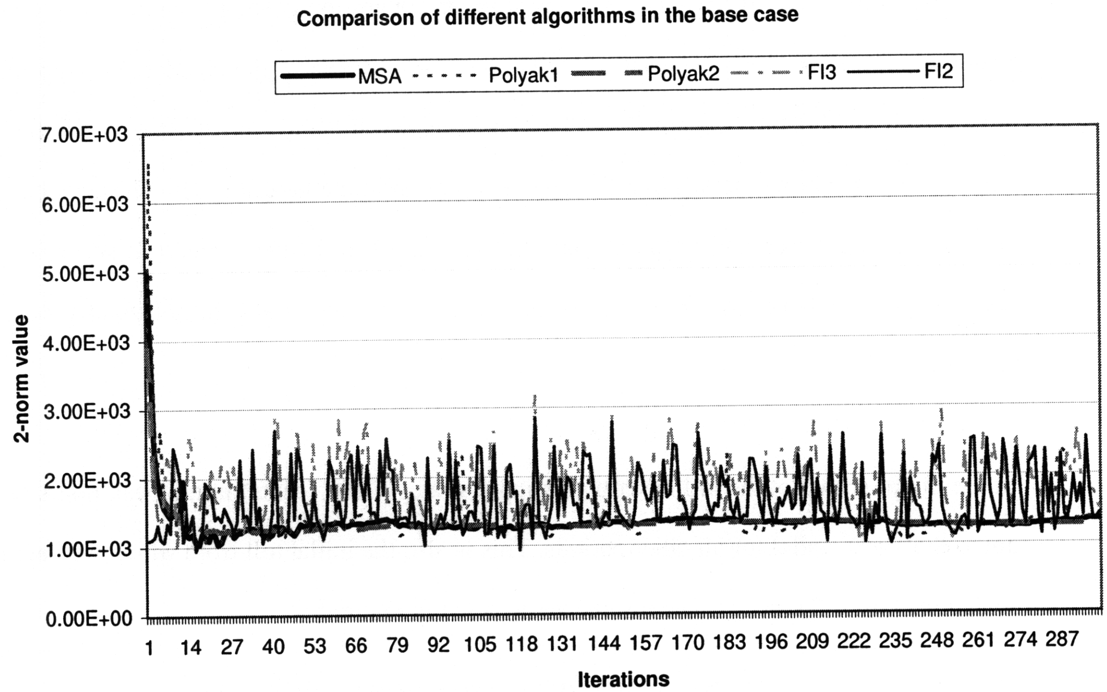


Figure 5.3: Comparison of different algorithms in the base case

Based on the test result, the following conclusions are drawn:

1. The function iteration algorithms are not feasible for the problem, because the results always fluctuate frequently even the weight in the current iteration is very small.
2. Polyak1 performs better than function iteration algorithms. However, it still fluctuates frequently. No obvious convergence is observed after 300 iterations.
3. MSA and Polyak2 are the best performance algorithms. The results of both algorithm converge after no more than 20 iterations, and the results are good only after 5-10 iterations.

Based on the results, MSA and Polyak2 are selected for the following tests.

5.2.4.3 Test results for other cases with tolls

Although MSA and Polyak2 work well in the base case, further tests are necessary for the performance of them in the toll cases, because all congestion pricing scenarios will be run under some toll conditions, and the results are trustable only if the fixed point algorithms works well in the toll scenarios. In the toll scenario, the link travel time is set by the equilibrium travel time obtained in the precious section in the no-toll cases. These values are considered as a better starting point than the free flow travel time. Three toll scenarios are considered: Initial scenario, random scenario with 15 minutes flat toll, and totally random scenario.

In the initial scenario, the toll is set by a simple strategy: 200 (cents) in the peak hours (8:00 am to 8:30 am), and 0 for all other periods. This is also the initial toll from which we begin to search the best toll strategy. The results are depicted in Figure 5.4.

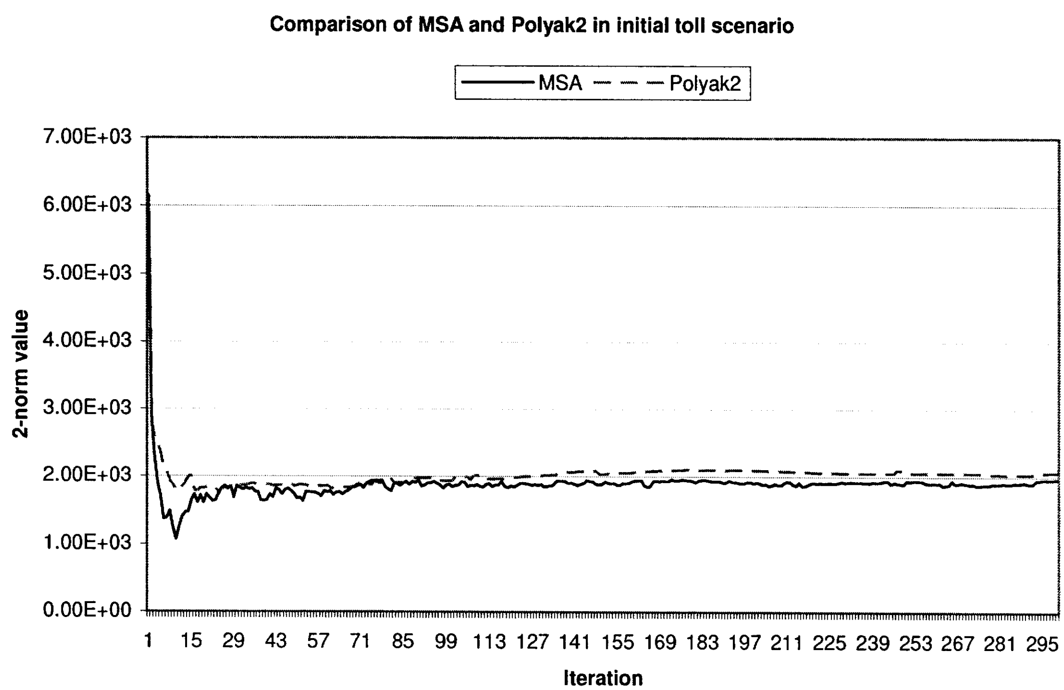


Figure 5.4: Comparison of MSA and Polyak2 in initial toll scenario

From the results, following conclusions are obtained:

1. Both MSA and Polyak2 perform well in this scenario. The results converge in no more than 20 iterations, and the results are good within only 5-10 iterations.

2. The toll makes the results more unstable than the no-toll scenario. The converged 2-norm values are much greater than the ones in the no-toll scenario.

In another toll scenario, we introduce more randomness. The toll is set to be flat in every 15 minutes, but is randomly generated from the range 0 to 400 (cents). The toll scenario is represented in Table 5.1, and the results are depicted in Figure 5.5.

Table 5.1: 15-min flat toll scenario

Toll Plaza	Time							
	7:30	7:45	8:00	8:15	8:30	8:45	9:00	9:15
1	0	0	348	336	15	131	46	75
2	6	0	392	322	105	108	89	157
3	148	222	126	170	36	107	173	46

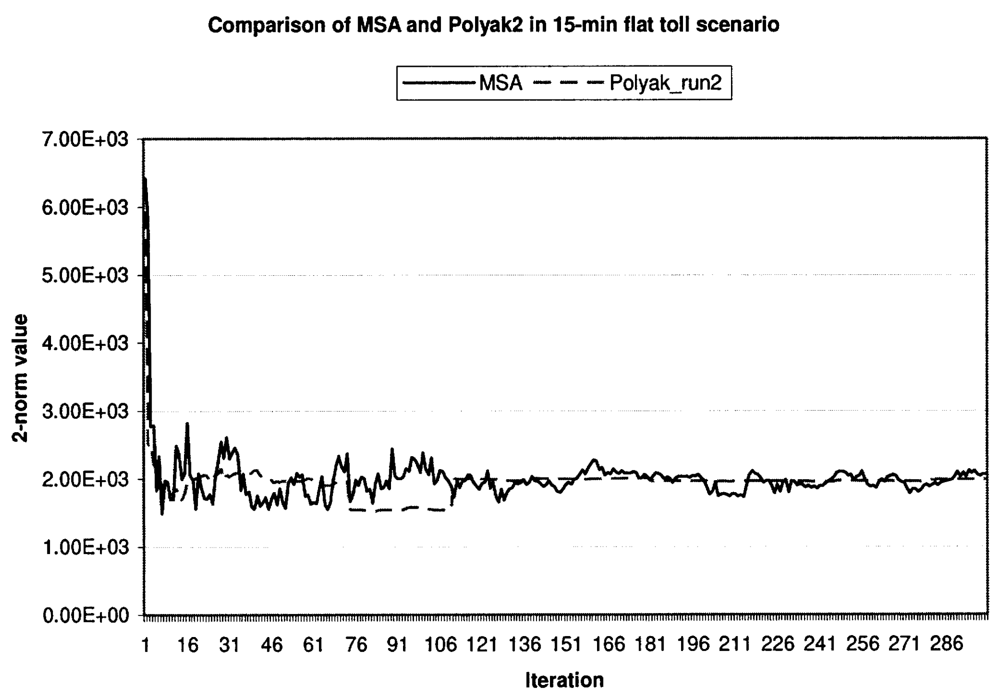


Figure 5.5: Comparison of MSA and Polyak2 in 15-min flat toll scenario

Following observations are made from the results:

1. Both MSA and Polyak2 perform well in this scenario. The results converge in no more than 20 iterations, and the results are good within only 5-10 iterations.
2. The toll converges to a similar level as the initial toll scenario, but with much more

fluctuation.

In the last scenario, we generate random toll for each toll plaza and each minutes. The toll range is still set by 0 to 400 (cents). The results are depicted in Figure 5.6.

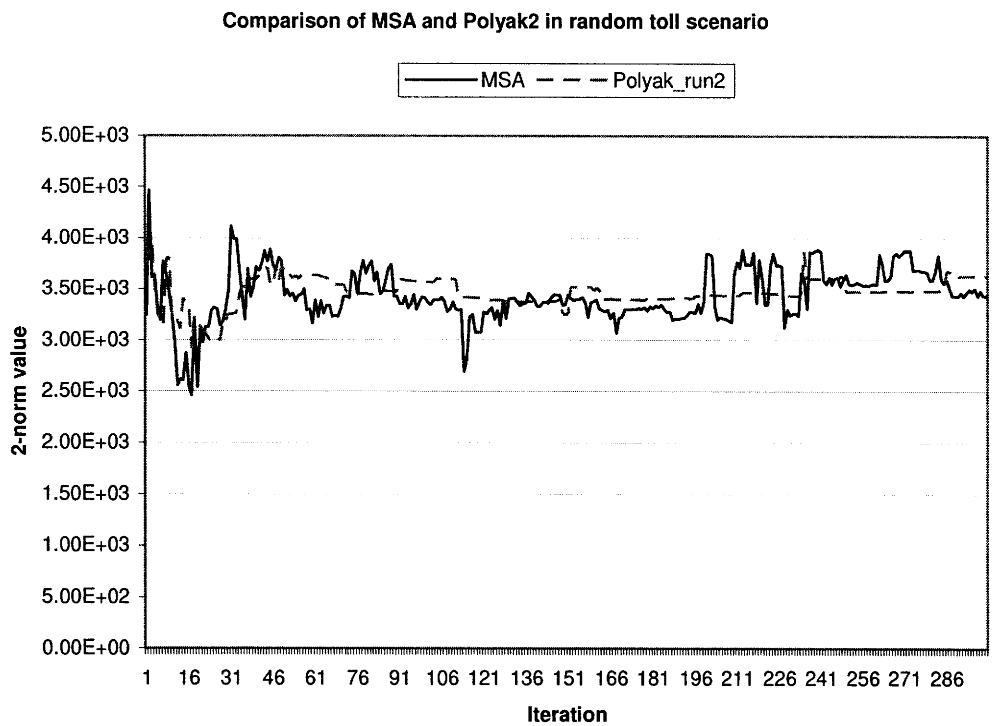


Figure 5.6: Comparison of MSA and Polyak2 in random toll scenario

Obviously, the results fluctuate after 300 iterations, and no obvious convergence is observed in both algorithms.

From the tests, the effect of tolls is clear. The introduction of tolls will cause difficulty of the convergence of link travel times. The more complex the toll scenario is, the more fluctuate the results do. If the toll is totally random, none of the algorithm can provide a reasonable estimation of link travel time. However, if the toll is set to be flat in 15 minutes, the results gotten by MSA and Polyak2 are both acceptable.

The last question remained is how to choose between MSA and Polyak2. In this thesis, because of the time constraint, only 10 iterations are used in the test network, and only 5 iterations are performed in the real network. Among this range, MSA usually performs a little better than Polyak2 from out tests. In addition, Polyak2 requires an additional estimation of the function, thus is less attractive than MSA when the running iteration is small. Based the above reasons, MSA is selected in this thesis.

5.2.5 Summary of fixed point algorithm

The calculation of equilibrium travel time is a fixed point problem. In this section, different fixed point algorithms are discussed, and three of them are tested, which are, function iteration, MSA and Polyak iterate averaging. Through the tests, MSA is selected as the best algorithm in this thesis. It is also observed that the fluctuation growths with the randomness of the toll strategies. When the toll is set to be flat in 15 minutes, the results of MSA are acceptable within only 5-10 iterations.

5.3 SPSA description and test

In previous chapters, SPSA has been identified as the best algorithm. Therefore, it is applied first in the case study.

5.3.1 Algorithm description

Based on the characteristics of the problem and the quality of the result, SPSA is modified by setting the toll to be flat in 15 minutes. There are two reasons for setting the flat toll. First, from the result in section 5.2, it claims that all fixed point algorithms in the totally random toll scenario (that is, the toll can be different every minute) do not converge in 300 iterations. Thus, the simulation result in that scenario is not trustable. Second, the flat toll is also much more realistic than the totally random toll, because the travelers usually cannot accept the toll charges which fluctuate too frequently.

SPSA algorithm is implemented in MATLAB, in which DynaMIT is called to calculate the total travel time among all passengers. The unknown variables are the number of flat tolls for all time intervals. In this case study, it is 3×120/15=24. In each iteration, all the tolls are updated and saved in the historical toll plaza file in DynaMIT for the estimation in the next iteration. The objective function value (total travel time) is calculated from the outputs of DynaMIT. Based on the result in section 5.2, MSA is used as the fixed point algorithm and the running iteration within DynaMIT is set to 10, after that the network is considered to reach an equilibrium. All the constraints are satisfied within DynaMIT software.

The process is described in Figure 5.7.

Algorithm description:

1. Initialize parameters a, A, c, α and γ;
2. Set cost $C = C_0$ at step $i = 0$;
3. Calculate the initial total travel time T_0 based on C_0 by DynaMIT, and set current total travel time $T = T_0$;
4. Update step $i = i + 1$;
5. Calculate $T+$ and $T-$ ($f(\theta_i^+)$ and $f(\theta_i^-)$ in formula (4.9)) by DynaMIT;
6. Calculate the current gradient vector by formula (4.10);
7. Repeat step 5 and 6 *grad_rep* times using independent Δ_i, and average the gradient vector for θ_i;
8. Update current cost C by formula (4.8), in which Π_c is simply defined as an upper and lower bound constraint;
9. Update T by DynaMIT based on current costs;
10. If the termination requirements are satisfied, stop; otherwise, go back to step 4. The termination requirements in this paper are either convergence, meaning the total travel time is flat in several iterations, or reaching maximum running times;
11. The best tolls are set by the ones under which the total travel time is minimal.

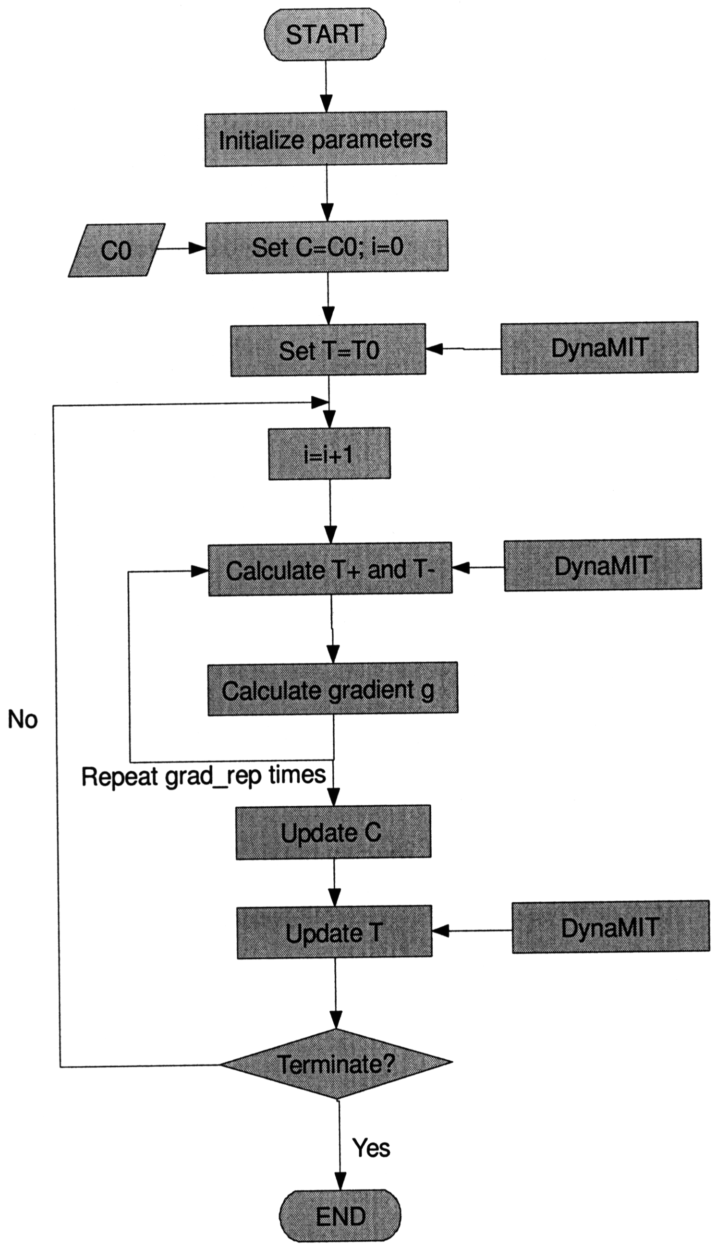


Figure 5.7: SPSA algorithm process

5.3.2 Parameter selection

The parameters a, A, c, α, γ and *grad_rep* are critical for the performance of SPSA. Spall (1998b) recommended the values for α and γ as 0.602 and 0.101, respectively. a, A and c are different from experiment to experiment. In this thesis, the value $a = 0.01$, $A = 10$ and $c = 30$ are found to work well.

Notice that a and A are used to control the step size of price changing in each iteration; and c is used for managing the distance between the evaluation points and the current price. In this synthetic case study, the value $c = 30$ implies that in the first step, $T+$ and $T-$ are evaluated by 0 cents lower or higher than the current price. Under $a = 0.005$ and $A = 10$, it is found that the costs of toll plaza are changed by no more than 100 cents (less than 50 cents usually) in the first several iterations. After 80 iterations (which is the maximum iteration in this case study), the value decreases to about ¼ of the initial value. Notice that a, A and c should be tested again if network topology, demand or maximum iteration changes.

grad_rep is used to control the calculation accuracy of gradient vector. The larger this value is, the more accurate the gradient vector is. The gradient vector is perfectly accurate when *grad_rep* approaches to infinity. However, the running burden also increases with the *grad_rep*. Thus, the selection of *grad_rep* is significant to the problem. Balakrishna (2006) has reported it works the best when *grad_rep* = 2. In this thesis, three numbers are tested, 1, 2, and 3, and the best one is chosen to perform the following tests.

5.3.3 SPSA test

Three tests are performed by SPSA independently for each *grad_rep*. In each of them, the maximum toll is set to $4.00, and the minimum toll is 0. In order to make the running time to be comparable, the maximum iteration is set to 80 for *grad_rep* = 1, 40 for *grad_rep* = 2, and 27 for *grad_rep* = 3. The test results are depicted in Figure 5.8 to Figure 5.10, and the best toll strategies are shown in Table 5.2 to Table 5.4. The comparison among them are shown in Table 5.5.

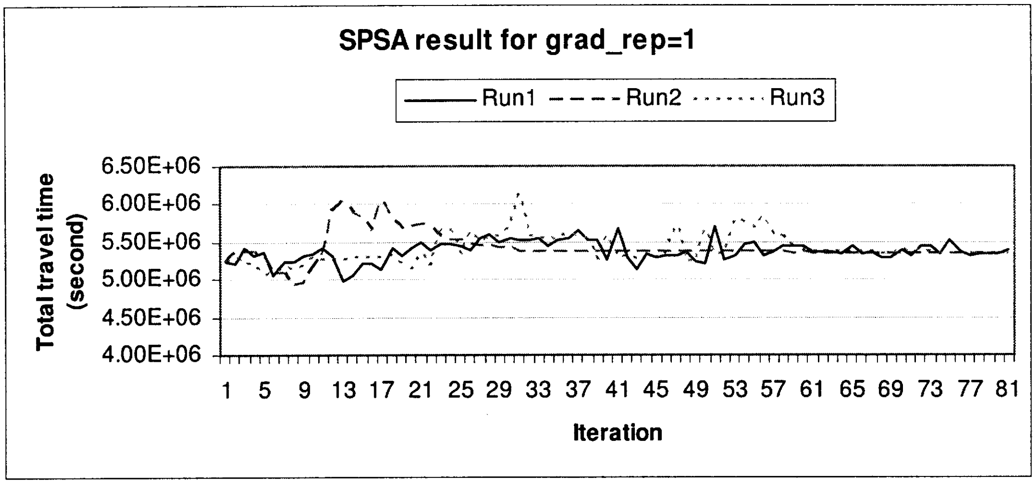


Figure 5.8: Result of SPSA for grad_rep=1

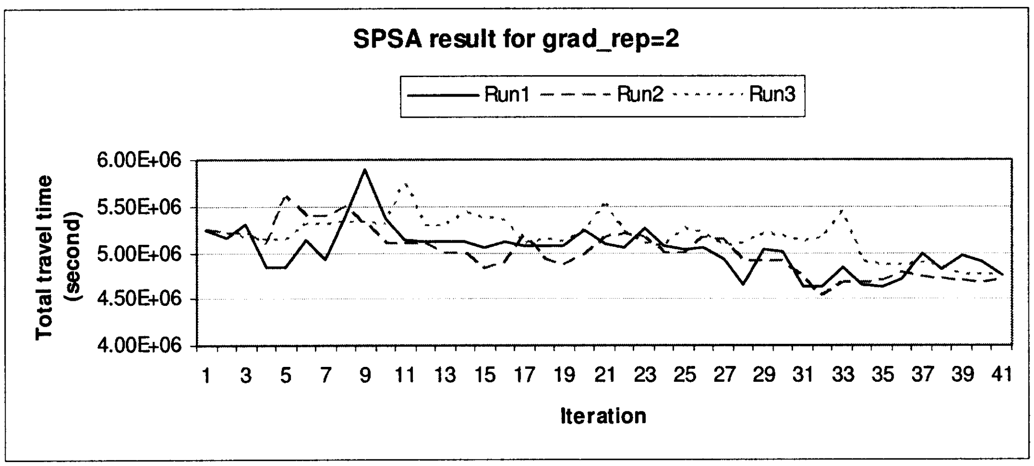


Figure 5.9: Result of SPSA for grad_rep=2

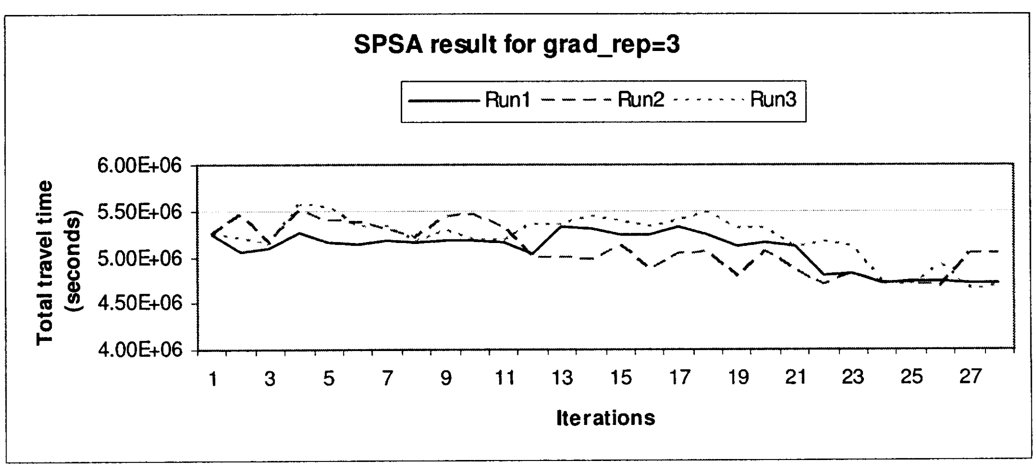


Figure 5.10: Result of SPSA for grad_rep=3

Table 5.2: Optimal toll of SPSA for grad_rep=1 (in cents)

Toll Plaza	Time					
	7:30	7:45	8:00	8:15	8:30	8:45
1	155	56	75	8	10	0
2	203	0	219	187	190	8
3	93	110	400	332	63	90

Table 5.3: Optimal toll of SPSA for grad_rep=2 (in cents)

Toll Plaza	Time					
	7:30	7:45	8:00	8:15	8:30	8:45
1	109	0	363	349	0	68
2	62	0	287	348	140	64
3	23	110	40	94	61	16

Table 5.4: Optimal toll of SPSA for grad_rep=3 (in cents)

Toll Plaza	Time					
	7:30	7:45	8:00	8:15	8:30	8:45
1	78	0	209	228	32	13
2	74	0	139	215	24	34
3	0	91	223	55	13	0

Table 5.5: Best results for different grad_reps (in seconds)

grad_rep	1	2	3
Run1	4978550	4621660	4710990
Run2	4925390	4517870	4683110
Run3	5058730	4752300	4650390

From the results, we have the following observations:

1. The algorithm works very poor when grad_rep equal to 1. When grad_rep is set by 1, the gradient is inaccurate. The perturbation is much more random than the cases where grad_rep is greater than 1. The effect of random perturbation is that the result will not converge after certain iterations, which is shown in Figure 5.10, where the objective function decreases in the first several iterations, but fluctuate after that.
2. The results for grad_rep=2 and grad_rep=3 are similar, while the former is slightly better. We can conclude that grad_rep=2 is a good choice in this example, although

we cannot prove it is the best simply from the result. Further, in the real case study in Chapter 6, the number of iterations is much less than the one in the synthetic case study because of the slow calculation of the network condition. In that case, the advantage of grad_rep=2 is expected to be greater, because the number of running iterations for grad_rep greater than 2 could be very small. In sum, in this thesis, grad_rep=2 is used in the following sections.

5.4 FDSA-M description and test

Although SPSA algorithm when *grad_rep* = 2 works very well for the test network, it is worth to test some other algorithm. One candidate is a modified version of FDSA algorithm. We name it FDSA-M in this thesis.

5.4.1 Algorithm description

Beginning from the first period, FDSA-M only considers one interval each time. For example, in the first period, only the three tolls in the first interval are regarded as the unknown parameters, to which the similar optimization process as SPSA is performed. During this process, all future tolls are referred to historical values. After converging or reaching the maximal running time, the algorithm moves on to the second period, and so on. The process of FDSA-M algorithm is depicted in Figure 5.11.

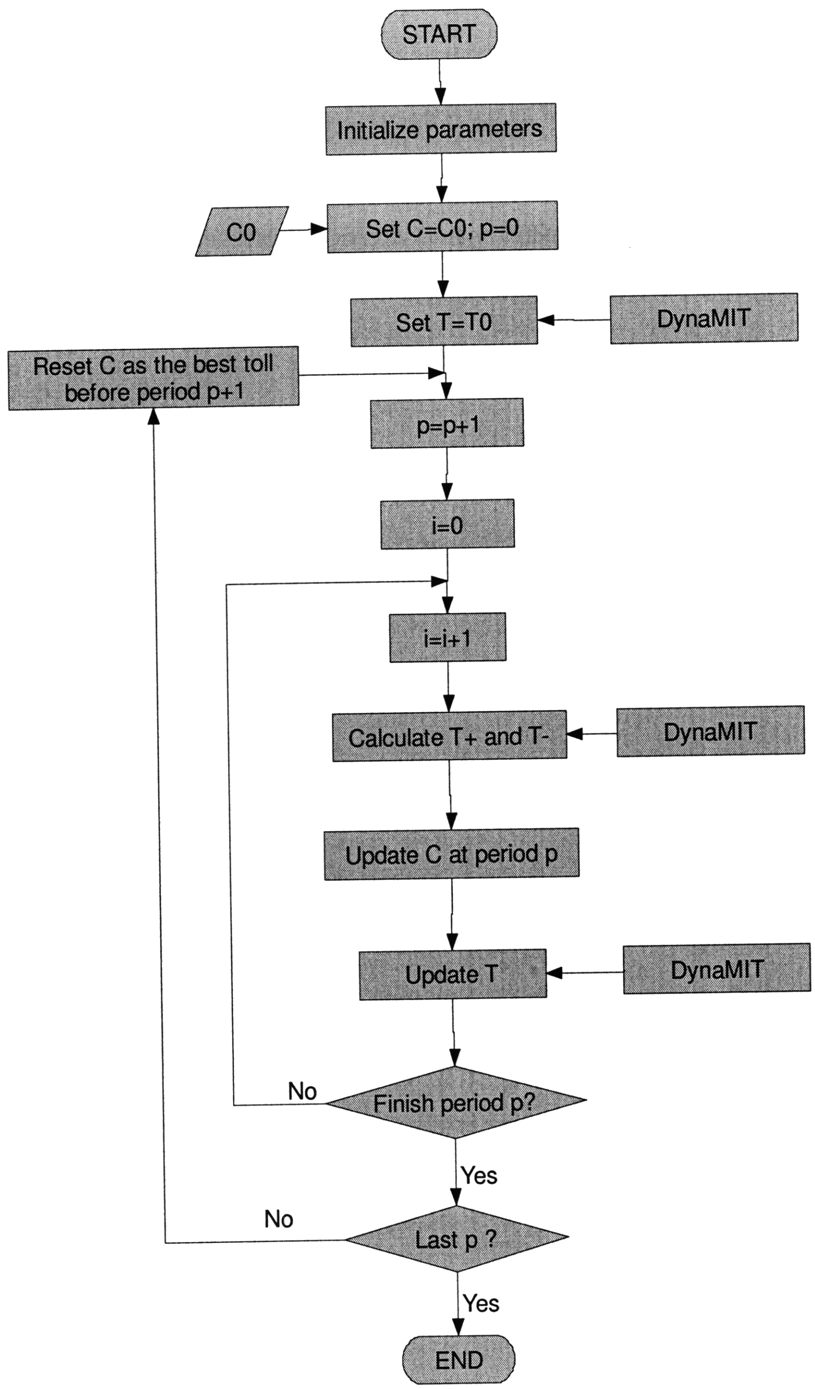


Figure 5.11: FDSA-M algorithm process

In FDSA-M, a similar process as SPSA will be applied to update the cost. The algorithm is described following:

1. Initialize parameters a, A, c, α and γ;

2. Set cost $C = C_0$ at period $p = 0$;
3. Calculate the initial total travel time T_0 based on C_0 by DynaMIT, set current total travel time $T = T_0$;
4. Move on to the next period $p = p + 1$;
5. Initialize $i = 0$;
6. Update step $i = i + 1$;
7. Calculate $T+$ and $T-$ ($f(\theta_i + c_i e_j)$ and $f(\theta_i - c_i e_j)$ in formula (4.5)) by DynaMIT;
8. Update current cost C by formula (4.6) and (4.8), in which Π_c is simply defined as an upper and lower bound constraint;
9. Update T by DynaMIT based on current costs;
10. If the termination requirements are satisfied for period p, go to step 11; otherwise, go back to step 6. The termination requirements are similar as SPSA;
11. If p is the last period, stop; otherwise, reset cost C as the best prices until now, and go back to step 4;
12. The best tolls are set by the ones under which the total travel time is minimal.

Notice that FDSA-M is different with FDSA. In FDSA, the costs in all periods are updated one by one in one running iteration, and then continue to the next; while in FDSA-M, the costs are only updated for one period, and move on to the next period after it converges or reaches the maximal running time. The difference between FDSA and FDSA-M are depicted in Figure 5.12, where "iteration" refers to the number of times for running the algorithm, and "period" means a length of time (e.g., 15 minutes) in the simulation process. In the figure, there are P period in the simulation time, and the maximal running iteration is set by N.

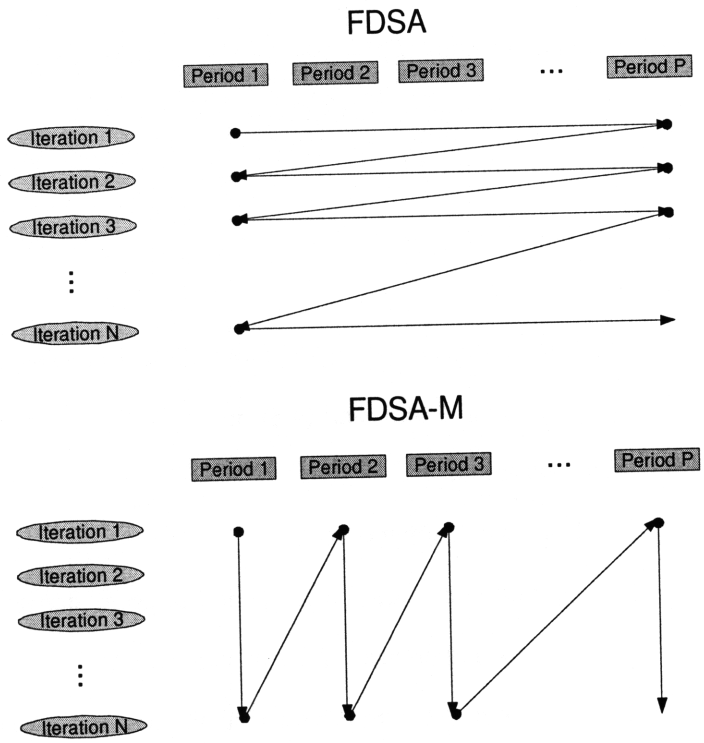


Figure 5.12: Differences between FDSA and FDSA-M

5.4.2 Parameter selection

The constants a, A, c, α and γ are similar as the ones in SPSA. For α and γ, the values are still set by 0.602 and 0.101, respectively. However, a, A and c may be different for FDSA-M. From tests, the value $a = 0.015$, $A = 5$ and $c = 30$ are found to work well. The meaning of these parameters can be discussed similar as Section 5.2.2. Because the simulation is performed for all period, the running iteration in each period is much smaller than that of SPSA. Thus, in FDSA-M, *grad_rep* = 1 for all tests.

5.4.3 FDSA-M test

Three tests are performed independently. The test results are depicted in Figure 5.13 and Table 5.6. In all tests, the maximum toll is set to \$4.00, and the minimum toll is 0, the same as SPSA. Notice the difference between "iteration" and "period". In the synthetic case study, the maximum iteration for SPSA is set to 80. For FDSA-M, there are 8 periods totally, and 10 iterations for each period. Thus, the total number of running iterations for both algorithms is the same; and running time is comparable between them.

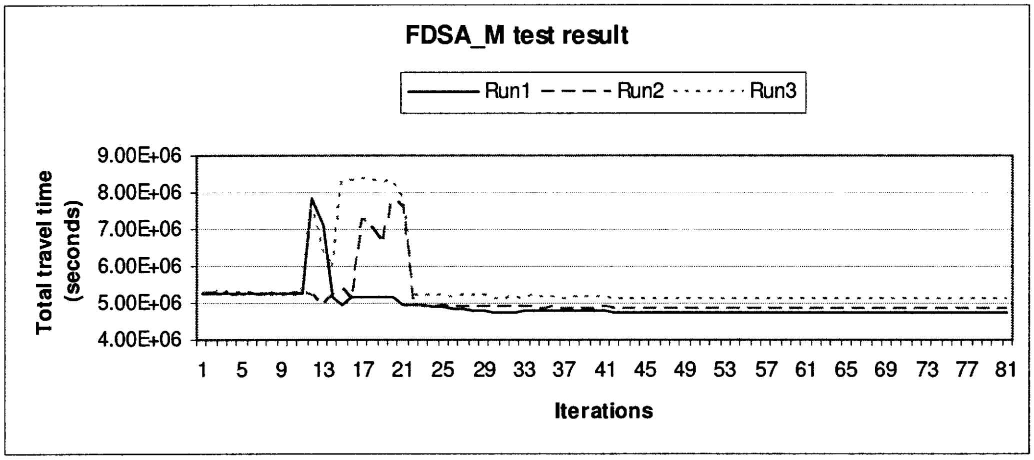


Figure 5.13: Result of FDSA-M

Table 5.6: Optimal toll for FDSA-M (in cents)

Toll Plaza	Time					
	7:30	7:45	8:00	8:15	8:30	8:45
1	26	0	261	200	5	0
2	5	0	251	200	0	1
3	48	220	220	200	6	0

Based on the simulation purpose of this thesis, FDSA-M is not as attractive as SPSA, because the best toll scenarios found in FDSA-M is worse than SPSA (Table 5.7). The potential reason is following: Compared to SPSA, the effect of the tolls in FDSA-M are significant in some "important" periods, while are insignificant in some others. In all three runs in this example, nearly all the main decreases of the total travel time happen in the third and fourth periods, which are the peak hours. However, the algorithm runs 10 iterations for all periods, either for the important ones or for the insignificant ones. This cause some runs are useless. In order to improve it, we may change the number of running iterations by setting more runs in the important period. However, it is usually hard to specify how to recognize the important periods, and how to assign the number of running iterations to them. This could be a topic in the future studies.

Although FDSA-M is not as good as SPSA in this thesis, it may be more useful than SPSA in the real time version. In this thesis, the toll is calculated offline. Thus, for a 2-hour simulation research, the toll should be calculated for 8 periods, causing the number of running iterations in each period much less than that of SPSA. However, in the real time version, the toll is calculated online. Thus, only the toll in the next period is needed to be simulated. In that case, the number of iterations in each period can be set the same as the one for SPSA. Under this case, FDSA-M works better than SPSA.

5.5 Result analysis

Previous sections in this chapter focus on the description and the test of different algorithms. Among them, SPSA is recommended. In this section, the results among SPSA, FDSA-M and some other scenarios are compared and discussed. The effectiveness of congestion pricing is also examined.

5.5.1 Comparison among different scenarios

From previous Sections 5.3 and 5.4, the best toll strategies are obtained by different algorithms. In this section, the results are summarized for SPSA and FDSA-M. Two other toll strategies are implemented for comparison. The four toll strategies are selected as:

1. Zero toll;
2. Initial toll setting at the beginning of the algorithm, which is, $2.00 in the peak hour and 0 at the rest time;
3. Best toll in three runs for SPSA;
4. Best toll in three runs for FDSA-M.

The results of the four toll strategies are summarized in Table 5.7. In order to compare the results more intuitively, the average travel time is also provided.

Table 5.7: Overview of four toll strategies

Toll strategies	Zero toll	Initial toll	SPSA			FDSA-M		
			Run1	Run2	Run3	Run1	Run2	Run3
Number of vehicles	12141	12141	12141	12141	12141	12141	12141	12141
Total travel time (seconds)	5724410	5240500	4621660	4517870	4752300	4743070	4827690	5114890
Average travel time (seconds)	471.5	431.6	380.7	372.1	391.4	390.7	397.6	421.3

From the results, it is observed that both SPSA and FDSA-M improve the performance of the network from the initial toll strategy. SPSA works better than FDSA-M. For a simple estimation, in the best toll scenario of SPSA, the total travel time is decreased by 21% from the zero toll case, and 14% from the initial toll scenario. This proves the usefulness of the congestion pricing, and the effectiveness of the SPSA algorithm.

Notice that in this synthetic network, the effect of congestion pricing seems unrealistic. The reason is that the network is very small, and the effect of congestion is too great to be true. A more realistic estimation will be provided in the real case study in Chapter 6.

5.5.2 Result for the effect of congestion pricing

Following from Section 5.5.1, more tests are performed by comparing the four scenarios, including travel time distribution, departure time choice and route choice. In the SPSA and FDSA-M scenario, the best toll strategies are selected among the three runs to perform the tests. The effect of congestion pricing is shown from the comparison.

The first test is for travel time distribution, in which the number of vehicles whose travel time is within a range will be summed up together. The travel time distribution is an intuitive indicator of the network condition. It is expected that under efficient congestion pricing, more vehicles will decrease their trip times. The travel time distribution among all passengers is illustrated in Figure 5.14, which shows the number of vehicles in each travel time length.

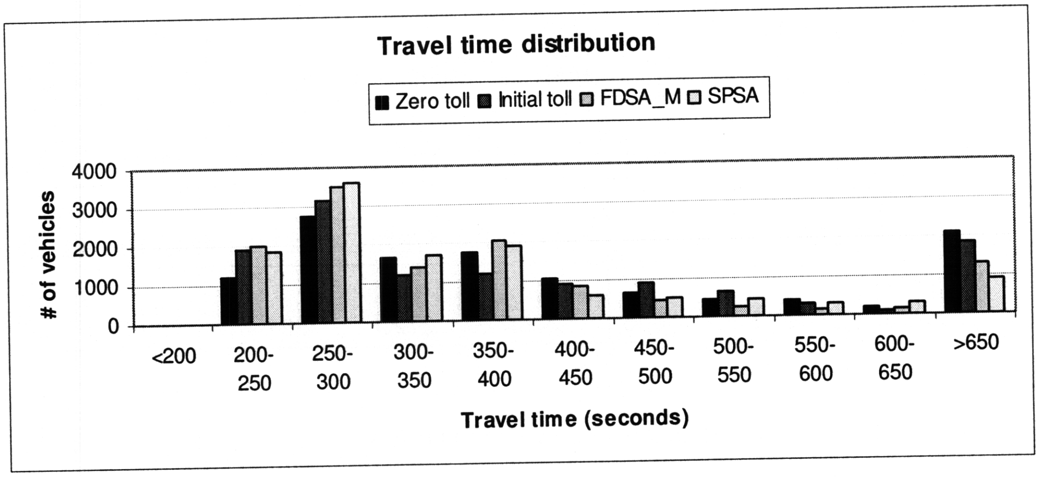


Figure 5.14: Travel time distribution among four scenarios

In this figure, it is observed that in two toll strategies, the travel time distribution is shifted to the left, and the shift is greater in SPSA than FDSA-M. This observation enhances our conclusion that congestion pricing improves the network condition, and the toll gotten by SPSA is the best.

The second comparison is the effect of congestion pricing to departure time choice, which is depicted in Figure 5.15. It is observed that in SPSA and FDSA-M scenarios, travelers have a wider distribution in departure time. Between them, SPSA shows better result. This comparison depicts the effectiveness in terms of "peak spreading", which is, how well does the congestion pricing move the traffic from peak hours to other time intervals.

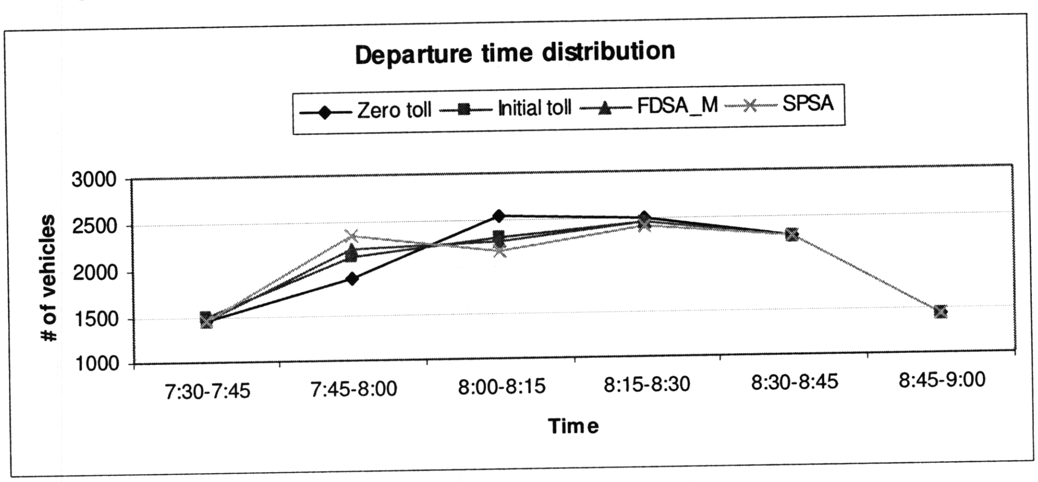


Figure 5.15: Departure time distribution for three toll strategies

The third comparison is the effect of congestion pricing to the route choice. In this section, the effectiveness of route choice is discussed by comparing the number of vehicles traveling through link 3 and link 4 in the simulation period. Because link 3 and link 4 are the only corridors to the destinations, every passenger should choose his path either through link 3 or link 4 based on the travel time and the cost. Thus, the effectiveness of the route choice can be reflected by the balance between the two links.

In this thesis, DynaMIT updates the simulation every minute. Thus, the estimation of number of vehicles in each link is also available in one minute length. Therefore, during the simulation period for each link, there are 90 estimation of number of vehicles in total, each for the estimation in one minute period. Among all these estimations, the balance between link 3 and link 4 are compared in Table 5.8.

Table 5.8: Density difference between link 3 and link 4 (in vehicle/foot)

Density difference between link 3 and 4	Zero toll	Initial toll	FDSA-M	SPSA
<0.1	2	2	2	5
0.1-0.2	0	0	4	6
0.2-0.3	2	0	5	3
0.3-0.4	9	1	7	3
0.4-0.5	42	53	57	58
0.5-0.6	21	21	11	10
0.6-0.7	14	13	4	5
>0.7	0	0	0	0

It is observed that in the zero toll and the initial toll strategy, the density is quite different between link 3 and link 4, which means, the passengers always travel through one link while another is empty, causing significant congestions. On the contrary, in SPSA and FDSA-M strategy, the density is much more balanced. The results demonstrate a significant impact of congestion pricing on route choice.

5.6 Summary

In this chapter, the fixed point problem and the corresponding algorithms are discussed first. Through the tests on a synthetic network, MSA is selected as the best one to apply as the time convergence algorithm. Then in the congestion pricing testes, several optimization algorithms are applied and tested in the synthetic network. SPSA with twice gradient calculation is selected as the best one based on its performance. FDSA-M, a modified version of FDSA, is also proved to be useful by decrease the total travel time from the initial toll scenario. The usefulness and effectiveness are shown based on the comparison among different toll scenarios at last. Based on the detailed evaluation, SPSA is selected for application in the real case study in the next chapter.

6. Case Study: Lower Westchester County

Previous chapters focused on the development of a dynamic congestion pricing model and the associated algorithm selection. Different algorithms have been discussed and tested in this regard using a synthetic network. Among all these algorithms, SPSA has been identified as the best one considering in terms of effectiveness. In this chapter, SPSA is applied in a real network – Lower Westchester County in New York State. The objectives of the case study are discussed first. The design of the case study is then described in detail. The results are analyzed and discussed next. A summary is provided at the end.

6.1 Objectives

The objectives of the case study are:

- To examine the usefulness and effectiveness of dynamic congestion pricing in the real network, this is rarely performed in previous research.
- To analyze the application of the dynamic congestion pricing model in Lower Westchester County network.
- To test the performance of SPSA in the complicated network.

6.2 Case study description

The case study is performed in the same network as Vaze (2007) and Rathi (2007). The network is located in Lower Westchester County (LWC) in New York State. The data, including network topology and characteristics of the passengers, are obtained from New

York State Department of Transportation (NYSDOT).

6.2.1 Experiment design

LWC is situated to the north of New York City. Many important highways are located in this area to connect the Connecticut region and Manhattan area in New York City. The network is extremely congested, especially in peak hours of weekdays when the commuter trip occur. This case study focuses on freeways and parkways, including I-95 (the New England Thruway), I-87 (the New York State Thruway) and I-287 (the Cross Westchester Expressway). While I-287 is the major east-west corridor, most of the freeways are in north-south direction.

6.2.1.1 Network topology

The research network consists of 1767 links and 825 nodes. Each link is further divided into multiple segments, depending on the characteristics of the link. The segment may contain different number of lanes, from 1 to 5. The number of lanes is the key factor to determine the capacity of the segment. Generally, the freeways and parkways contain two or three lanes in each direction. The simulation period is chosen at the peak hour in the morning, from 7:00 am to 9:00 am. The simulation period is one hour longer than that (7:00 am to 10:00 am) in order to let all vehicles to finish their trips. The estimation length is 15 minutes, which is also the interval length. The simulation is updated every minute. The network is presented in Figure 6.1.

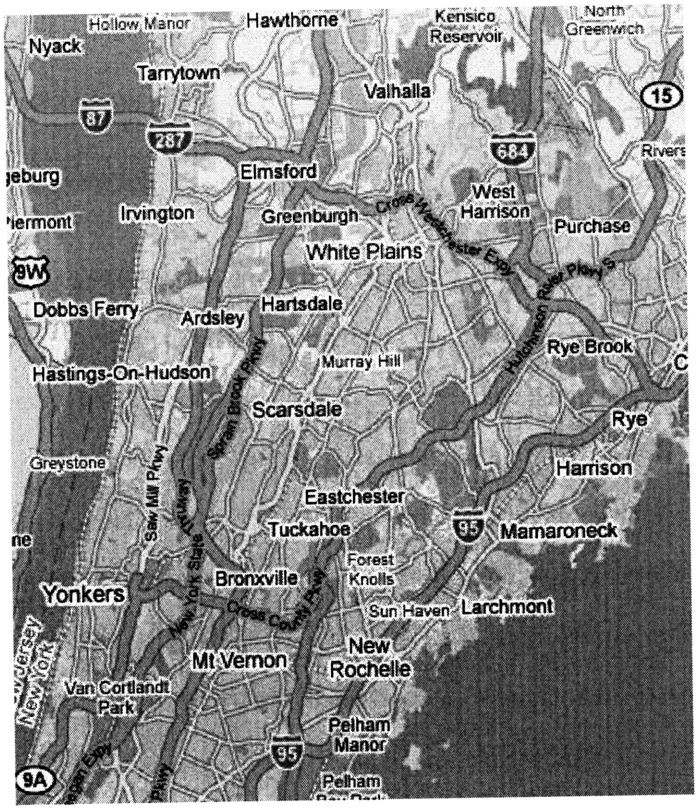


Figure 6.1: Network of LWC*

6.2.1.2 OD flow information

There are 482 OD pairs assigned on the network. Similar as the synthetic case study, half of travelers are assumed to be available to receive the real time toll information. The demand information is calibrated using DynaMIT with a random perturbation that is uniformly distributed between 0.7 and 1.1. See Antoniou et al. (2006) and Balakrishna et al. (2007) for demand calibration. The possible paths for each OD pair are searched directly by DynaMIT.

6.2.1.3 Toll plaza information

In order to determine the locations of the toll plazas, the network is simulated in MITSIMLab (Yang and Koutsopoulos, 1996), a microscopic transportation simulation software developed in MIT ITS lab. In contrast with DynaMIT, MITSIMLab aims to simulate each driver's

---

* Source: http://maps.google.com

behavior depending on the network conditions, e.g., route choice, lane changing, and acceleration. For the detailed description of MITSIMLab, see Yang and Koutsopoulos (1996) and Yang et al. (2000).

The simulation period in MITSIMLab without toll in the same as DynaMIT, from 7:00 am to 10:00 am. The network condition in 8:00 am is depicted in Figure 6.2, in which red color is the most congested area, blue color represents uncongested road, and green color represents intermediate level of congestion. From the simulation of the network condition, ten toll plazas are set on the most congested links, shown in Figure 6.2 as well. Notice that the tolls for two directions may be different, so there are two toll booths for each toll location. The historical toll is set by 0 in all time periods. This is different with the synthetic case study, because in the real case, it is hard to "guess" a good start toll. Thus, a zero toll scenario is always acceptable if no previous knowledge available.

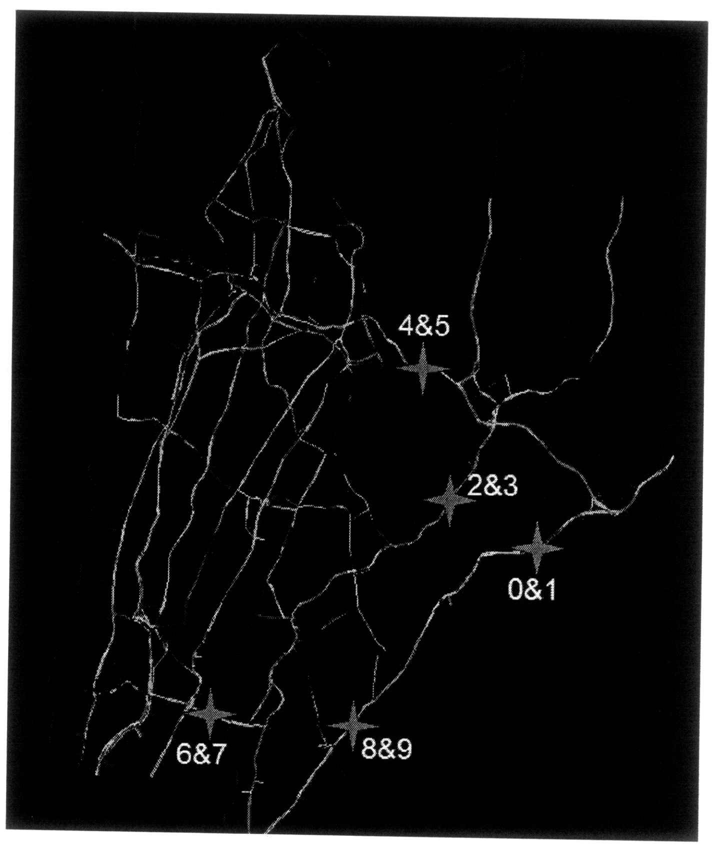


Figure 6.2: Locations of the toll plazas

6.2.2 Parameter calculation

The calculation of parameters for LWC network is similar as the process for the synthetic network. The results are:

- Parameter for time in the route choice utility function β_2 = -0.008;
- Parameter for the cost in route choice β_1 = -0.0192;
- Parameter for the early arrival in departure time choice β_3 = -0.0024;
- Parameter for the early arrival in departure time choice β_4 = -0.020;
- Constant for habitual departure time in departure time choice α^0 = 10;
- Parameter for the cost in departure time choice β_1' = -4.4;

6.3 Time convergence algorithm test

In Chapter 5, MSA is selected as the best fixed point algorithm for the time convergence in the synthetic case study. Before performing the tests in the real network, the usefulness and effectiveness of MSA should be tested again in the real network. Similar as Chapter 5, a simple scenario without toll is tested from the free flow link travel time. Then, using the convergence result as the starting point, two random toll scenarios are tests as well. The results are depicted in Figure 6.3.

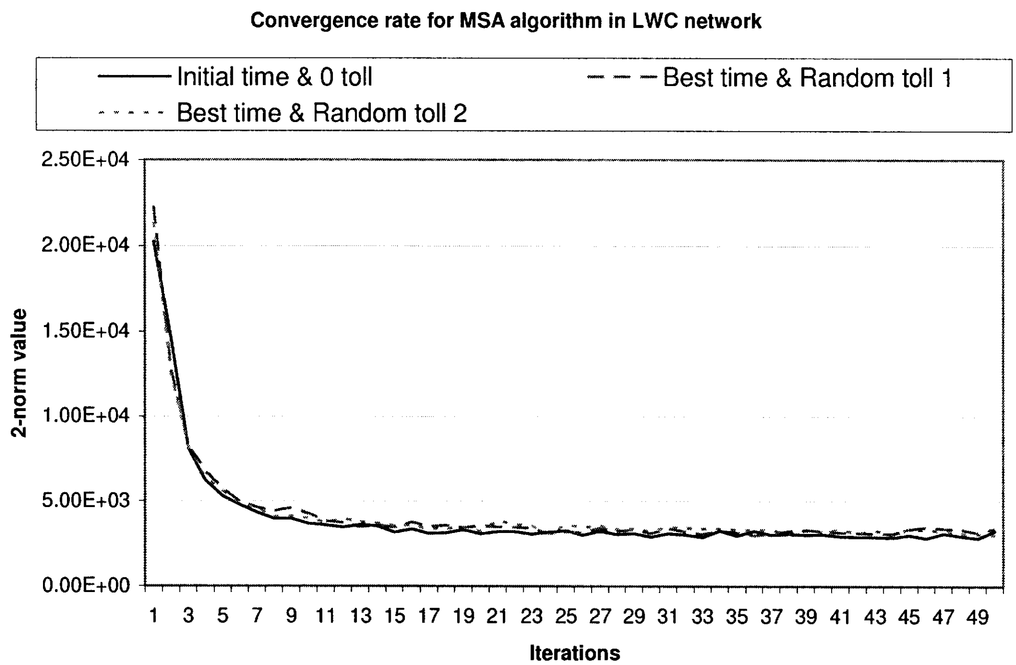


Figure 6.3: Results of MSA algorithm in LWC network

From the results, it is observed that MSA works perfectly in all scenarios in the real network. The results converge within 15 iterations, and are acceptable only in 5 iterations. Based on the result and the running time constraints, in the real case study, only 5 iterations are simulated to reach equilibrium.

6.4 Result analysis

SPSA algorithm is applied in LWC network based on the previous discussion. The maximum toll is $6.00, and the minimum is 0. There are 12 periods in total in the simulation period from 7:00 am to 10:00 am. Because of the heavy running burden and the convergence results in Section 6.3, only 5 iterations are simulated to reach equilibrium in each run of DynaMIT. From the results of Chapter 5, two calculations are performed to get the gradient vector. The total number of simulation is set to 20.

Although the simulation time has been extended by one hour, because of the large size of the network, there may still be some vehicles cannot finish their trips. In this case, the average travel time is used instead of the total travel time as the objective function. Notice that for inelastic demand, these two objective functions are exactly the same.

6.4.1 Case study result

The running result of SPSA is depicted in Figure 6.4 and Table 6.1.

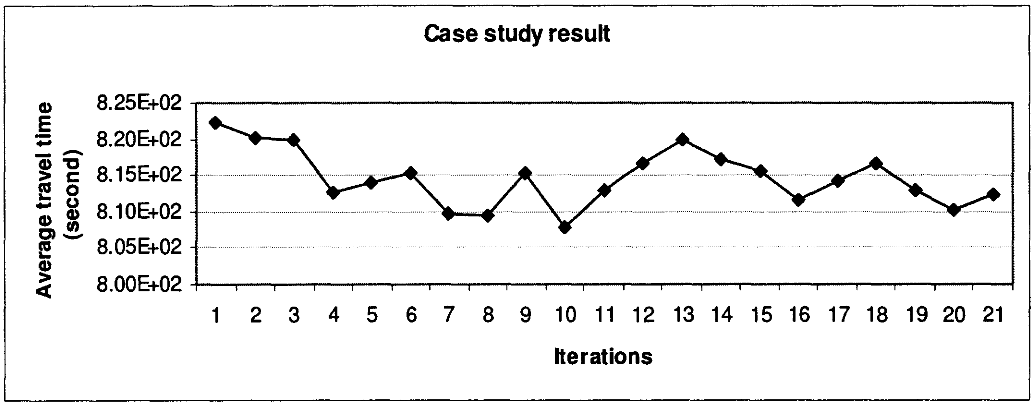


Figure 6.4: Result of FDSA-M application in LWC network

Table 6.1: Best toll strategy for LWC network (in cents)

Toll Plaza	Time			
	7:00	7:15	7:30	7:45
0	70	90	3	0
1	58	16	113	44
2	29	0	0	10
3	19	24	13	49
4	35	29	0	36
5	37	19	25	0
6	37	7	69	23
7	90	32	26	1
8	21	29	33	0
9	71	15	126	70
	8:00	8:15	8:30	8:45
0	75	3	37	40
1	0	27	38	47

2	3	22	38	9
3	0	68	34	0
4	13	73	82	75
5	38	0	23	14
6	0	68	64	39
7	46	4	0	0
8	44	0	28	69
9	3	61	73	0

It is observed that the algorithm does not converge, because the number of iterations is small. However, the total travel time tends to decrease except for some fluctuations. This observation proves the effectiveness of SPSA in the dynamic congestion pricing model.

Notice that there is no guarantee that the final toll is optimal.

6.3.2 Usefulness and effectiveness of dynamic congestion pricing

In this section the usefulness and effectiveness of dynamic congestion pricing from the test result is analyzed. Two scenarios are compared: zero toll scenario and best toll scenario obtained from previous sections (Table 6.1). Notice that in the real case study, the initial toll is set by zero in all time periods.

The basic performance of these two scenarios is compared in Table 6.2. From the comparison, it is observed that the network condition is improved under the best toll scenario. However, the magnitude of this improvement is rather small compared to the synthetic example. That is, in the real network, even the "best" congestion pricing strategy cannot substantially improve the network condition. In the "zero toll" base case, the average travel time for a commuter passenger is 820 seconds, about 13.7 minutes. When improving the network condition by the best toll, the average travel time became 808 seconds, about 12 seconds, or 1.5% saving from the base case scenario.

Table 6.2: Comparison among three scenarios

	Zero toll	Best toll
Number of vehicles	141385	141385
Total travel time (hour)	32211	31721
Average travel time (sec)	820	808

It should be noticed that one of the important reasons that the dynamic congestion pricing performed poorer in the LWC network than in the synthetic network is the network topology. When building the LWC network, all small branches are eliminated to simplify the network. However, these small roads are critical for congestion pricing scenario, because while there is a toll in a road, the travelers should have sufficient alternatives to switch. However, in the current network, the traveler may choose a much longer path to avoid the toll than in realistic. This increases the travel time of the travelers substantially. In future research, a more detailed network may improve the performance more of dynamic congestion pricing.

Although the improvement in the LWC network is not as great as the synthetic network, similar conclusions are made through the similar tests.

The first test is the travel time distribution among all passengers, which is also illustrated in Figure 6.5. From the comparison among the two scenarios, it is observed that in the best toll strategies, the travel time distribution is shifted to the left, reflecting the travel time saving and demonstrating the usefulness of the congestion pricing again.

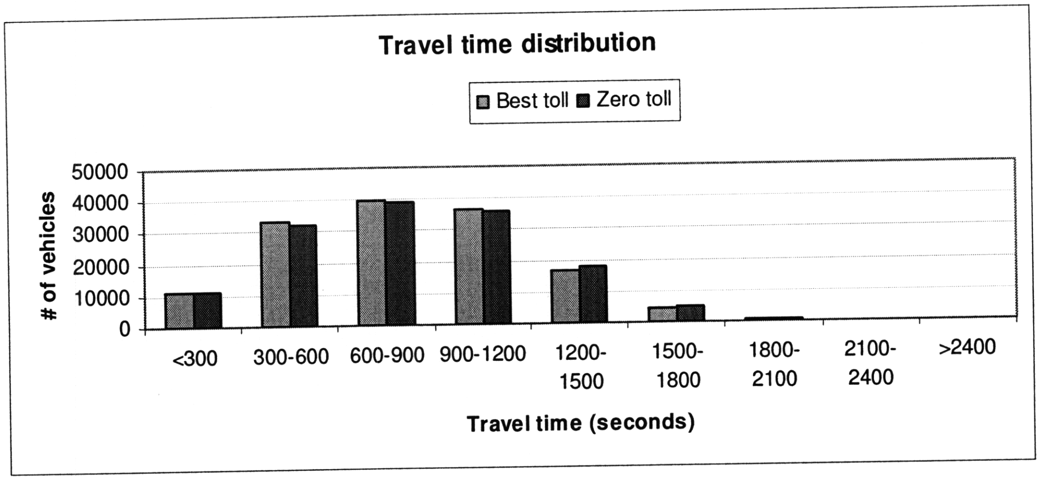


Figure 6.5: Travel time distribution in three scenarios

Another test is the departure time distribution. In this real network, the number of vehicles departing from each time interval is quite flat. Thus, instead of comparing the departure time distribution, we examine the time-vehicle relationship, which is, how many vehicles are on the network in each time period (every 15 minutes), depicting in Figure 6.6. Notice that in the best toll strategy, the peak hour is moved forward so that the number of vehicles on the network has flatter pattern than the zero toll scenario.

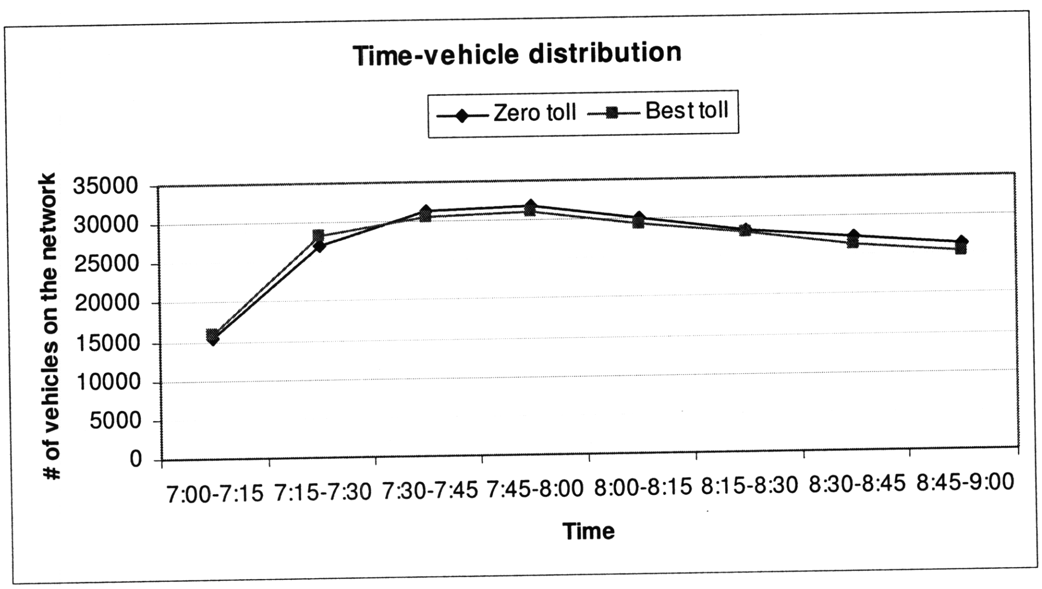


Figure 6.6: Time-vehicle distribution among three scenarios

6.4 Summary

In this chapter, the congestion pricing model is applied in the LWC network using SPSA algorithm. The results show that the algorithm works well in the real network and improve the network condition only after 20 iterations. The usefulness and effectiveness of dynamic congestion pricing is also discussed from the comparison among two scenarios: zero toll (initial toll) and best toll obtained from the case study. The comparison shows that by applying the best toll strategy: (1) both the total travel time and the average travel time on the network decrease; (2) the network flow has a flatter pattern by moving part of the peak hours flow to the non-peak hours. These results are coincident with the ones in the synthetic case study, and demonstrate the effect of dynamic congestion pricing.

Although the usefulness is demonstrated by the test on the LWC network, there are several limitations that need to be improved in further research. First, several parameters in this thesis are cited from literature rather than obtained from estimation and calibration. The cited values are reasonable in some sense; however, it is criticized that the values should be specified for the study area, i.e., the LWC network in this thesis. Thus, although the results are reasonable in some sense, they may not be the most realistic ones. If the congestion pricing data is available in the future, all these parameters should be estimated by appropriate datasets and calibrated in DynaMIT.

Second, in this thesis, the dynamic congestion pricing model is tested in DynaMIT based on SPSA. Although it performs excellent in the synthetic network by decreasing the average travel time by 21%, in the LWC network, the decrease is only 2%. One potential reason is that many small branches are eliminated so that the travelers do not have many alternatives to choose. The effectiveness of dynamic congestion pricing may be demonstrated better by a more detailed network.

7. Conclusion

This thesis focuses on modeling and testing of dynamic congestion pricing. After reviewing previous research in congestion pricing, a model for dynamic congestion pricing has been developed with the objective function of minimizing the total travel time on the network as a non-linear optimization problem. The optimization algorithms for the problem have then been discussed. Several algorithms have been selected from the literature review, such as, SPSA, FDSA, etc. From tests in a synthetic network, SPSA with two calculations for the gradient vector has been selected as the best algorithm and is applied in a real case study in LWC network in New York State. From both case studies, the usefulness and effectiveness of dynamic congestion pricing has been demonstrated.

7.1 Summary

The focus of this thesis is a simulation model to improve the drawbacks from the previous research. Previous research on congestion pricing has first been reviewed in this regard. These studies have been classified into two categories, either on a specific model such as route choice model and departure time choice model, or on a general structure of congestion pricing. In the specific model research, route choice and departure time choice have been reviewed in detail. Some models have been implemented in this thesis, such as path-size logit model for the route choice. The general structure research can be sub-divided into two categories: analysis model which focuses on the theoretical formulation and simulation model which concentrates on the practical application. Since analysis model is only useful theoretically, simulation model is the focus of the current congestion pricing research. The drawbacks of the current research on simulation model have been concluded to have three aspects: first, traffic condition is considered based on network equilibrium rather than travelers' choice behavior; second, the optimization algorithm is usually not efficient; third, case study on real network is not done in most cases.

A simulation model has then been formulated as a non-linear optimization problem. Because the demand is assumed to be inelastic, the objective function is chosen by minimizing the total travel time among all users. The constraints include demand side, supply side, network topology side and traveler choice side. DynaMIT software has been used as the tool for simulation. Because demand, supply and network topology constraints have been already captured in the process of DynaMIT, only traveler choice is discussed in detail. In this thesis, both route choice and departure time choice have been considered.

The solution algorithms have been discussed after that. By reviewing previous research, SPSA and FDSA have been selected to solve the dynamic congestion pricing model. DynaMIT has been used as the simulation tool. In order to get the equilibrium travel time, several fixed point algorithms have been discussed, and MSA has been selected as the best one in this thesis. A test has then been performed in a synthetic network for different optimization algorithms. From the test, SPSA with two calculations of the gradient vector has been chosen as the best algorithm. The usefulness and effectiveness of congestion pricing has also been examined and discussed by comparing different toll scenarios. It has been found that by applying optimal toll, the network improved.

The model has then been applied in a real case study in LWC network. SPSA algorithm has been found to work well in the real network by reducing the total travel time in just 20 running iterations. Through several comparisons of different toll scenarios, congestion pricing has been demonstrated as an effective approach to reduce congestion.

7.2 Contributions

The major contributions of this thesis are follows:

- Developing a comprehensive dynamic congestion pricing model based on discrete choice theory, in which both route choice and departure time choice are considered.
- Considering travelers' choice behavior rather than network equilibrium in the network condition calculation for dynamic congestion pricing simulation. Though application of traveler's choice behavior in congestion pricing research is not new, according to the published literature, this is the first application of its integration in simulating dynamic congestion pricing scenarios.
- Enhancing the capability of the DynaMIT software to simulate congestion pricing. Traveler's behavior choice models, such as departure time choice, have also been improved in this regard.
- Demonstrating the usefulness and effectiveness of dynamic congestion pricing in the real network by performing a case study on the LWC network in New York State. It may be noted that such application of dynamic congestion pricing in the real network is rare.

7.3 Future research directions

The effectiveness and practical realism of the model and algorithm for dynamic congestion pricing have been demonstrated in this thesis. However, it still has some limitations. Some directions for future research are as follows:

- Estimation and calibration of the parameters in the congestion pricing model: The parameters for congestion pricing model are critical. Because of insufficient data, some parameters have been cited from other research. However, as discussed in Section 6.5, this is not the best approach since there are several limitations. Therefore, if the congestion pricing data are available in the future, estimation and calibration should be performed to obtain the value of the parameters.
- Better optimization algorithm: Although SPSA has been proved to be both useful and efficient, it cannot satisfy the on-line running requirements for the dynamic congestion

pricing model. In order to run the model in real time, better algorithms are necessary.

- Different objective functions: The objective function in this research is to minimize the total travel time by all users. However, there are several other candidate objective functions, such as: maximizing consumer surplus; maximizing social welfare; minimizing the maximum travel time; maximizing total toll revenue; etc., which are more appropriate in some applications. Therefore, the models may be expanded to consider other objective functions in the future.
- Departure time choice model improvement: In this research, departure time choice has been considered by both the time and the cost. However, current data cannot support a detailed examination on the structure of the departure time model. Since departure time choice is still immature in the congestion pricing research, it deserves more attention in future studies.
- Cancelling trip choice and mode choice: This research only considers private cars, and no cancelling trip is allowed. In order to simulate the congestion pricing more realistically, a refined model is needed for the mode choice and the cancelling trip choice.
- Detailed network topology: The performance of dynamic congestion pricing in the LWC network is poorer than the synthetic case study. One important reason is that in the LWC network, many small branches are eliminated so that the travelers do not have many alternative choices. In order to demonstrate the effectiveness of dynamic congestion pricing better, a more detailed network is recommended.
- Other toll collection approaches: In this research, the toll is only collected on links. However, the tolls can also be charged in a cordon, by a single facility, or by parking. In future research, these toll collection approaches should also be considered in the dynamic congestion pricing model and the solution algorithm.

Appendix A: MATLAB Code for SPSA

SPSA function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%
SPSA algorithm for Dynamic Congestion Pricing
%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Shunan Xu, Jan. 2009
%This code implements SPSA with constraints for dynamic congestion pricing.

clear all

%%%%%%%%%%%
Global variables
%%%%%%%%%%%

global no_intervals       % # Time intervals
global start_time         % starting time of estimation interval
global main_dir
global cost_init          % initial ODs
global no_prev_intervals
global no_tollplazas      % # of tollplazas
global no_costs           % # of costs = # of tollplazas * # of times
global total_time_pack

%%%%%%%%%%%%%%%%%%%%%%%
Filenames
%%%%%%%%%%%%%%%%%%%%%%%

initial_fn_value_file = 'spsa_output/fn_theta_0.dat';
delta_file = 'spsa_output/delta.dat';
scaled_delta_file = 'spsa_output/delta_scaled.dat';
gradient_log_file = 'spsa_output/grad_log.dat';
final_obj_fn_file = 'spsa_output/final_fn.dat';
final_theta_file = 'spsa_output/x_final.dat';
iter_theta_file = 'spsa_output/x_values.dat';
```

```
iter_obj_fn_file = 'spsa_output/fn_values.dat';
fn_path_file = 'spsa_output/fn_path.dat';
rmsn_file = 'spsa_output/best/rmsn_best.dat';
best_theta_file = 'spsa_output/best/best_od.dat';
main_dir = '/home/xushunan/DTA/';
best_snob_file = 'spsa_output/best_snob.dat';
best_theta_file = 'spsa_output/best_theta.dat';
snob_change_file = 'spsa_output/snob_change.dat';

%%%%%%%%%%%%%%%%%%%%%%
Variable initialization
%%%%%%%%%%%%%%%%%%%%%%

total_time_pack = [];
start_time = 25200;          % 7:00 am
no_intervals = 12;           % 7:00 am to 10:00 am
no_tollplazas = 10;

no_costs = no_tollplazas*(no_intervals*15+30);

cd(main_dir)

%%%%%%%%%%%%%%%%%%%%%%
Read original costs
%%%%%%%%%%%%%%%%%%%%%%

system(['rm temp.dat']);
cost_seed_temp = load('lwc/cost_seed.dat');
cost_seed = [];

for ii=1:no_tollplazas
    for jj=1:(no_intervals*15+30)
        cost_seed = [cost_seed; cost_seed_temp(ii+(jj-1)*no_tollplazas)];
    end
end
cost_seed = cost_seed';
cost_init = cost_seed;
size(cost_seed)

theta_0 = cost_seed(1:no_costs);

%%%%%%%%%%%%%%%%%%%%%%
Read simulation parameters
%%%%%%%%%%%%%%%%%%%%%%
```

```
n = 40                  %total no. of loss measurements
n_time = 15;            %assume in n_time minutes the toll is unchange

cases = 1;
grad_reps = 2;              % no. of reps for averaging the gradient approximation
alpha =.602;
gamma =.101;
a = 1400;
c = 30;                     % c is used to control the step size of search
A = 10;                     % A is used to control the speed of decrease

lossfinalsq=0;              %variable for cum.(over 'cases')squared loss values
lossfinal=0;                %variable for cum. loss values

%low bound is 0, and upper bound is 600, means $6.
theta_lo = 0*theta_0;
theta_hi = 600 + theta_lo;

rand('seed',31415927)
randn('seed',3111113)

%%%%%%%%%%%%%%%%%%%%%%
Begin simulation
%%%%%%%%%%%%%%%%%%%%%%

cd mfiles;
    initial_fn_value = snob_func(theta_0); % snob_func: calculate the average travel time
cd ..;
best_snob = initial_fn_value;
best_theta = theta_0;

system(['rm lwc/tollplaza_iter_0.dat']);
system(['cp lwc/tollplaza.dat lwc/tollplaza_iter_0.dat']);

tmp_x = theta_0';
tmp_fn = initial_fn_value;

eval(['save ', initial_fn_value_file, ' tmp_fn -ascii;']);

x_values = [];
fn_values = [];
fn_path = initial_fn_value;
grad_log = [];
```

```
cont = 'n';

for i=1:cases

   theta=theta_0;                      % Start at seed parameter values

   best_snob = initial_fn_value;
   best_theta = theta_0;

   prev_path_y = 0;
   prev_theta = best_theta;
   prev_path_path = [];
   best_path_path = [];

   for k=0+0:n-1+0
      ak = a/(k+1+A)^alpha;
      ck = c/(k+1)^gamma;
      ghat = 0;                        % store sum of gradient reps for averaging
      tmp_fn_grad_reps = [];           % Store the fn values for averaging

      for xx = 1:grad_reps,
            delta = []
            num_loops = no_costs/n_time;
            %   generate delta
           for yy = 1:num_loops,
                  ran_num = 2*round(rand(p,1))-1;
                  for zz = 1:n_time,
                        delta = [delta; ran_num];
                  end
            end %yy loop end

         eval(['save ', delta_file, ' delta -ascii;']);
         delta = delta' ;
         eval(['save ', scaled_delta_file, ' delta -ascii;']);

         thetaplus = theta + ck*delta;
         thetaminus = theta - ck*delta;

         % These four lines below invoke component-wise constraints
         thetaplus=min(thetaplus,theta_hi);
         thetaplus=max(thetaplus,theta_lo);
         thetaminus=min(thetaminus,theta_hi);
         thetaminus=max(thetaminus,theta_lo);
```

```
         x_values = [x_values; thetaminus' tmp_x thetaplus'];
         tmp_x = theta'; % reset to theta for next grad rep

cd mfiles/;
         yplus=snob_func(thetaplus)
         yminus=snob_func(thetaminus)
cd ..;

%    Store the two fn values

        tmp_fn_grad_reps = [tmp_fn_grad_reps; yminus yplus];
        ghat_temp = [];

        for zz = 1:no_costs,%generate delta
            ghat_temp = [ghat_temp; ((yplus - yminus)/(2*ck*delta(zz)))];
        end %generate delta

        ghat_temp
        ghat = ghat + ghat_temp';

    end % end of reps for grad averaging

    aa = mean(tmp_fn_grad_reps(:,1));
    bb = mean(tmp_fn_grad_reps(:,2));

    fn_values = [fn_values; aa bb];

    ghat = ghat/grad_reps;
    grad_log = [grad_log; ghat'];
    eval(['save ', gradient_log_file, ' grad_log -ascii;']);

    theta=theta-ak*ghat

    % Two lines below invoke component-wise constraints
    theta=min(theta,theta_hi);
    theta=max(theta,theta_lo);

cd mfiles/;
    path_y = snob_func(theta);
cd ..;
    system(['rm lwc/tollplaza_iter_',int2str(i),'_',int2str(k+1),'.dat']);
    system(['cp lwc/tollplaza.dat lwc/tollplaza_iter_',int2str(i),'_',int2str(k+1),'.dat']);

    if path_y < best_snob
```

```
            best_theta = theta;
            best_snob = path_y;
            eval(['save ', best_snob_file,' best_snob -ascii;']);
            eval(['save ', best_theta_file,' best_theta -ascii;']);
    end % if

    % If the current result is greater than previous value / 0.9, return to the previous one
    if ((k > 0) && ((path_y*0.9) > prev_path_y))
        prev_path_path = [prev_path_path; k path_y prev_path_y];
        path_y = prev_path_y;
        theta = prev_theta;
    end % if

    prev_path_y = path_y;
    prev_theta = theta;

    % If the current result is greater than best value / 0.7, return to the best one
    if ((k > 0) && ((path_y*0.7) > best_snob))
        best_path_path = [best_path_path; k path_y best_snob];
        path_y = best_snob;
        theta = best_theta;
    end % if

    fn_path = [fn_path; path_y];

    eval(['save ', fn_path_file, ' fn_path -ascii;']);
    eval(['save ' , snob_change_file, ' prev_path_path    -ascii;' ]);
    eval(['save ' , snob_change_file, ' best_path_path    -ascii;' ]);

    tmp_x = theta';

    eval(['save ', iter_theta_file, ' x_values -ascii;']);
    eval(['save ', iter_obj_fn_file, ' fn_values -ascii;']);

  end % iterations (k = 0:n-1)

% save current fn values (at theta)

  eval(['save ', final_theta_file, ' theta -ascii;']);

  lossvalue = path_y;
  lossfinalsq=lossfinalsq+lossvalue^2;
  lossfinal=lossfinal+lossvalue;
```

```
end % replications (i = 1:cases)

% Display results: Mean loss value and standard deviation

disp(['mean loss value over ', int2str(cases), ' runs']);
final_fn = lossfinal/cases;

eval(['save ', final_obj_fn_file, ' final_fn -ascii;']);
eval(['save ', best_theta_file, ' best_theta -ascii;']);

end

cd mfiles/;
```

SNOB_fun function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
This function is used to call DynaMIT to calculate the objective function value
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function y = snob_func(X)

%%%%%%%%%%%%
Global variables
%%%%%%%%%%%%

global no_intervals         % # Time intervals
global start_time           % starting time of estimation
global main_dir
global cost_init            % initial costs
global no_tollplazas        % # of tollplazas
global no_costs             % # costs = # tollplazas * # times (interval*interval lenth)
global total_time_pack

%%%%%%%%%%%%
Filenames
%%%%%%%%%%%%

cd(main_dir);
current_iter_counter_file = 'spsa_output/avg_iter.dat';
obj_fn_value_file = 'spsa_output/bigsum.dat';
```

```
TotalTime_CAL = 'ccfiles/totalTravelTime';
travel_time_file = 'lwc/tmp/est_packettime1.out';
total_time_file = 'spsa_output/totaltime.dat';
time_pack_file = 'spsa_output/time_pack.dat';

n_iterations = 1; % no. of runs over which averging is done

format long;
cd(main_dir);

y = 0;
bigsum = 0;
parameter_set = X;

%%%%%%%%%%
Update tollplaza
%%%%%%%%%%

save temp.dat parameter_set -ascii;
cd mfiles;
    CHANGEPARAMETER(parameter_set); % CHANGEPARAMETER: update tollplaza
cd ..;

%%%%%%%%%%
Run DynaMIT
%%%%%%%%%%

for counter = 1:n_iterations

    eval(['save ', current_iter_counter_file, ' counter -ascii']);

    cd('lwc/');

    system('rm linktime.dat');
    system('cp linktime_original.dat linktime.dat');

    system('./cleanup');
    system('./DynaMIT_P dtaparam_P.dat > zzz');

    cd ..;

    % calculate total travel time
    eval(['!', TotalTime_CAL,...
        ' ', travel_time_file,...
```

```
            ' ', total_time_file]);

        times = load(total_time_file);
        total_time = times(1)
        total_pack = times(2)

        compare = [total_time total_pack]
        save compare.out compare -ascii;

    % Add to total
        bigsum = bigsum + total_time;

end

bigsum = bigsum/n_iterations;

bigsum = bigsum/total_pack;

if (isnan(bigsum))
    bigsum = 1000000000000000;
end

%%%%%%%%%%
Save and return
%%%%%%%%%%

eval(['save ', obj_fn_value_file, ' bigsum -ascii'])

total_time_pack = [total_time_pack; total_time bigsum total_pack];
eval(['save ', time_pack_file, ' total_time_pack -ascii;']);

cd mfiles;

y = bigsum;
```

CHANGEPARAMETER function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
This function is used to reset tollplaza based on the result of the previous iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function CHANGEPARAMETER(Y)

%%%%%%%%%%%
Global variables
%%%%%%%%%%%

global no_intervals          % # Time intervals
global start_time            % starting time of estimation
global main_dir
global cost_init             % initial costs
global no_tollplazas         % # of tollplazas
global no_costs              % # costs = # tollplazas * # times (interval*interval lenth)

%%%%%%%%%%%
Filenames
%%%%%%%%%%%

ResetToll_EXEC = 'ccfiles/resetTollplaza';
cost_file = 'temp.dat';
tollplaza_file = 'shunan_test/tollplaza.dat';
new_tollplaza_file = 'fdsa_output/new_tollplaza.dat';

%%%%%%%%%%%%%%%%%%%%%%%%%%
Call C++ program to change tollplaza data
%%%%%%%%%%%%%%%%%%%%%%%%%%

cd ..;
    eval(['!', ResetToll_EXEC,...
        ' ', cost_file,...
        ' ', tollplaza_file,...
        ' ', new_tollplaza_file,...
        ' ', int2str(no_tollplazas)]);
    system(['rm shunan_test/tollplaza.dat']);
    system(['mv spsa_output/new_tollplaza.dat shunan_test/tollplaza.dat']);
cd mfiles/;
```

References

1. Akiyama, T. and Tsuboi, H. 1996, 'Description of route choice behavior by multi-stage fuzzy reasoning', in *Highways to the Next Century Conference*, Hong Kong.
2. Akiyama, T. and Yamanishi, H. 1993, 'Travel time information service device based on fuzzy sets theory', in Ayyub, B.M. (eds.), Proceedings of ISUMA 93, *The Second International Symposium on Uncertainty Modeling and Analysis*, IEEE Computer Press, College Park, Maryland, pp. 238-245.
3. Alberta, G. and Mahalel, D. 2006, 'Congestion tolls and parking fees: a comparison of the potential effect on travel behavior', *Transport Policy*, vol. 13, pp. 496-502.
4. Algers, S., Bergstrom, P., Dahlberg, M. and Dillen, J.L. 1998, 'Mixed logit estimation of the value of travel time', *Working Paper Series*, Department of Economics, Uppsala University, US.
5. Antoniou C. 1997, 'Demand simulation for dynamic traffic assignment', Master Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.
6. Antoniou, C., Wen, Y., Vaze, V. and Rathi, V. 2006, 'Development of ITS analysis tolls for Lower Westchester County. Technical memorandum on the calibration and validation of DynaMIT in closed loop with MITSIMLab for the Lower Westchester County network', Submitted to New York State Department of Transportation.
7. Balakrishna, R. 2006, 'Off-line calibration of dynamic traffic assignment models', PhD Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.
8. Balakrishna, R., Antoniou, C., Ben-Akiva, M. E., Koutsopoulos, H. N. and Wen, Y. 2007, 'Calibration of microscopic traffic simulation models: methods and applications', In *Transportation Research Record: Journal of the Transportation Research Board, No. 1999*, Transportation Research Board of the National Academics, Washington, D.C., pp.

198-207.

9. Balakrishna, R., Sundaram, S., Wen, Y., Xu, S. and Diwan, S. 2008, 'Development of a dynamic traffic assignment system for real-time and planning purposes. DynaMIT system user guide – version 2.0', Intelligent Transportation Systems Program, Massachusetts Institute of Technology, Cambridge, Massachusetts.
10. Bekhor S., Ben-Akiva, M. and Ramming, M.S. 2002, 'Adaptation of logit kernel to route choice situation', *Transportation Research Board*, vol. 1805, pp. 78-85.
11. Ben-Akiva, M. and Bierlaire, M. 1999, 'Discrete choice methods and their applications to short-term travel decisions', in Hall, R.W. (eds.), *Handbook of Transportation Science*, Kluwer Academic, Dordrecht.
12. Ben-Akiva, M. E. and Lerman, S. R. 1985, *Discrete choice analysis: theory and application to travel demand,* MIT Press, Cambridge, Massachusetts.
13. Black, I.G. and Towriss, J.G. 1993, *Demand effects of travel time reliability,* Centre for Logistics and Transportation, Cranfield Institute of Technology.
14. Borjesson, M. 2008, 'Joint RP-SP data in a mixed logit analysis of trip timing decisions', forthcoming in *Transportation Research,* Part E.
15. Bottom, J. 2000, 'Consistent anticipatory route guidance', PhD Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.
16. Box, M. J. 1965, 'A new method of constrained optimization and a comparison with other methods', *Computer Journal*, vol. 8, no. 1, pp. 42-52.
17. Calfee, J. and Winston, C. 1998, 'The value of automobile travel time: implications for congestion policy', *Journal of Public Economics*, vol. 69, 83-102.
18. Cascetta, E. 1996, 'A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks', Proceedings of *the Thirteenth International Symposium on Transportation and Traffic Theory (ISTTT)*, Lyon, France.
19. Cascetta, E. and Postorino M.N. 1998, 'Fixed point models for the estimation of OD matrices using traffic counts on congested networks', submitted to *Transportation Science*.

20. Chin, D.C. 1997, 'Comparative study of stochastic algorithms for system optimization based on gradient approximations', *IEEE Transaction on System, Man and Cybernetics*, Part B, vol. 27, no. 2, pp. 244-249.

21. Corana, A., Marchesi, M., Martini, C. and Ridella, S. 1987, 'Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm', *ACM Transactions on Mathematical Software*, vol. 13, 262-280.

22. Dantzig, G. 1963, *Linear programming and extensions,* Princeton University Press.

23. De Plama, A. and Fontan C. 2001, 'Choix modal et valeur du temps en Ile-de-France', Working Paper, THEMA.

24. De Plama, A. and Rochat D. 1995, 'Etude emprique du choix de l'Heure de depart au travail', Working Paper, THEMA.

25. Dupuit, J. 1844, 'On the measurement of the utility of public works', Reprinted in Munby D. (eds.) 1968, *Transport*, Penguin, London, pp. 19-57.

26. Gabuthy, Y., Neveu, M., and Boemont, L.D. 2006, 'Structural model of peak-period congestion: an experimental study', *Review of Network Economics*, vol. 5, 273-298.

27. Ghosh, A. 2000, 'To pay or not to pay: commuters' mode choice under real time congestion pricing', Working Papers, University of California, Irvine.

28. Goffe, W.L., Ferrier, G.D. and Rogers, J. 1994, 'Global optimization of statistical functions with simulated annealing', *Journal of Econometrics*, vol. 60, pp. 65-99.

29. Goldberg, D.E. 1989, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, New York.

30. Goodwin, P. 1992, 'A review of new demand elasticities with reference to short and long run effects to price changes', *Journal of Transport Economics and Policy*, vol. 26, no. 2, pp. 155-169.

31. Hager, W.H. 2004, 'Jules Dupuit – eminent hydraulic engineer', Journal of Hydraulic Engineering, vol. 130, no. 9, pp. 843-848.

32. He. Y. 1997, 'A flow-based approach to the dynamic traffic assignment problem: formulations, algorithms and computer implementations', Master thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.

33. Henderson, J. and Fu, L. 2004, 'Applications of genetic algorithms in transportation engineering', in *the 83rd Annual Meeting of the Transportation Research Board*, Washington DC.

34. Hensher, D.A. 1996, 'Extending valuation to controlled value functions and non-uniform scaling with generalized unobserved variances', in *Proceedings of the Theoretical Foundations of Travel Choice Modeling*, Stockholm.

35. Hestenes, M.R. and Stiefel, E. 1952, 'Methods of conjugate gradients for solving linear systems', *J. Research Nat. Bur. Standards*, vol. 49, pp. 409–436.

36. Holland, J.H. 1975, *Adaptation in natural and artificial systems*, University of Michigan Press, Michigan.

37. Hooke, R. and Jeeves, T.A. 1961, 'Direct search solution of numerical and statistical problems', *Journal of the ACM*, vol. 8, pp. 212-229.

38. Hultkrantz, L. and Mortazavi, R. 2001, 'Anomalies in the value of travel-time changes', *Journal of Transport Economics and Policy*, vol. 35, no. 2, pp. 285-300.

39. Huyer, W. and Neumaier, A. 2004, 'SNOBFIT – stable noisy optimization by branch and fit', Submitted to ACM Transactions on Mathematical Software.

40. Ieromonachou, P., Potter, S. and Warren, J.P. 2006, 'Norway's urban toll rings: evolving towards congestion charging', *Transport Policy*, vol. 13, no. 5, pp. 367-378.

41. Joksimovic, D., Bliemer, M. and Bovy P.H.L. 2005, 'Optimal toll design problem in dynamic traffic networks – with joint route and departure time choice', in *the 84th Annual Meeting of the Transportation Research Board*, Washington DC.

42. Jou, R.C., Kitamura, R., Weng, M.C. and Chen, C.C. 2008, 'Dynamic commuter departure time choice under uncertainty', forthcoming in *Transportation Research*, Part A.

43. Karmarkar, N. 1984, 'A new polynomial-time algorithm for linear programming', *Mathematics and Statistics*, vol. 4, no. 4, pp. 373-395.

44. Kiefer, J. and Wolfowitz, J. 1952, 'Stochastic estimation of the maximum of a regression function', *Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462-466.

45. Kim, K.O. and Rilett, L. R. 2004, 'A genetic algorithm based approach to traffic

micro-simulation calibration using its data', in *the 83rd Annual Meeting of the Transportation Research Board*, Washington DC.

46. Kleijnen, J.P.C. 1987, *Statistical tools for simulation practitioners*, Marcel Dekker, New York.
47. Knight, F. 1924, 'Some fallacies in the interpretation of social cost', *Quarterly Journal of Economics*, vol. 38, pp. 582-606.
48. Koppelman, F.S., Wen, C.H. 2000, 'The paired combinatorial logit model: properties, estimation and application', *Transportation Research*, Part B, vol. 34, no. 2, pp. 75-89.
49. Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E. 1998, 'Convergence properties of the nelder-mead simplex method in low dimensions', *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112-147.
50. Lawphongpanich, S. and Hearn, D. 2004, 'An MPEC approach to second-best toll pricing', *Mathematical Programming*, vol. 101, no. 1, pp. 33-55.
51. Leurent, F. 1998, 'The distribution of car commuters' values of time in Marseilles', *Recherche Transports Securite*, vol. 60, pp. 19-38.
52. Lotan, T. and Koutsopoulos, H. 1993, 'Models for route choice behavior in the presence of information using concepts from fuzzy control and approximate reasoning', *Transportation*, vol. 20, no. 2, pp. 129-155.
53. Luo, Z.Q., Pang, J.S. and Ralf, D. 1996, *Mathematical programs with equilibrium constraints*, Cambridge University Press, New York.
54. Mackett, R. 2000, 'How to reduce the number of short trips by car', *European Transport Conference*, vol. 438, pp. 87-98.
55. McFadden, D. 1978, 'Modeling the choice of residential location', in *Transportation Research Record*, vol. 672, Washington DC.
56. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. 1953, 'Equation of state calculations by fast computing machines', *Journal of Chemical Physics*, vol. 21, pp. 1087-1092.
57. Nelder, J.A. and Mead, R. 1965, 'A simplex method for function minimization', *Computer Journal*, vol. 7, no. 4, pp. 308-313.
58. Oum, T., Waters, W. and Yong J.S. 1992, 'Concepts of price elasticities of transport

demand and recent empirical estimates', *Journal of Transport Economics and Policy*, vol. 26, no. 2, pp. 139-154.

59. Ozbay, K. and Tuzel, O.Y. 2008, 'Valuation of travel time and departure time choice in the presence of time-of-day pricing', *Transportation Research,* Part A, vol. 42, no. 4, pp. 577-590.
60. Partiksson, M. 1999, *Nonlinear programming and variational inequality problems: a unified approach,* Kluwer Academic Publisher, Dordrecht, Netherlands.
61. Pigou, A.C. 1912, *Wealth and welfare*, MacMillan, London.
62. Polyak B.T. and Juditsky A.B. 1992, 'Acceleration of stochastic approximation by averaging', *SIAM Journal of Control and Optimization*, vol. 30, no. 4, pp. 838-855.
63. Ramming, M.C. 2002, 'Network knowledge and route choice', PhD thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.
64. Rathi, V. 2007, 'Assessment of impact of dynamic route guidance through variable message signs', Master Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts.
65. Road Charging Scheme: South America - Chile, Santiago de Chile, 2004, from <http://www.cfit.gov.uk/map/southamerica-chile-santiago.htm>.
66. Santos, G. 2003, 'Urban congestion charging: optimal and second best alternatives', *Mimeo*, Department of Applied Economics, Cambridge.
67. Schiller, E. 1998, 'The road ahead: the economic and environmental benefits of congestion pricing', Cache Copy.
68. Schrank, D. and Lomax, T. 2007, 'The urban mobility report', Texas Transportation Institute, College Station, TX.
69. Sheffi, Y and Powell W.B. 1982, "An algorithm for the equilibrium assignment problem with random link times", *Networks*, vol. 12, pp. 191-207.
70. Skinner, R.E. 2000, 'Transportation in the 21st century', *Public Roads*, vol. 64, no. 2, pp. 42-49.
71. Small, K. and Sullivan, E.C. 2001, 'Choice models of route, occupancy, and time-of-day with value priced tolls', in *the 80th Transportation Research Board Annual Meeting*,

Washington DC.

72. Spall, J.C. 1992, 'Multivariate stochastic approximation using a simultaneous perturbation gradient approximation', *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332-341.

73. Spall, J.C. 1994a, 'Developments in stochastic optimization algorithms with gradient approximations based on function measurements', in Tew, J.D., Manivannan, S., Sadowski, D.A. and Seila, A.F. (eds.), Proceedings of *the 1994 Winter Simulation Conference*, pp. 207-214.

74. Spall, J.C. 1994b, 'A second order stochastic approximation algorithm using only function measurements', Proceedings of *the 33rd Conference on Decision and Control*, Lake Buena Vista, Florida, pp. 2472-2477.

75. Spall, J.C. 1998a, 'Implementation of the simultaneous perturbation algorithm for stochastic optimization', *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 817-823.

76. Spall, J.C. 1998b, 'An overview of the simultaneous perturbation method for efficient optimization', *Johns Hopkins APL Technical Digest*, vol. 19, no. 4, pp. 482-492.

77. Spall, J.C. 1999, 'Stochastic optimization, stochastic approximation and simulated annealing', in Webster, J.G. (eds.), *Wiley Encyclopedia of Electrical and Electronics Engineering*, Wiley-Interscience, pp. 529-542.

78. Sullivan, E. 2000, 'Continuation study to evaluate the impacts of the SR 91 value-priced express lanes', Final Report, State of California, Department of Transportation.

79. TCRP 2003, 'Road value pricing: traveler response to transport system changes', Transit Cooperative Research Program Report 95, Transportation Research Board, from <http://onlinepubs.trb.org/onlinepubs/tcrp/tcrp_rpt_95c14.pdf>.

80. Teodorovic, D. and Kikuchi, S. 1990, 'Transportation route choice model using fuzzy inference technique' in Ayyub, B.M. (eds.), Proceedings of ISUMA 90, *The First International Symposium on Uncertainty Modeling and Analysis*, IEEE Computer Press, College Park, Maryland, pp. 140-145.

81. Teodorovic D. 1999, 'Fuzzy logic systems for transportation engineering: the state of the art', *Transportation Research*, Part A, vol. 33, no. 5, pp. 337-364.

82. Vaze, V. 2007, 'Calibration of dynamic traffic assignment models with point-to-point traffic surveillance', Civil and Environmental Engineering Department, Master Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

83. Verhoef, E. 2002, 'Second-best congestion pricing in general networks: heuristic algorithms for finding second-best optimal toll levels and toll points', *Transportation Research* Part B, vol. 36, no. 8, pp. 707–729.

84. Verhoef, E. 2002, 'Second-best congestion pricing in general static transportation networks with elastic demands', *Regional Science and Urban Economics*, vol. 32, pp. 281-310.

85. Verhoef E., Nijkamp, P. and Rietveld, P. 1996, 'Second-best congestion pricing: the case of an untolled alternative', *Journal of Urban Economics*, vol. 40, no. 3, pp. 279–302.

86. Verhoef, E., Nijkamp, P. and Rietveld, P. 1995, 'The economics of regulatory parking policies: the (im)possibilities of parking policies in parking regulation', *Transportation Research* Part A, vol. 29, no. 2, pp. 141-156.

87. Victoria Transport Policy Institute 2000, 'Transportation elasticities, in traffic demand management encyclopedia', from <http://www.vtpi.org/tdm/tdm11.htm>.

88. Wardrop, J. 1952, 'Some theoretical aspects of road traffic research', *Proceedings of the Institute of Civil Engineers,* vol. 1, no. 2, pp. 325–378.

89. Wie, B. and Tobin, R.L. 1998, 'Dynamic congestion pricing models for general traffic networks', *Transportation Research* Part B, vol. 32, no. 5, pp. 313-327.

90. Yan, H. and Lam, W.H.K. 1996, 'Optimal road tolls under conditions of queuing and congestion', *Transportation Research* Part A, vol. 30, no. 5, pp. 319-332.

91. Yang H. 1995, 'Heuristic algorithms for the bi-level origin-destination matrix estimation problem', *Transportation Research* Part B, vol. 29, no. 4, pp. 231-241.

92. Yang H. and Yagar S. 1994, 'Traffic assignment and traffic control in general freeway-arterial corridor systems', *Transportation Research* Part B, vol. 28, no. 6, pp. 463-486.

93. Yang, Q. and Koutsopoulos, H.N. 1996, 'A microscopic traffic simulator for evaluation of dynamic traffic assignment systems', *Transportation Research* part C, vol. 4, no. 3, pp. 113-129.

94. Yang, Q., Koutsopoulos, H.N. and Ben-Akiva, M. 2000, 'A simulation model for evaluating dynamic traffic management systems', *Transportation Research Record*, no. 1710, pp. 122-130.

95. Yildirim, M.B. 2001, 'Congestion toll pricing models and methods for variable demand networks', Dissertation, Department of Industrial and Systems Engineering, University of Florida, Gainesville, Florida.

96. Ypma, T.J. 1995, 'Historical development of the Newton-Raphson method', *SIAM Review*, vol. 37, no. 4, pp. 531–551.

97. Zadeh, L.A. 1965, 'Fuzzy sets', *Information and Control,* vol. 8, no. 3, pp. 338-353.