# ABSTRACT

Title of dissertation: MULTI-CRITERION DYNAMIC TRAFFIC ASSIGNMENT MODELS AND ALGORITHMS FOR ROAD PRICING APPLICATIONS WITH HETEROGENEOUS USERS

Chung-Cheng Jason Lu, Ph.D., 2007

Dissertation Directed by: Professor Hani S. Mahmassani, Department of Civil and Environmental Engineering

This study develops a simulation-based dynamic traffic assignment, or dynamic user equilibrium (DUE), model for dynamic road pricing applications. This proposed model is considered as the bi-criterion DUE (BDUE) model, because it explicitly considers heterogeneous users with different values of time (VOT) choose paths that minimize the two path attributes: travel time and out-of-pocket cost. This study assumed trip-makers would select their respective least generalized cost paths, the generalized cost being the sum of travel cost and travel time weighted by the trip-maker's VOT. The VOT is modeled as a continuous random variable distributed across all users in a network.

The BDUE problem is formulated as an infinite dimensional variational inequality (VI), and solved by a column generation-based algorithmic framework which embeds (i) a parametric analysis (PAM) to obtain the VOT breakpoints which determine multiple user classes, and find the set of extreme non-dominated paths, (ii) a simulator to determine experienced travel times, and (iii) a multi-class path flow equilibrating scheme to update path assignments. The idea of finding and assigning heterogeneous trips to the

set of extreme non-dominated paths is based on the assumption that in the disutility minimization path choice model with convex utility functions, all trips would choose only among the set of extreme non-dominated paths. Moreover, to circumvent the difficulty of storing the grand path set and assignment results for large-scale network applications, a vehicle-based implementation technique is proposed. This BDUE model is generalized to the multi-criterion DUE (MDUE) model, in which heterogeneous users with different VOT and values of reliability (VOR) make path choices so as to minimize their path travel cost, travel time, and travel time variability.

Another important extension of the BDUE model is the multi-criterion simultaneous route and departure time user equilibrium (MSRDUE) model, which considers heterogeneous trip-makers with different VOT and values of schedule delay (VOSD) making simultaneous route and departure time choices so as to minimize their respective trip costs, defined as the sum of travel cost, travel time weighted by VOT, and schedule delay weighted by VOSD. The MSRDUE problem is also solved by the column generation-based algorithmic framework. The Sequential Parametric Analysis Method (SPAM) is developed to find the VOT and VOSD breakpoints that define multiple user classes, and determine the least trip cost alternative (a combination of departure time and path) for each user class.

# MULTI-CRITERION DYNAMIC TRAFFIC ASSIGNMENT MODELS AND ALGORITHMS FOR ROAD PRICING APPLICATIONS WITH HETEROGENEOUS USERS

By

Chung-Cheng Jason Lu

Dissertation submitted to the Faculty of the Graduate School of the University of Maryland, College Park, in partial fulfillment of the requirements for the degree of Doctor of Philosophy
2007

Advisory Committee:
Professor Hani S. Mahmassani, Chair
Professor Bruce L. Golden
Professor Paul M. Schonfeld
Associate Professor Elise D. Miller-Hooks
Assistant Professor Cinzia Cirillo

Acknowledgements

First of all, I would like to express my sincerest gratitude to my advisor, Dr. Hani S. Mahmassani, for his support, encouragement, and guidance throughout my Ph.D. study. His exemplary working ethic and commitment to excellence and quality are inspiring. It is a great honor to work with him. I am very thankful to have this opportunity. Indeed, this challenging experience forms the strongest basis for my professional life.

I am also grateful to my advisory committee, Drs. Bruce Golden, Paul Schonfeld, Elise Miller-Hooks, and Cinzia Cirillo, for their valuable suggestions and comments. Special thanks to Dr. Schonfeld for his timely help and insightful advice.

I have been fortunate to work with a great team during my study in the University of Maryland. Special thanks to Dr. Xuesong Zhou for his help and guidance on every aspect of my research. Working with him has been one of the most delightful and productive experiences in my life. I am also thankful to my colleagues, Dr. Xiao Qin, Hayssam Sbayti, Roger Chen, Sevgi Erdogan, Xiang Fei, Jing Dong, Stacy Eisenman, Kuilin Zhang, Samer Hamdar, and Vishnu Arcot, for their pleasant collaboration and genuine friendship. In addition, I am thankful to all the friends I have met here. Their presences make life and experience in this country so joyful and interesting.

My heartfelt thanks go to my parents and parents-in-law for their love and endless support. Finally, I am deeply indebted to my beloved wife, Chiu-Chun Joanna Chen, for her love, thoughtful understanding, perpetual encouragement, and meticulously taking care of every aspect of my daily life. I extremely appreciate her unconditional support and contribution throughout my Ph.D. study.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1 Introduction

## 1.1 Motivation and Objective

The user equilibrium (UE) traffic assignment problem has been studied extensively in the past five decades since the pioneering work of Beckmann et al. (1956), introducing a mathematical program whose Kuhn-Tucker conditions coincide with Wardrop's UE principle (1952). An important extension of the problem is the dynamic user equilibrium (DUE) traffic assignment (or user equilibrium dynamic traffic assignment, UEDTA) problem, which addresses the dynamic nature of traffic demands and flows in road networks as well as the path choice and/or departure time decisions of network users (Boyce et al., 2005). DUE models have evolved substantially in the last decade, and are seeing wider use in practice for predicting dynamic traffic flow patterns in evaluating traffic control and travel demand management measures.

Among various travel demand management strategies, road (congestion) pricing has long been considered by economists (e.g. Walters, 1961; Vickrey, 1963 and 1969; Roth, 1967), transportation researchers (e.g. Yang and Bell, 1997; Verhoef, 2002; Yildirim and Hearn, 2005) and authorities as an effective way of reducing traffic congestion and improving system performance during peak periods in many metropolitan areas, notwithstanding a general attitude of public opposition. Because of the time-varying nature of traffic and congestion, dynamic road pricing has recently drawn increasing attention from the congestion pricing research community (Arnott et al, 1990; Wie and Tobin, 1998; Joksimovic et al., 2005). To support the planning, operation, and evaluation of various dynamic road pricing schemes on road traffic networks, DUE models are often applied to predict path choices and the resulting network flow patterns, which in turn form the basis for assessing the economic and financial impacts or benefits of proposed toll facilities or schemes. To this end, DUE models for dynamic road pricing applications should essentially be able to

(i) address the heterogeneous user preference of path (and/or departure time) choice in response to time-varying toll charges,

(ii) capture traffic flow dynamics and spatial and temporal vehicular interactions,

(iii) adhere to the time-dependent generalization of Wardrop's UE principle (or so-called DUE conditions), and

(iv) be deployable on road traffic networks of practical sizes.

This study aims to develop simulation-based DUE models and solution algorithms that fulfill these fundamental requirements. The following subsections summarize the three major tasks of achieving the objective of the dissertation.

### 1.1.1 Address user heterogeneity in DUE models

Recent advances in the development of DUE models (Mahmassani, 2001; Peeta and Ziliaskopoulos, 2001; Lu et al., 2006) have facilitated the design and evaluation of various road pricing scenarios that vary with location, time, and prevailing network states. One of the critical tasks in developing DUE models for dynamic road pricing applications is to realistically model trip-makers' path (and/or departure time) choice decisions in response to time-varying toll charges. The most widely studied (path) choice model in the literature is the probabilistic discrete choice model which assumes that, in a (random) disutility minimization decision framework, each trip-maker would choose a path that minimizes his/her own perceived disutility. Generally, this disutility is considered as the sum of several path attributes (e.g. travel time and out-of-pocket cost) and users' social-economic characteristics (e.g. age and income) weighted by their respective coefficients plus a random term representing unobserved influences on path choice behavior. While some stochastic traffic assignment models (e.g. Sheffi, 1985; Maher, 1997 and 1998; Cantarella and Binetti, 1998; Abdelghany, 2001; Nielsen, et al., 2002; Florian, 2006) have explicitly incorporated this type of random disutility function in the underlying discrete path choice model to determine path choice probabilities, deterministic traffic assignment models commonly adopt the other type of path choice model based on the generalized path cost function in which path travel time is weighted by a trip-maker's value of time (VOT) representing how much money the trip-maker is willing to tradeoff for unit time saving (e.g. Dial, 1979; Cantarella and Binetti, 1998).

Conventional (static) traffic assignment models (e.g. Yang and Meng, 2000) for road pricing applications consider a homogeneous perception of tolls for all trip-makers by assuming that every trip-maker is willing to tradeoff the same amount of money for a unit time saving, corresponding to a constant VOT (or constant time and cost coefficients) in the underlying path choice model. However, empirical studies (e.g. Ben-Akiva et al., 1993; Hensher, 2001a and 2001b) have found that discrete path choice models with random coefficients have better goodness of fit than those with constant coefficients and others (e.g. Small and Yan 2001; Brownstone and Small, 2005; Small et. al. 2005; Cirillo and Axhausen, 2006) suggested that the VOT varies significantly across individuals because of different socio-economic characteristics, trip purposes, and inherent preferences. This user heterogeneity is manifested in the fact that some trips take slower paths to avoid tolls while others choose toll roads to save time. Therefore, it is essential to explicitly recognize and represent heterogeneous users in modeling users' response to toll charges in DUE models for road pricing applications. This is especially important in assessing the feasibility of a proposed toll facility and its financial viability from the standpoint of the public or private entity that will be operating it.

The need to address the user heterogeneity in evaluating road pricing applications has indeed drawn increasing attention from the traffic assignment research community. Recent studies on the static traffic assignment model that take into account the user heterogeneity in the underlying path choice model have either embedded random coefficient discrete path choice models (e.g. Nielsen, et al., 2002) or have relaxed the conventional assumption of constant VOT to a discrete (e.g. Yang et al. 2002; Nagurney and Dong, 2002) or continuous (e.g. Leurent, 1993; Dial, 1996 and 1997; Marcotte and Zhu, 1997) random variable distributed across the population of network users. Nevertheless, none of existing DUE models for dynamic road pricing applications has explicitly considered heterogeneous preference of user path choices, at least to the author's current knowledge. In fact, the attempt to accurately design and evaluate dynamic pricing schemes relies on a realistic representation of complex traffic dynamics and spatial and temporal vehicular interactions in network equilibrium assignment models, hence necessitating the extension of the heterogeneous traffic assignment model from the static regime to the DTA context. To this end, this study aims at developing a bi-criterion dynamic user equilibrium (BDUE) traffic assignment model wherein the user heterogeneity, in terms of different VOT preferences, is addressed and the two essential path choice criteria: travel time and out-of-pocket cost are simultaneously taken into account in the underlying path choice framework.

### 1.1.2 Improve the theoretical basis for simulation-based DUE approach

Existing DUE models and algorithms can be generally classified into either analytical or simulation-based (Peeta and Ziliaskopoulos, 2001). Analytical DUE models (e.g. Janson, 1991a and 1991b; Friesz et al., 1989; Ran et al., 1993) typically employ link/node exit constraints to propagate traffic flows and link performance functions to determine path travel costs. Using well-defined exit constraints and cost functions makes it possible to establish theoretically the properties of solutions (e.g. existence and uniqueness) and the adherence to the DUE conditions. However, theoretical elegance is obtained at the cost of behavioral realism in terms of representing the dynamics of traffic flow. On the other hand, the simulation-based approach describes traffic flow propagation, captures spatial and temporal vehicular interactions, and determines link and path travel costs through traffic simulation instead of analytical evaluation (e.g. Smith, 1993; Peeta and Mahmassani, 1995; Ben-Akiva et al., 1997). This provides considerable modeling flexibility (e.g. of traffic control measures and information supply strategies) for a wide range of engineering applications. However, using traffic simulation to reflect the properties of the actual underlying real systems, which are generally not well-behaved mathematically, often precludes guaranteed algorithmic convergence and solution optimality (i.e. adherence to the DUE conditions). Therefore, while analytical models have served primarily to derive theoretical insights, simulation-based models have successfully tackled many practical aspects that enable deployment in real networks.

This study intends to develop a theoretically sound simulation-based DUE model and its solution algorithm, with a particular emphasis on obtaining solutions (i.e. time-varying path flows) that adhere to the DUE conditions. During the past decade, most analytical DUE studies (e.g. Wie et al. 1995; Huang and Lam, 2002; Jang et al. 2005) had mainly attempted to develop more sophisticated flow propagation constraints and elegant link performance functions, which may lead to computational intractable mathematical models. On the other hand, the development of simulation-based models (e.g. Ben-Akiva et al., 1997; Tong and Wong, 2000) had mostly focused on including more traffic behavioral realism (i.e. toward microscopic simulation) and applying computationally efficient heuristic approaches, such as the method of successive averages (MSA). Although a few recent studies (Ziliaskopoulos and Waller, 2000; Lo and Szeto, 2002) have embedded macroscopic traffic flow models, such as the Cell Transmission Model (CTM; Daganzo, 1994 and 1995a), in their analytical DUE frameworks, to circumvent the need to use flow propagation constraints and link performance functions, none has addressed the theoretical weakness for the simulation-based DUE approach, at least to the author's current knowledge. Thus, to attain the objective of developing a simulation-based DUE model for dynamic road pricing applications, an important task of this dissertation is to improve the theoretical basis for the simulation-based DUE approach.

### 1.1.3 Address practical deployment issues for large-scale road networks

While the theoretical background and algorithmic framework have been the primary focus in the relevant literature of (static) heterogeneous traffic assignment problems as well as DUE problems, limited attention has been accorded to practical deployment issues of their models and solution algorithms, such as computational efficiency and solution-storing space requirements, especially for large-scale network planning and real-time operational applications.

Traffic assignment model formulations can be basically classified into two categories: link flow-based and path flow-based; the former seeks a unique UE solution as a link flow vector while the latter finds a (non-unique) path flow vector satisfying the UE conditions. When the problem is extended from the static regime to the DTA context, researchers have shown greater interest in the path flow-based formulation that seeks a time-varying path flow vector satisfying the DUE conditions than in the link flow-based formulation, due to the recent advancement and deployment of intelligent transportation systems (ITS), in particular the route guidance information systems. However, solving the path flow-based formulation would require enumerating and storing the paths, for each OD pair and departure time interval, on which trip-makers would be assigned. Although some efficient time-dependent shortest path algorithms have been proposed in the literature (e.g. Ziliaskopoulos and Mahmassani, 1993), finding the path set in large-scale network problem instances is still computationally intensive. Moreover, the memory requirement for storing the grand path set and path assignment results would lead to a technique bottleneck for deploying path flow-based DTA models (Peeta, 1994).

It has been recognized that, for large-scale DTA problems, classical optimization algorithms for solving static UE problems could not be readily applied, because the temporal dimension renders the task of calculating partial derivatives (i.e. gradient) associated with descent search directions and performing line searches to determine optimal step sizes computationally intensive or intractable. Furthermore, when experienced path costs are obtained through a simulation-based dynamic traffic model (i.e. traffic simulator), analytical calculations of partial derivatives are not available.

Though it is possible to compute them using numerical methods, the stability and accuracy of numerically calculated derivatives are not guaranteed.

In summary, to meet the requirements (or challenges) in developing and deploying DUE models for evaluating dynamic road pricing schemes, the solution algorithm should be able to efficiently find and store the set of time-dependent paths (and the corresponding path assignment results), as well as to avoid relying on the gradient information in the search process (while maintain satisfactory solution quality). This study focuses on not only the theoretical and algorithmic aspects of the DUE problem but also the above important practical deployment issues for large-scale DUE models.

## 1.2 Problem Definition

### 1.2.1 BDUE problem

Given a time-dependent network G = ($N$, $A$), where $N$ is the set of nodes and A is the set of directed links ($i$, $j$), $i \in N$ and $j \in N$. The time period of interest (planning horizon) is discretized into a set of small time intervals, $S = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$, where $t_0$ is the earliest possible departure time from any origin node, $\sigma$ is a small time interval during which no perceptible changes in traffic conditions and/or travel cost occur, and $M$ is a large number such that the intervals from $t_0$ to $t_0 + M\sigma$ cover the planning horizon $S$. Denote $c_{ij}(t)$ and $d_{ij}(t)$ the travel cost (e.g. toll) and travel time, respectively, required for traveling on link ($i$, $j$) in time interval $t$. The experienced *path generalized cost* is defined as the sum of path travel cost and path travel time weighted by the trip-maker's VOT. The VOT relative to each trip represents how much money the trip-maker is willing to trade for a unit time saving. To reflect heterogeneity of the population, the VOT in this study is treated as a continuous random variable distributed across the population of trip-makers, with a known probability density function and a given feasible range. The time-dependent origin-destination (OD) demand for the entire feasible range of VOT over the planning horizon (i.e. number of trips for each OD pair, each departure time interval and each possible value of VOT) is also known a priori. In practice, the OD demand pattern and the VOT distribution will be considered independent of each other.

The bi-criterion dynamic user equilibrium (BDUE) traffic assignment problem addressed in this dissertation explicitly considers heterogeneous trip-makers with different VOT choosing paths that minimize the two essential path choice criteria: travel time and out-of-pocket cost. By following the modeling framework typically adopted in discrete time, deterministic DUE models for describing trip-makers' path choice behavior, each trip-maker is assumed to choose a path minimizing the generalized cost with respect to his/her own VOT. Based on this behavioral assumption, the bi-criterion dynamic user equilibrium (BDUE), a bi-criterion and dynamic extension of Wardrop's UE principle (1952), is defined as:

*For each OD pair and for each departure time interval, every trip-maker cannot decrease the experienced generalized trip cost with respect to that trip's particular VOT* $\alpha$ *by unilaterally changing paths.*

This implies that, at BDUE, each trip-maker is assigned to a path with the least generalized cost with respect to his/her own VOT. This definition can be also viewed as the dynamic extension of Dial's bi-criterion equilibrium traffic assignment (Dial, 1996) or Leurent's cost versus time equilibrium (Leurent, 1993). Given the assumptions and definition above, this study aims at solving the BDUE traffic assignment problem, under a given time-dependent road pricing scheme, to obtain the time-varying path flow pattern satisfying the BDUE conditions. Specifically, the focus is to determine the BDUE path flows (routing policies) in a vehicular network for each OD pair, each departure time interval and all possible values of time.

### 1.2.2 MDUE problem

This BDUE problem defined above is generalized to the multi-criterion DUE (MDUE) problem, in which heterogeneous users with different VOT and values of reliability (VOR) make path choices so as to minimize their path travel cost, travel time, and travel time variability (or reliability). The travel time variability of a path in a departure time interval is defined as the variance (or standard deviation) of experienced path travel times of vehicles entering that path in that departure time interval, and the VOR reflects the monetary value perceived by a trip-maker for a unit reduction in travel time variability. Both VOT and VOR are considered as continuous random variables distributed across the population of trip-makers in a network, with known probability density functions and given feasible ranges. Each trip-maker is assumed to choose a path minimizing the generalized cost with respect to his/her own VOT and VOR, the path generalized cost being defined as the sum of travel cost, travel time weighted by VOT, and travel time standard deviation weighted by VOR. Based on this assumption, the multi-criterion dynamic user equilibrium (MDUE), a multi-criterion and dynamic extension of Wardrop's first principle (1952), is defined as:

*For each OD pair and for each departure time interval, every trip-maker cannot decrease the experienced generalizedt with respect to that trip's particular VOT and VOR by unilaterally changing paths.*

This implies that, at MDUE, each trip-maker is assigned to a path with the time-dependent least generalized cost with respect to his/her own VOT and VOR. Given the assumptions and the definition above, this study aims at solving the MDUE problem, under a given dynamic road pricing scheme, to obtain a time-varying path flow pattern satisfying the MDUE conditions. Specifically, the focus is to determine the MDUE path flows (routing policies) in a vehicular network for each OD pair, each departure time interval and all possible values of time and values of relaibility.

### 1.2.3 MSRDUE problem

The BDUE problem defined in the previous subsection assumes the time-varying OD demands for the entire feasible range of VOT and over the planning horizon are known and fixed, a priori; or equivalently trip-makers' departure times are fixed. However, in general, a trip-maker facing a toll road with time-varying charges would not only change path (or route) but also adjust departure time so as to minimize his/her total trip cost. Therefore, a realistic generalization of the BDUE problem is to allow trip-makers to make departure time choices, in addition to path choices, in response to time-varying toll charges. This dissertation deals with this important extension of the BDUE problem – the multi-criterion simultaneous route and departure time user equilibrium (MSRDUE) problem, which explicitly considers heterogeneous trips (or trip-makers) with different values of time (VOT) and values of (early or late) schedule delay (VOESD or VOLSD) simultaneously choosing departure times and paths that minimize the set of trip attributes: travel time, out-of-pocket cost, and schedule delay cost (or arrival time cost defined in Janson and Robles, 1993), where *schedule delay* is determined by the difference between actual and preferred arrival times (PAT).

To realistically reflect heterogeneity of the population, VOT, VOESD, and VOLSD in this study are considered as continuous random variables distributed across the population of trips, with known probability density functions and feasible ranges. Additionally, this study allows each trip to have its own PAT interval by assuming the PAT pattern follows a given discrete distribution with a given probability mass function. The behavioral assumption made in this study is: each trip-maker would choose the *alternative*, a combination of departure time and path, which minimizes his/her *trip cost*, defined as the sum of travel cost, travel time weighted by VOT, and early or late schedule delay weighted by VOESD or VOLSD. Based on this assumption, the MSRDUE, a multi-criterion and dynamic extension of Wardrop's first principle (Wardrop, 1952), is defined as the following.

*For each OD pair, every trip cannot decrease the experienced trip cost with respect to that trip's particular VOT, VOESD, VOLSD, and PAT interval by unilaterally changing departure time and/or path.*

This implies that, at MSRDUE, each trip-maker is assigned to the alternative with the least trip cost with respect to his/her own PAT, VOT, VOESD, and VOLSD. This definition can be viewed as the heterogeneous (or multi-criterion) generalization of the simultaneous route and departure time user equilibrium (SRDUE) in the literature (Freisz et al. 1993; Zilliaskopoulos and Rao, 1999). Given the assumptions and definition above, this study aims at solving the MSRDUE problem, under a given set of time-varying link tolls and given heterogeneous OD demands, to obtain temporal splits (among departure times) and spatial distributions (over paths) satisfying the MSRDUE conditions.

## 1.3 Research Overview

This section presents the overview of the dissertation, which consists of the studies of four related research topics summarized in the following.

### 1.3.1 Modeling and solving the DUE problem

This study begins with developing a reformulation and its solution algorithm for the (single-criterion) DUE problem, in which the VOT is assumed as a constant (i.e. all trip-makers have the same VOT). The particular emphasis is on improving the theoretical basis of the simulation-based DUE approach by introducing a gap function as the measure (or objective function) of deviations from the DUE conditions and developing a solution algorithm able to minimize that gap measure. The DUE problem is formulated, via that gap function and using path-based decision variables, as a nonlinear minimization problem (NMP). To circumvent the difficulty of enumerating all feasible paths for a path-based formulation, the NMP is solved by a column generation-based optimization procedure which embeds a simulation-based dynamic network loading model to capture traffic dynamics and determine experienced path travel costs for any given path flow pattern; and a descent direction method to solve the restricted NMP defined by a subset of feasible paths. The descent direction method circumvents the needs to compute the gradient of the objective function in finding search directions and to determine suitable step sizes, which are especially valuable for large-scale simulation-based applications.

### 1.3.2 Modeling and solving the BDUE problem

A major work in this dissertation is the development of the BDUE model, which explicitly considers, in the underlying path choice model, heterogeneous trip-makers with different VOT choosing paths that simultaneously optimize the two essential path choice criteria: travel time and out-of-pocket cost. To realistically capture trip-makers' path choice decisions in response to toll charges, the VOT is assumed to be continuously distributed among trip-makers, in contrast to the constant VOT assumed in conventional DTA/DUE studies. Although this critical issue of user heterogeneity has been considered in the literature (see section 2.2), all those network equilibrium assignment models (e.g. Leurent, 1993; Dial, 1996; Marcotte and Zhu, 1997) were developed only for static road pricing schemes, rather than dynamic (or time-dependent) ones. In fact, successful design and evaluation of dynamic pricing schemes relies on a realistic representation of complex traffic dynamics and spatial and temporal vehicular interactions in traffic assignment models, hence necessitating the extension of the heterogeneous traffic assignment model from the static regime to the DTA context.

The BDUE problem is formulated as an infinite dimensional variational inequality (VI), and solved by the column generation-based algorithmic framework which embeds (i) the extreme non-dominated path finding algorithm – PAM (parametric analysis method) to obtain the breakpoints which partition the entire range of VOT into many subintervals and determine the multiple user classes, and find the least generalized cost path for each user class, (ii) the traffic simulator – DYANSMART (Jayakrishnan, et al. 1994; Mahmassani, 2001) to capture traffic dynamics and determine experienced path travel times for any given path flow pattern; and (iii) the multi-class path flow updating/equilibrating scheme to solve the restricted multi-class dynamic user equilibrium (RMDUE) problem defined by a subset of feasible paths. Moreover, to circumvent the difficulty of storing the memory-intensive path set and routing policies for large-scale network applications, a vehicle-based implementation technique using the vehicle path set as a proxy for keeping track of the path assignment results is applied.

### 1.3.3 Modeling and solving the MDUE problem

This study extends BDUE model developed in chapter 3 to the multi-criterion context by explicitly considering the travel time variability in trip-makers' path choices and allowing not only the VOT but also the VOR to be continuously distributed among trip-makers. Specifically, the multi-criterion dynamic user equilibrium (MDUE) problem is formulated as an infinite dimensional variational inequality (VI), and solved by a column generation-based solution algorithm, which embeds (i) the sequential parametric analysis method (SPAM) to obtain the set of time-dependent extreme efficient (or non-dominated) paths and the corresponding breakpoint vectors of VOT and VOR that naturally define the multiple user classes, each of which corresponds to particular ranges of VOT and VOR, (ii) the traffic simulator – DYANSMART to capture traffic dynamics and determine experienced path travel times and their travel time standard deviations for any given path flow pattern, and (iii) the multi-class path flow updating/equilibrating scheme to solve the restricted multi-class dynamic user equilibrium (RMDUE) problem defined by a subset of time-dependent extreme efficient paths.

### 1.3.4 Modeling and solving the MSRDUE problem

The other major task of this dissertation is to develop the model and solution algorithm for the multi-criterion simultaneous route and departure time user equilibrium

(MSRDUE) problem, which considers heterogeneous trip-makes with different PAT, VOT, VOESD, and VOLSD making simultaneous route and departure time choices so as to minimize their respective *trip cost*.

The MSDUE problem is formulated as an infinite dimensional variational inequality (VI), and solved by the column generation-based algorithmic framework which embeds (i) the (extreme non-dominated) alternative finding algorithm – SPAM (sequential parametric analysis method) to obtain the VOT, VOESD, and VOLSD breakpoints that define multiple user classes, and determine the least trip cost alternative for each user class, (ii) the traffic simulator - DYANSMART (Jayakrishnan, et al. 1994) to capture traffic dynamics and determine experienced path travel times; and (iii) the multi-class path flow equilibrating scheme to solve the restricted multi-class SRDUE (RMC-SRDUE) problem defined by a subset of feasible alternatives.

## 1.4 Organization of the Dissertation

This dissertation is structured as follows. Chapter 2 gives the literature review of models and algorithms of DUE, (static) bi-criterion user equilibrium (BUE), and SRDUE problems, as well as the algorithms for finding the bi-criterion shortest paths (BSP). The gap function-based reformulation and column generation-based solution algorithm for solving the DUE problem (with constant VOT) are presented in Chapter 3, followed by the infinite dimensional VI formulation and solution algorithm of the BDUE problem described in Chapter 4. This chapter also includes the extreme non-dominated path finding algorithm – parametric analysis method (PAM). The model and algorithm for the MDUE problem is described in Chapter 5. Chapter 6 presents the MSRDUE model and the solution algorithm, in which the sequential parametric analysis method (SPAM) is developed to obtain VOT, VOESD, and VOLSD breakpoints that define multiple user classes and to find least trip cost alternative for each of them. Numerical results of conducted experiments are reported separately in each of the three chapters: 3, 4, and 5, instead of compiling all numerical experiments in one independent chapter. Concluding remarks and future research extensions are given in chapter 6.

# Chapter 2 Background Review

This chapter presents the review of the literature relevant to the problems of interest in the dissertation. Section 2.1 gives the overview of previous studies on solving the dynamic user equilibrium (DUE) traffic assignment problem. Both the analytical approach and the simulation-based approach are included in this section. Section 2.2 reviews the bi-criterion (static) user equilibrium (BUE) traffic assignment models, which generalize the conventional user equilibrium (UE) traffic assignment models by relaxing the value of time (VOT) from a constant to a discrete or continuous random variable, and their solution algorithms in the literature. The review of solution algorithms for solving the bi-criterion (or bi-objective) shortest path (BSP) problem is given in Section 2.3, where the exact and the approximate solution approaches, as well as the extension to time-dependent networks, are surveyed. Section 2.4 presents past studies on the simultaneous route (or path) and departure time user equilibrium (SRDUE) traffic assignment problem, in which trip-makers in a network are considered to not only change paths but also adjust departure times so as to minimize individual generalized travel disutility, which is the weighted combination of travel time, monetary cost, and (late or early) schedule delay penalty.

## 2.1 The DUE Traffic Assignment Problem

At the core of the Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) lies Dynamic Traffic Assignment (DTA) with its potential to provide the operators and users of traffic networks with descriptive information of the current and future traffic conditions, as well as normative information in the form of route guidance for travelers. Dynamic traffic assignment (DTA) problems have substantially evolved and been extensively studied in the past three decades, since the pioneering work of Merchant and Nemhauser (1978a and 1978b), due to the rapid advancements of ATIS and ATMS. One common feature of these DTA models is that they differ from the static traffic assignment assumptions to deal with time-varying nature of traffic flow. As in the static case (Sheffi, 1985; Patriksson, 1994), based on different assumptions made for the individual path choice decisions, there are two major classes of DTA problems: system optimal dynamic traffic assignment (SODTA), in which the total system travel cost is minimized, and user equilibrium dynamic traffic assignment (UEDTA), in which any individual chooses a path that minimizes his experienced (predictive UEDTA) or instantaneous (reactive UEDTA) travel cost. This section reviews mainly the models and solution algorithms for the (predictive) UEDTA problem with given time-varying origin-destination (OD) demands, to find a time-varying path flow pattern that satisfies the time-dependent generalization of Wardrop's first principle: travelers with the same OD and departure time *experience* the same and minimum travel cost along any used path, with no unused path offering a lower travel cost. This can be mathematically stated as follows.

The time-varying path flow vector $r^* \equiv \{ r_{odp}^{\tau} *, \forall o, d, \tau, \text{ and } p \in P(o,d,\tau) \} \in \Omega$ is a solution to the DUE problem if the following DUE conditions are satisfied:

$$r^* \times [c(r^*) - \pi(r^*)] \qquad (2.1.1)$$

$$c(r^*) - \pi(r^*) \geq 0 \qquad (2.1.2)$$

where $c(r^*)$ and $\pi(r^*)$ are the path cost vector and the least travel cost vector, respectively, with respect to the time-varying path flow vector $r^*$, and $\Omega \equiv \{r\}$is a set of feasible path flow vectors satisfying the path flow conservation and non-negativity constraints:

Following the terminology given by Smith (1993), and increasingly adopted in the literature, the problem is referred to as the dynamic user equilibrium (DUE) problem in this study; though this term had previously been used for the simultaneous route-departure time equilibrium problem, and the term time-dependent UE (TDUE) would more accurately describe the problem of interest. Existing DUE models and algorithms can be generally classified into either analytical or simulation-based (Peeta and Ziliaskopoulos, 2001), which will be discussed in the following two sub-sections.

### 2.1.1 The analytical approach

Analytical DUE model generally includes three types of methods: mathematical programming, optimal control, and variational inequality. Janson (1991a and 1991b) was among the first to model the DUE problem with the mathematical programming method; specifically, a nonlinear mixed integer programming model using path flows as decision variables was proposed. One important feature of his approach is that it seeks an equilibrium described in terms of experienced path travel times, instead of the instantaneous travel times. However, the properties of his procedure are not well-established, and it relies on static link performance functions for traffic flow modeling, which may not be able to realistically capture traffic dynamics. Friesz et al. (1989) presented link-based continuous time optimal control formulation for the UEDTA problem with a single destination. The UEDTA problem considered sought for the network equilibration in terms of instantaneous user path costs. The model applied link exit (time) functions to propagate traffic and link performance functions to determine travel costs. Ran et al. (1993) also used the optimal control approach for the reactive (or instantaneous) UEDTA problem. Recognizing the inability of the usual link cost functions to represent dynamic queueing and congestion delays, they proposed to split the link travel cost into (free-flow) moving and queueing components. Nevertheless, the functions were still assumed to be increasing and differentiable, and thus may not reflect traffic realism.

Among various analytical DUE models, the variational inequality (VI) approach, capable of handling general asymmetric cost functions and illustrating, with relative ease, the notion of experienced travel costs for the DUE problem, has increasingly been accepted for both theoretical analysis and computation of the DUE. The VI problem is a general problem formulation that encompasses a family of mathematical problem, including nonlinear equations, optimization problems, complementarity problems, and fixed point problems. VI was originally developed as a tool for studying certain classes of partial differential equations and defined over infinite-dimensional spaces. Nagurney (1998) and Patriksson (1999) provided comprehensive reviews of models, properties, solution algorithms, and applications of VI.

Since the early application of the VI approach to the fixed demand static UE problem (Smith, 1979; Dafermos, 1980), many researchers have generalized or applied the VI approach to the DUE context. Extending his work (Smith 1979) on the static UE traffic assignment problem, Smith (1993) proposed that solving the DUE traffic assignment problem is equivalent to solving the following discrete-time and path-based (finite-dimensional) VI problem: find a time-varying path flow vector $r^* \in \Omega$ such that

$$c(r^*)\bullet(r - r^*) \geq 0, \ \forall\ r \in \Omega \qquad (2.1.3)$$

where the symbol • denotes the inner product between vectors of appropriate dimensions. This path-based VI formulation of the DUE problem was widely adopted in many studies, for example, Smith and Winsten (1995), Lo and Szeto (2002), Jang et al. (2005). Since the formulation is path-based, a set of feasible paths on which the OD demands are to be equilibrated is required. It is generally very difficult, if not impossible, to enumerate the complete set of feasible paths of all OD pairs for a road network of practical size. Thus, there is a need for an efficient method to identify the subset of competing paths (e.g. the column generation-based approach used by Larsson and Patrikson, 1992). To circumvent the difficulty of enumerating paths, Ran and Boyce (1996) proposed a link-based discretized VI model for the (predictive) DUE problem. Chen and Hsueh (1998) also presented a link-based VI formulation for the DUE problem, and a solution algorithm based on the nested diagonalization procedure. However, these link-based VI formulations were still considered prohibitively expensive to be implemented on real networks (Peeta and Ziliaskopoulos, 2001).

VI theory is also a powerful tool in the qualitative analysis of equilibrium solution properties, in particular, the existence and the uniqueness. Existence of a solution to a VI problem follows from continuity of the path cost function $c(r)$ in the VI, provided that the feasible set Ω is compact convex set (e.g. Theorem 1.4 in Nagurney, 1998). Furthermore, if the path cost function $c(r)$ is strictly monotone on the feasible set Ω, then the solution is unique, if one exists (e.g. Theorem 1.5 in Nagurney, 1998), where $c(r)$ is said to be strictly monotone if $(c(r) - c(r'))^T(r - r') > 0$, $\forall r$ and $r' \in \Omega$, $r \neq r'$.

Over the past decade, researchers have developed various algorithms to solve DUE problems of the VI form. Usually, the algorithms developed for solving the finite-dimensional VI problem (e.g. Eq.2.1.3) proceed to the equilibrium solutions iteratively and progressively via some equilibration procedure that solves a linearized or relaxed substitute of the original problem in each iteration. Particularly, the equilibration sub-problem encountered at each iteration can be reformulated as an optimization problem and solved by using an appropriate nonlinear programming algorithm. The most common iterative scheme of this type includes the projection, relaxation, and linearization methods. The core of the algorithmic procedure is the calculation of $r^{k+1}$ from $r^k$, where $k$ is the iteration counter. This algorithmic step can be written in standard form as follows.

$$r^{k+1} = r^k + \rho_k \times d^k \tag{2.1.4}$$

where $d^k$ is the descent direction and $\rho_k$ is the move size along $d^k$. Note that the cosine of the angle between the gradient direction and the descent direction $d$ is always negative (see e.g. Bertsekas, 1995).

For link flow-based VI DUE models, the classical Frank-Wolfe (linearization) algorithm (Ran and Boyce, 1996) and the diagonalization algorithm (Chen and Hsueh, 1998) was extended from the static regime (see e.g. Sheffi, 1985) to the dynamic context. For path flow-based VI DUE models, the most common approach is the type of sequential decomposition algorithms (e.g. Patriksson, 1994) that decompose the original problem into many sub-problems, each of which corresponds to an origin-destination-departure time combination. For example, some studies have adopted a path-swapping method (e.g., Smith and Winsten, 1995; Cybis, 1995; Huang and Lam, 2002; Szeto and

Lo, 2005) that could be derived from the reduced gradient algorithm by Florian and Nguyen (1974), an optimization-based algorithm for solving the static traffic assignment problem. The method is based on the intuitive swaps of flows from more expensive paths to the shortest path(s), for each origin-destination-departure time combination, and the amounts swapped are proportional to the flows on the current path ($r_p$), the difference in travel cost between the current path and the shortest path $(c_p - c_{p*})$, and the step size. Specifically, the flow moved from a non-shortest path *p* to the shortest path *p** is:

$$\Delta r_p = \max\{0, \rho_k \times r_p \times (c_p - c_{p*})\} \quad (2.1.5)$$

No systematic ways of determining the swapping rate have been reported in the literature; the swapping rate should be carefully chosen to prevent undue oscillations (Smith and Winsten, 1995) as it has a significant impact on the algorithmic convergence and computational time (Szeto and Lo, 2005). Jayakrishnan et al. (1994b) adapted the gradient projection method proposed by Bertsekas and Gafni (1983). This method leads to the following path flow swap scheme.

$$\Delta r_p = \max\{0, \rho_k \times s_k^{-1} \times (c_p - c_{p*})\} \quad (2.1.6)$$

where $s_k$ is the scaling factor, functioned as the diagonal element of the Hessian matrix. A similar approach is proposed by Nagurney and Zhang (1996 and 1997) in the following dynamic path choice adjustment processes.

$$r^{k+1} = P_\Omega\{r^k - \rho_k \times c(r^k)\} \quad (2.1.7)$$

where $P_\Omega\{r\}$ denotes the unique projection of flow vector *r* onto the feasible set Ω. This path choice adjustment process leads to the following path flow swap scheme.

$$\Delta r_p = \max\{0, \rho_k \times (c_p - c_{p^*})\} \tag{2.1.8}$$

Note that (2.1.5), (2.1.6), and (2.1.8) can be considered as a family of general path-swapping methods that differ only in the swapping rate at which the flow is moved from the non-shortest paths to the shortest path. Moreover, this family of methods moves flows from *all* non-shortest paths to the shortest path(s); while the equilibration operator type approach, such as the equilibration algorithm (e.g. Dafermos, 1968; Dafermos and Sparrow, 1969) and the convex simplex algorithm (e.g. Nguyen), moves flows from the most expensive path (or tree) to the least expensive path (or tree), and the amounts shifted are determined by some line search method.

### 2.1.2 The simulation-based approach

The analytical DUE models mentioned above typically employ link/node exit functions or constraints to propagate traffic flows, and assume convex, continuous and strictly monotonic (increasing) link performance functions to determine path travel costs. Using well-defined exit constraints and cost functions makes it possible to establish theoretically the properties of solutions (e.g. existence and uniqueness) and the satisfaction of DUE conditions. However, theoretical elegance is obtained at the cost of behavioral realism in terms of representing the dynamics of traffic flow. For instance, including the kind of exit constraints necessary to ensure first-in, first-out in an analytical DUE model leads to a loss of analytical tractability (Carey, 1992). Widely-used macroscopic travel time functions (e.g., the Bureau of Public Road functions) in a dynamic formulation is not consistent with elementary traffic flow relations, and hence does not adequately capture traffic flow dynamics, such as queue build-up, spillback and dissipation in congested networks (Daganzo, 1995b).

Recognizing that analytical representations of realistic traffic dynamics in a general network, with well-behaved mathematical properties, remain to be developed, the simulation-based approach describes traffic flow propagation, captures spatial and temporal vehicular interactions, and determines link and path travel costs through traffic simulation instead of analytical evaluation. This provides considerable modeling flexibility (e.g. of traffic control measures and information supply strategies) for a wide range of engineering applications. Another advantage of simulation-based approach is to circumvent the need to specify link performance functions which are assumed in most analytical DTA approaches as convex, continuous, and increasing function of link traffic volume to simply the evaluation of (actual or experienced) path travel times. The computation of those path travel times is not trivial in the DTA context, as the paths followed by future trips may share common links with paths assigned to current trips, thereby influencing the travel times experienced by the vehicles currently assigned. Thus, the experienced path travel times are the net result of the complex nonlinear spatial and temporal interactions among many classes of trips in the system over a period of time, virtually precluding the ability to analytically evaluate the path travel times. Also, analytical evaluation would call for a correct representation of the various dynamic traffic flow phenomena (queue formation and discharge, congestion build-up and dissipation), a task that is beyond the capability of the state-of-the-art in traffic flow modeling (Peeta and Ziliaskopoulos, 2001).

Despite these aforementioned advantages of realistically capturing traffic dynamics and describing traffic realism, the simulation-based approach typically lacks the ability to study the solution properties (such as existence, uniqueness, stabilities, and adherence to the DUE conditions), and often preclude guaranteed algorithmic convergence. Therefore, in general, while analytical models have served primarily to derive theoretical insights, simulation-based models have successfully tackled many practical aspects that enable deployment in real networks.

Smith (1993) presented a packet-based dynamic traffic model on congested capacity-constrained road networks and showed that (at least) a DUE exists if this model is used to determine path costs, which were shown to depend continuously on path inflows. Tong and Wong (2000) applied a traffic simulator similar to Smith's (1993) model to develop a simulation-based predictive UEDTA model. Peeta and Mahmassani (1995) developed a DTA model using a mesoscopic traffic simulator (Jayakrishnan et al., 1994a; Mahmassani, 2001), as part of an iterative DUE algorithm. Based on the microscopic traffic simulator – MITSIM, Ben-Akiva et al. (1997) also developed a simulation-based DTA model for generating route guidance information. Recently, some studies have tried to incorporate a macroscopic traffic model in a DTA framework. Lo and Szeto (2002), for instance, embedded the Cell Transmission Model (CTM; Daganzo, 1994 and 1995a) in their DUE model to determine path travel costs. Ziliaskopoulos and Waller (2000) also developed a simulation-based DTA model, in which the traffic simulator (RouteSim) employed the CTM for traffic propagation.

The type of (heuristic) solution method, which has been widely used in conjunction with simulation-based DTA models, is the method of successive averages (MSA) or similar (adaptive) averaging schemes (Magnanti and Perakis, 1997a and 1997b). To circumvent the need to explicitly determine the move size in each iteration using some line search method, the MSA, which uses pre-determined step sizes satisfying the following conditions: $\sum_{k=1}^{\infty} \rho_k = \infty$ and $\sum_{k=1}^{\infty} \rho_k^2 < \infty$, is generally adopted in the step of path assignment updating in the simulation-based DUE solution algorithms (e.g., Peeta and Mahmassani, 1995). The most common form of the MSA is to set the step size equal to the reciprocal of the iteration counter: $\rho_k = 1/k$. Figure 2.1 presents the flow chart of the simulation-based UEDTA algorithm proposed by Peeta (1994).

Satisfactory computational experience has been reported with the MSA in some simulation-based models (e.g. Tong and Wong, 2000). However, the MSA does not guarantee descent (or improvement in the objective function) at every iteration (Bertsekas, 1995). It also uses an across-the-board step size for updating path assignments, so the degree to which the path flows deviate from DUE conditions is not taken into account for different OD pairs and departure intervals. This may lead to slow convergence or even failure to converge for some problem instances. More recently, Sbayti et al. (2006) proposed an efficient vehicle-based implementation of the MSA that uses a sorting technique in updating vehicles path assignments based on some path travel attributes (e.g. trip time). The computational results on some large real networks demonstrate the algorithm being able to improve the convergence and the solution quality in terms of a gap measure.

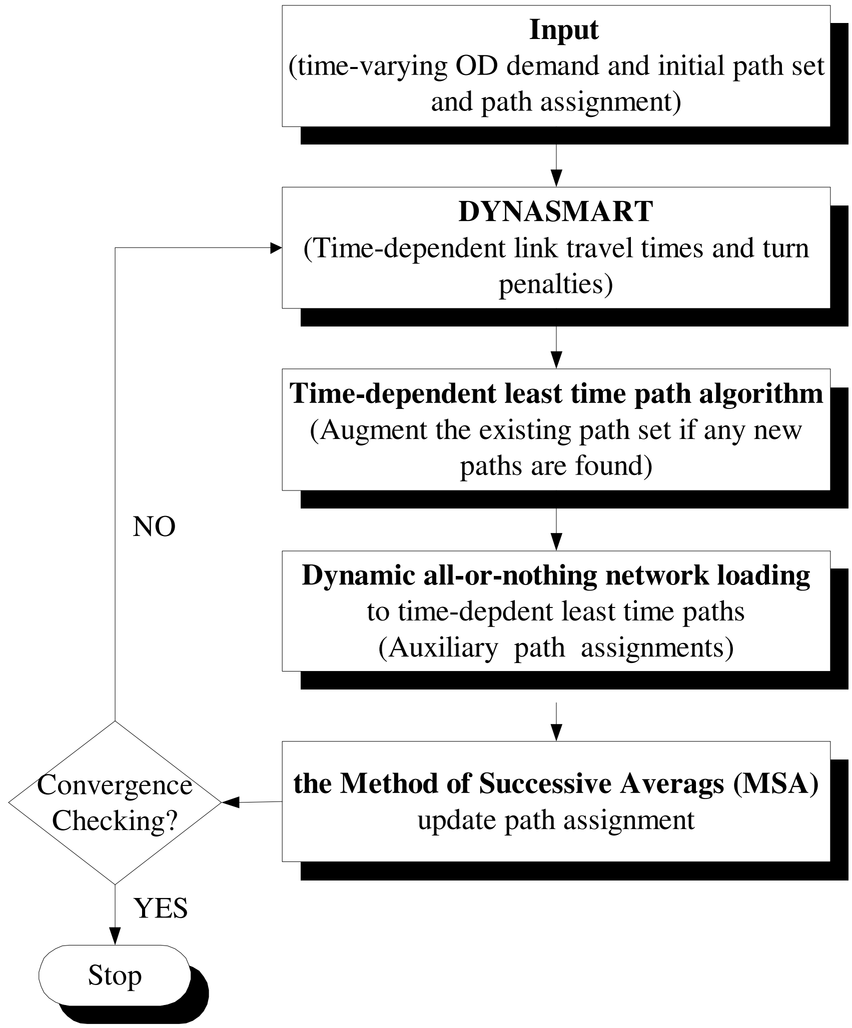



Figure 2.1 The simulation-based UEDTA algorithm by Peeta (1994)

## 2.2 The BUE Traffic Assignment Problem

The BUE traffic assignment problem departs from the classic single-criterion UE traffic assignment problem by explicitly considering the fact that trip-makers have different VOT preferences (i.e. individual users would value differently for a unit travel time saving). This has mainly been accomplished in the literature by assuming random coefficients or VOT in the underlying path choice decision framework where trip-makers are assumed to use paths that simultaneously optimize the two main path travel attributes:

time and cost. This section first reviews (discrete) path choice models that consider user heterogeneity in terms of VOT and how the (random) VOT distributions are derived in the literature. Then the attention turns to BUE traffic assignment models that incorporated random coefficient discrete path choice models or allow the VOT to be a random variable distributed among trip-makers in a network. Note that there is not any DTA model (for dynamic road pricing applications) that explicitly takes into account the user heterogeneity of path choice decisions in response to toll charges, at least to the author's latest knowledge,. Thus, the models and algorithms discussed in this section are all in the static traffic assignment regime.

### 2.2.1 Path choice models that consider user heterogeneity in VOT

The theory of discrete choice model (see e.g. Ben-Akiva and Lerman, 1985) has been widely applied in the studies of trip-makers' path choices. Discrete path choice models generally assume that, in a (random) disutility minimization decision framework, each trip-maker *n* would choose a path *p* that minimizes his/her own perceived disutility. A simple form of this disutility could be specified, for instance, as the following.

$$U_{np} = \beta_p^0 + \beta^c c_p + \beta^t t_p + \beta^X X + \varepsilon_{np}, \forall p \in P \tag{2.2.1}$$

where $\beta_p^0$ is the alternative specific constant; $c_p$ and $t_p$ are the travel cost and travel time, respectively, of path *p*; *X* is the vector of additional observed attributes of individual and of path *p*; $\varepsilon_{np}$ denotes the influence of unobserved factors affecting the utility of path *p*; $\beta_p^0$ (scalar), $\beta^c$ (scalar), $\beta^t$ (scalar) and $\beta^X$ (vector) are the set of coefficients to be estimated. The VOT of trip-maker *n* can be derived as

$$VOT_n = \frac{\partial U_{np} / \partial t_p}{\partial U_{np} / \partial c_p} = \frac{\beta_t}{\beta_c} \qquad (2.2.2)$$

Since constant time and cost coefficients are assumed, this trade-off ratio would be the same for the entire specified group of the population. The well-known multinomial logit (MNL) path choice model can be obtained by assuming that $\varepsilon_{np}$ is independent and identically distributed (IID) with Gumbel distribution, and has the following path choice probability function for trip-maker $n$ choosing path $p$:

$$Prob(n, p) = \exp(U_{np}) / \Sigma_{q \in P} \exp(U_{nq}) \qquad (2.2.3)$$

The MNL model was found to have the following limitations due mostly to the IID hypothesis on the error terms $\varepsilon_{np}$.

- MNL can only handle deterministic taste variations; coefficients (or parameters) have to be constant.
- MNL can not account for correlations in repeated choice observations.
- MNL implies proportional substitution patterns and exhibits the property of independence from irrelevant alternatives (IIA).

To overcome these limitations, some alternatives models, such as multinomial probit (MNP, e.g. Daganzo, 1979) and mixed logit (ML, e.g. Brownstone and Train, 1999; McFadden and Train, 2000) have been proposed in the literature. Among them, the ML (or random parameter logit) is currently regarded as the most flexible and computationally practical discrete choice specifications, providing a convenient approximation to the MNP (Hensher, 2001a). To capture both observed and unobserved user heterogeneity (or equivalently to allow for random taste variations) in the ML model, one could specify some parameters (including alternative specific constants) to be random parameters with both mean and variance estimated together, and re-write the disutility in (2.2.1) as follows:

$$U_{np} = \beta_{np}^0 + \beta_n^c c_p + \beta_n^t t_p + \beta_n^X X + \varepsilon_{np}, \forall p \in P \tag{2.2.4}$$

where $\beta_{np}^0 = \overline{\beta}^0 + \xi_{np}$, $\beta_n^c = \overline{\beta}^c + \zeta_n^c$, $\beta_n^t = \overline{\beta}^t + \zeta_n^t$, and $\beta_n^X = \overline{\beta}^X + \zeta_n^X$. Observed heterogeneity is captured by the variables: $\overline{\beta}^0$, $\overline{\beta}^c$, $\overline{\beta}^t$, and $\overline{\beta}^X$, while unobserved heterogeneity is captured by random terms: $\xi_{np}$, $\zeta_n^c$, $\zeta_n^t$, and $\zeta_n^X$. These random terms can take a number of pre-specified distributions, such as normal, lognormal, or triangular.

Let $\beta$ be the vector of (random) parameters (i.e. $\beta_{np}^0$, $\overline{\beta}^c$, $\overline{\beta}^t$, and $\beta_n^X$), $\theta$ be the true parameters of the distributions, and $f(\beta \mid \theta)$ be the density function. The (unconditional) mixed logit probability can be expressed as the integral of standard MNL probabilities (2.2.3) over a density of parameters (McFadden and Train, 2000):

$$P_{np}(\theta) = \int prob(n, p, \beta) \times f(\beta \mid \theta) d\beta \tag{2.2.5}$$

The vector of unknown parameters is then estimated by the maximum likelihood method (or maximizing the log-likelihood function of (2.2.5).

$$\max LL(\theta) = \max_\theta \sum_{n=1}^{N} \ln P_{np(n)}(\theta) \tag{2.2.6}$$

where $p(n)$ is the path selected by trip-maker $n$.

The main difficulty of solving (2.2.6) is the evaluation of (2.2.5) for each individual $n$ (because it requires the computation of one multidimensional integral for each individual). By normalizing all parameters with the cost parameter, Ben-Akiva (1993) allowed the obtained VOT parameter to follow a lognormal distribution, and used a Gauss-Hermite quadrature to evaluate the integral. This approach is impractical because the dimension of the integrals to be evaluated increases with the number of random parameters (Algers et al., 1998). Recent advances in econometrics (McFadden and Train, 2000; Train, 2003) have suggested the use of simulation estimation techniques for approximating the choice probabilities in (2.2.6). The value of $P_{np(n)}(\theta)$ is replaced by a simulation estimate by sampling over $\beta$. Maximization in (2.2.6) is then conducted on the simulated log-likelihood function (i.e. maximum simulated likelihood).

This mixed logit technique has been widely applied by many researchers to estimate the distribution of VOT with revealed preference (RP) data (e.g. Lam and Small, 2001; Cirillo and Axhausen, 2006), stated preference (SP) data (e.g. Algers et al. 1998; Hensher, 2001a and 2001b), or combined RP and SP data (e.g. Small et al. 2005; Brownstone and Small, 2005). In all cases, significant improvements in model fit were obtained when random (time and cost) parameters or VOT was allowed. General conclusions obtained from these studies include: standard MNL tends underestimate the VOT (Hensher, 2001a and 2001b); motorists exhibit high VOT and substantial heterogeneity in the VOT (Small et al. 2005); VOT estimated with RP data is significantly larger than that estimated with SP data (Brownstone and Small, 2005).

Some examples of incorporating the discrete path choice model in the (static) traffic assignment model are as follows. Nielsen et al. (2002) presented a large-scale stochastic traffic assignment model considering several classes of passenger cars (with different trip purposes), vans and trucks, each with its own utility function on which path choices are based. Their utility functions include random coefficients estimated on SP data in a mixed logit model.  A number of alternative specifications of random coefficient utility functions were estimated and calibrated, and the resulting distributions of VOT are discussed. A similar logit-based path choice model with explicit choice of toll facilities was used in a network model proposed by Florian (2006) for analyzing toll highways, but constant coefficients were assumed in his model.

The other type of path choice modes is based on the generalized path cost (time) function in which path travel time (path cost) is weighted by a trip-maker's VOT, representing how much money the trip-maker is willing to tradeoff for unit time saving.

$$GC_p(\alpha) = c_p + \alpha \times t_p \qquad (2.2.7)$$

where $GC_p(\alpha)$ is the generalized path cost of path p perceived by a trip-maker with VOT equal to $\alpha$. Dial (1979) and Cantarella and Binetti (1998) considered a random VOT in their path choice models based on the generalized path cost functions (2.2.7).

### 2.2.2 The multi-class approach with a discrete VOT random variable

Previous studies in the static traffic assignment context that address the user heterogeneity can be classified into two categories.

The first category is the multi-class approach, in which the entire feasible VOT range is divided into several predetermined intervals according to a discrete VOT distribution or some trip-related or socio-economic characteristics, such as trip purpose or income. In an elastic demand multi-class network equilibrium model proposed by Yang et al. (2002), the feasible range of VOT is divided into a predetermined number of intervals of equal length based on different income levels; the entire population of trips is segmented accordingly into different groups with corresponding group-specific demand functions. Network users are assumed to minimize their individual generalized trip cost, and thus divide themselves among the various paths differentiated based on travel time and monetary cost. The elastic demand multi-class network equilibrium model developed to describe route choices of heterogeneous users is based on Beckmann's transformation (1956) using link flows as decision variables. Kuhn-Tucker conditions characterizing optimal solutions are derived, and the classic Frank-Wolfe algorithm was extended to solve the problem. In the numerical experiments conducted on a small simple test network, they compared and contrasted the outcomes with the case of a single average VOT, and investigated how the VOT distribution affects traffic flow and profit forecasts of private toll roads. The numerical results highlighted the importance of incorporating user heterogeneity in private toll road modeling.

Nagurney and Dong (2002) developed a multi-class, multi-criterion traffic network equilibrium model with elastic demand in which travelers of a class perceived their generalized path cost as a weighing of travel time and monetary cost, both of which are flow dependent. The weighting parameter (i.e. VOT) was considered as not only class-dependent but also link-dependent. They also allowed the demand function for each class to be OD-dependent. The problem was formulated as a finite dimensional VI with some qualitative analyses of equilibrium solutions (e.g. existence and uniqueness), and solved by the modified projection method of Korpelevich (1977). Other examples of this approach can be found in Mekky (1995 and 1997) and Yang and Huang (2004).

### 2.2.3 The approach with a continuous VOT random variable

The second category considers VOT to be continuously distributed across the population of trips. Leurent (1993) was among the first to propose a cost versus time (CVT) network equilibrium model for road pricing applications, defining such equilibrium is achieved when every trip-maker is assigned a path that minimizes his/her own generalized cost. CVT equilibrium solutions are characterized by the solutions of an extremal convex problem. Conditions for the existence and uniqueness of CVT solutions were also provided and proved. The method of successive averages (MSA) was adapted to solve for the CVT equilibrium. Numerical results based on a two-link (single OD pair) network demonstrated a significant difference in link (or path) flows between the single VOT model and the CVT model.

Dial (1996) developed a static bi-criterion user equilibrium traffic assignment model with continuous VOT to forecast path choice and associated total arc flows in the presence of tolled alternatives. The path choice behavior assumption made is that a traveler chooses a path $p$ that minimizes its perceived generalized cost (i.e. Eq.(2.2.4)). Under this assumption, in the case of a continuous VOT, only paths corresponding to extreme efficient points on the efficient frontier (EF) represent rational path choices; that is travelers are only assigned to those extreme efficient paths in the disutility minimization path choice framework. To find the EF is equivalent to finding minimum generalized cost paths for an appropriate set of VOTs. The bi-criterion user equilibrium (BUE), or so-called T2-ETA in Dial's paper, is a generalization of Wardrop's principle (1952), and states that each trip uses only paths that minimize their particular perceived generalized costs. With the VOT continuously distributed across the population, this generalization of classic UE would admit a large, probably infinite, number of categories (classes) of trips in simultaneous equilibrium. Thus, in Dial's work, only the total flow on each arc was concerned; note that the individual class arc flows are not unique. Dial also showed that this model essentially can be reduced to a (infinite dimensional) variational inequality (VI) problem, which then permits the application of existing VI algorithms, such as the generalized Frank-Wolfe algorithm (Magnanti and Perakis, 1993a and 1993b). An efficient solution algorithm, named restricted simplicial decomposition (RSD), based on the simplicial decomposition method (Lawphongpanich and Hearn, 1984) was developed in a subsequent paper (Dial, 1997) to solve the BUE problem. One of the important components of the RSD is the minimum path assignment (MPA) algorithm that finds the set of extreme efficient paths and assigns the corresponding share of trips to each of the paths.

Note that Leurent's CVT equilibrium model considered elastic demand and allowed only one criterion (travel time) to be flow dependent; while Dial's model assumed fixed demand and allowed both criteria to be flow dependent. Additionally, the CVT model is a finite dimensional approach that takes path flows as variables unlike Dial's infinite-dimensional model that uses link flows as variables.

Marcotte and Zhu (1997) considered the problem of determining an equilibrium state resulting from the interaction of infinitely many classes of customers, differentiated by a continuously distributed class-specific parameter. Solutions to the infinite dimensional VI problem, with link flows as the decision variables, were used to describe the equilibrium and obtained by a linearization algorithm, an infinite dimensional extension of the Frank-Wolfe algorithm. Marcotte (1999) further presented several VI formulations of the bi-criterion equilibrium model, suggesting that (1) all the proposed formulations can be incorporated into a unified algorithmic framework that iteratively solves the parametric shortest path problem and performs a line search in the descent direction; (2) solving the parametric shortest path problem approximately, by selecting parameter values in a suitable manner, could allow solving the bi-criterion assignment problem as efficiently as the single criterion problem.

## 2.3 The BSP Problem

Given a directed network $G$ ($N$, $A$), where $N$ is the set of nodes and $A$ is the set of directed links. Each link $(i, j) \in A$ is associated with two attributes: $d_{ij}$ and $c_{ij}$, where for simplicity $d_{ij}$ and $c_{ij}$ are assumed the time and cost to traverse link $(i, j)$, respectively. The objective of solving the bi-criterion (or bi-objective) shortest path (BSP) problem is to find a shortest path from the origin node $r \in N$ to the destination node $s \in N$ that simultaneous optimizes both travel attributes. Mathematically, the BSP problem is given as follows:

(BSP) Minimize $\sum_{(i,j)\in A} c_{ij} x_{ij}$ (2.3.1)

Minimize $\sum_{(i,j)\in A} d_{ij} x_{ij}$ (2.3.2)

Subject to

$$\sum_{\{j|(i,j)\in A} x_{ij} - \sum_{\{j|(j,i)\in A} x_{ji} = \begin{cases} 1 & \text{if } i = r \\ 0 & \text{if } i \neq r, s \\ -1 & \text{if } i = s \end{cases} \tag{2.3.3}$$

$$x_{ij} \in \{0,1\}, \forall (i, j) \in A \tag{2.3.4}$$

Because it is generally impossible to find a unique optimal path in terms of all the objectives in a general network, the BSP problem usually aims to find a set of efficient (Pareto optimal or non-dominated) paths. Let $p(r,s)$ be a feasible path starting from a given origin $r$ to a destination $s$, and TT($p$) and TC($p$) the travel time and travel cost, respectively, associated with path $p$. Let $P(r,s)$ be the set of all feasible paths $p(r,s)$ for a given $(r,s)$. For simplicity and clarity, we denote $P = P(r,s)$.

**Definition 2.1** A path $p \in P$ is efficient (Pareto Optimal or non-dominated) if and only if it is not possible to find a different path $q \in P$ such that $TT(q) \leq TT(p)$ and $TC(q) \leq TC(p)$ with at least one strict inequality.

Let $P^e = P^e(r,s)$ be the set of efficient paths for a given $(r,s)$. An efficient path $p \in P^e$ in the solution space corresponds to an efficient point (or vector) $Z(p) = [TT(p), TC(p)]$ in the criterion space. Accordingly, the set of efficient points is denoted as $\Omega^e$.

Hansen (1980) proved that the BSP problem is NP-hard by showing that there exists a family of graphs for which the number of efficient paths grows exponentially with the number of nodes in the network (i.e. intractable). He showed that listing these paths requires an exponential number of operations and so no polynomial behavior is expected. Garey and Johnson (1979) also showed this problem is NP-hard by transforming from a 0-1 knapsack problem.

### 2.3.1 Algorithms for finding efficient paths

Various algorithms have been developed in the literature for finding the set of efficient paths in solving the BSP problem. According to Skriver and Andersen (2000), those algorithms can be generally classified into the node-labeling approach and the path/tree handling procedure.

Hansen (1980) was among the first to propose an algorithm for solving the BSP problem. As an extension of the modified (Dijkstra) label-setting algorithm, his algorithm is a multi-labeling approach in which each node in the graph is associated with a vector of quadruplets such that each quadruplet refers to one of the paths in the efficient path set at that node. The first two elements in a quadruplet are the values of the optimized attributes, and the last two give the required information to backtrack the efficient path found. As a result, the functional equation of updating nodes labels is extended from scalar functions to vector valued functions, and the standard minimization performed at each node is replaced by dominance checking step (i.e. removal of dominated labels). The selection step was also modified by choosing the lexicographically smallest label in the set of all labels. By implementing balanced tree to store the quadruplets at each node, the algorithm has a complexity of $O(nmD \times \log(nD))$, where $n$ and $m$ are the number of nodes and the number of links in the network, respectively, and $D$ = max. $d_{ij}\ \forall (i, j) \in A$. Similar to the standard single-objective label setting algorithm, one label is labeled permanently in each iteration. By choosing the lexicographically smallest label, it is ensured a non-dominated label. In this perspective, it can be seen that a non-dominated path uses only non-dominated sub-paths. The generalization of Hansen's (1980) algorithm to multiple criteria was proposed by Martins (1984).

Following the same multi-labeling approach, several implementation of a general label correcting algorithm for the BSP problem are tested and compared in a paper by Brumbaugh-Smith and Shier (1989). As in the case of Hansen's work (1980), the complexity of the algorithm ($O(mn^2D^2)$) is bounded by the network characteristics and the size of the data. They found that the CPU-times depend heavily on the way the different label sets are scanned and deleted. The worst principle LIFO (last in first out) is more than a factor 10 slower than the fastest principle FIFO (first in first out). Skriver and Andersen (2000) introduced some improvements that can be implemented with Brumbaugh-Smith and Shier's (1989) algorithm, which was consider the most efficient algorithm for solving the BSP problem. Specifically, their first improvement lies in a fast predomination check, which rules out expensive edges by considering the present set of labels at each node. This condition can be implemented through using initialization with Dijkstra's shortest path method to set upper bounds on all labels at all nodes or to set bounds during the routine. The second improvement is based on initializing node information from the terminal node in order to find the cheapest and fastest paths from an intermediate node to the terminal node. With these two improvements, their modified label-correcting algorithm can reduce considerably the number of iterations and CPU-times needed to find all the efficient paths in the network.

This multi-labeling label-correcting approach for solving the BSP problem was extended by Miller-Hooks and Mahmassani (1998) to the routing of hazardous materials in stochastic time-varying networks. Abdelghany (2001) applied the same approach to solve the time-dependent multi-criterion shortest path problem in his dynamic multimodal trip assignment model. Both of their algorithms are based on the efficient time-dependent single-criterion label-correcting algorithm proposed by Ziliaskopoulos and Mahmassani (1993).

The k-shortest path algorithm has also been used to solve this problem. In this case, the efficiency of the BSP algorithm depends on the efficiency of the k-shortest path algorithm and on the number of paths (*k*) to be generated to determine all the elements in the efficient path set. The idea of using k-shortest path algorithm is that paths are being determined by non-decreasing order of one of the attributes until a well-determined lower bound for the other attribute is achieved. The lower bound is determined such that all the efficient paths are determined. Climaco and Martins (1982) gave examples to show the use of the k-shortest path algorithm in determining the non-dominated paths set. The algorithm is initialized with the determination of the cheapest path and the fastest path. The cheapest path is the first element in the efficient path set. The fastest path is used to get an estimate for the value of the bound to be used for the other (cost) attribute. The k-shortest paths are sequentially computed by relaxing the cost criterion, each time finding the best path with respect to the time criterion, until the specific bound on the cost criterion is reached. The dominated paths are excluded from the set to obtain the set of Pareto-optimal paths. In the worst case, this procedure has the danger of enumerating all possible paths from a source to a destination. Thus, a terminal value of K equal to (n–1)! could occur, resulting in an exponential increase in the computational effort.

Since the BSP problem is known to be NP-hard (Hansen, 1980), the computational time and storage requirements for finding a complete set of efficient solutions increase exponentially with the size of problem. Alternatively, many approximation algorithms have been developed to find a subset of efficient paths within limited computational resources. Warburton (1987) introduced the concept of $\varepsilon$-

domination (or $\varepsilon$-approximate) to quantify the degree of accuracy in approximating trade-off curves and surfaces in a multiple criteria space.

**Definition 2.2** A path $p$ $\varepsilon$-dominates path $q$ if $\frac{TT(p)}{TT(q)} \leq 1+\varepsilon$ and $\frac{TC(p)}{TC(q)} \leq 1+\varepsilon$. In this case, path $p$ is considered $\varepsilon$-efficient vis-à-vis path $q$.

Note that, when $\varepsilon = 0$, this definition reduces to the common notation of vector dominance. Based on rounding and scaling techniques, he also developed a fully polynomial $\varepsilon$-approximate algorithm subject to a desired degree of accuracy. In Hassin's study (1992), the $\varepsilon$-approximate concept is combined into a binary search scheme, which iteratively adds the new weighting breakpoints, reducing the approximation error at intervals. Nielsen (2003) applied a similar approximation scheme for solving the bi-criterion shortest hyper-path problem in random time-dependent networks under a priori and adaptive route choice strategies. Mahmassani et al. (2005) also incorporated the $\varepsilon$-approximate algorithm in a binary search framework to find an approximate subset of efficient paths in time-dependent networks.

### 2.3.2 Algorithms for finding extreme efficient paths

Henig (1985) introduced the concept of extreme (or supported) efficient paths, which correspond to extreme points in the boundary (so-called efficient frontier) of the convex hull containing all the efficient points in the criterion space. The following definition is given according to Henig (1985).

**Definition 2.3** A path $p^* \in P$ is an extreme (or supporter) efficient path with respect to the parameter $\lambda \in [0,1]$ if $\lambda \times TC(p^*) + (1-\lambda) \times TT(p^*) < \lambda \times TC(p) + (1-\lambda) \times TT(p)$ for all $p \in P$ such that $(TC(p),TT(p)) \neq (TC(p^*),TT(p^*))$.

Let $P^{ex} = P^{ex}(r,s)$ be the set of extreme efficient paths for a given $(r,s)$. The size of the extreme efficient paths is unknown a priori and in the worst case, it may equal to the number of all the elements in the efficient path set. However, it is usually expected to be significantly less than the size of the efficient path set. Figure 2.2 depicts the criterion space formed by the two criteria – travel time (TT) and travel cost (TC) and the corresponding efficient frontier. $Z_1$, $Z_2$, $Z_3$, and $Z_4$ are extreme efficient points on the efficient frontier. An efficient path corresponds to one of these four points is an extreme efficient path. $Z_6$ and $Z_9$ are (non-extreme or unsupported) efficient points. $Z_5$, $Z_7$, and $Z_8$ are dominated points. These extreme efficient paths are very important, as in the disutility minimization-based traffic assignment framework with convex utility functions all trips are distributed only among the set of extreme efficient paths (Dial, 1996; Marcotte and Zhu, 1997).

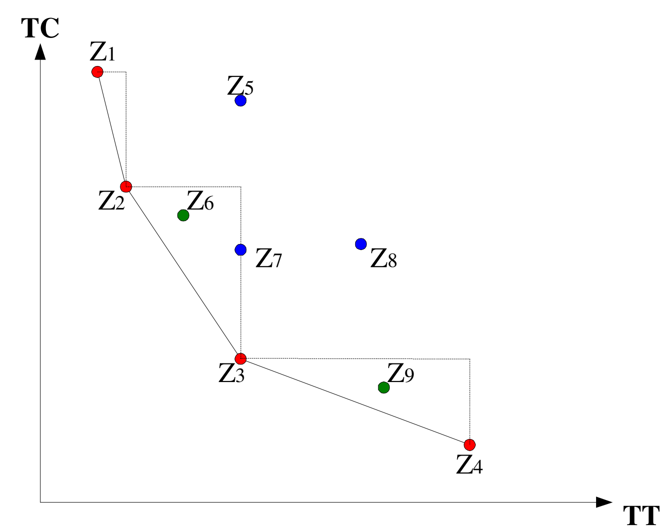



Figure 2.2 The criterion space and efficient frontier

One common method for finding the set of extreme efficient paths is the weighting method. According to Definition 2.3, it is equivalently to say that $P^{ex}$ is a subset of the efficient paths that includes all those which minimize convex combinations of the two objectives. Define the combined objective $W(x,\lambda)$ as follows:

$$W(x,\lambda) = \sum_{(i,j)\in A} \lambda \times c_{ij} x_{ij} + (1-\lambda) \times d_{ij} x_{ij} \text{ , } 0 < \lambda < 1 \qquad (2.3.5)$$

where $x$={ $x_{ij}, \forall (i,j) \in A$ }. The function $W(x,\lambda)$ is a convex combination (or weighted sum) of the two objective function (2.3.1) and (2.3.2). Optimizing this function with different $\lambda$'s will give the extreme efficient solutions (i.e. paths). This method is often referred to as the weighting method. The weighting method combines different attributes into a single utility function and systematically varies the weights (i.e.$\lambda$'s) to generate extreme efficient solutions. This method has been widely applied to solve the BSP, since it can utilize efficient solution algorithms developed for the single-objective shortest path problem. In the recursive weighting algorithm presented by Dial (1979), a weighting parameter is iteratively generated based on the slope of the line connecting the two extremes (i.e. the trade-off between two solutions), and the search process is repeated recursively until all (or a given number) of the efficient paths are found. It should be note that, as was also pointed out by Henig (1985), the weighting method can only enumerate the extreme efficient paths because non-extreme efficient solutions are dominated by a convex combination of extreme efficient solutions. This can be illustrated in Figure 2.2. The four points $Z_1$, $Z_2$, $Z_3$, and $Z_4$ are solutions to the problem consisting of (2.3.5), (2.3.3), and (2.3.4), with respect to different values of $\lambda$. The three triangles are non-dominated areas defined by the four points. The points inside one of the triangle areas (i.e.

$Z_6$ and $Z_9$) are non-dominated in the problem of (2.3.1)-(2.3.4), but dominated in the problem of (2.3.5), (2.3.3), and (2.3.4). Therefore, we have to search the (non-extreme) efficient paths in between the extreme efficient paths, if the weighting method is used to solve the BSP problem (2.3.1)-(2.3.4). For instance, Mote et al. (1991) developed a two-stage approach to solve the BSP problem, where the parametric analysis was applied in the first stage to solve the problem of (2.3.5), (2.3.3), and (2.3.4) to obtain all the extreme efficient paths, and then in the second stage a label-correcting-based algorithm was used to find the (non-extreme) efficient paths in between the present extreme efficient paths.

Heing (1985) also suggested several algorithms based on the parametric analysis of $\lambda$ to generate these extreme efficient paths. The parametric analysis approach generates a finite sequence $\{\lambda_k: \lambda_0 = 0 < \lambda_k < \lambda_K = 1\}$ along with the list $\{(TC_k, TT_k)\}$, where K is the number of extreme efficient paths and $(TC_k, TT_k)$ is the (criterion) value of the extreme efficient path with respect to every $k \in (\lambda_k, \lambda_{k+1})$. The parametric analysis method solves the expanded BSP problem from an origin node $s$ to all other nodes in the network and was also applied by several researchers such as Mote et al. (1991), and Dial (1997), to efficiently identify extreme efficient shortest paths.

## 2.4 The SRDUE Traffic Assignment Problem

While the DUE and BUE problems discussed in sections 2.1 and 2.2 consider trip-makers choose only least time or least generalized cost paths to avoid congestion by assuming the time-varying OD demands are fixed and known a priori (i.e. fixed departure times), the simultaneous (or joint) route and departure time user equilibrium (SRDUE) traffic assignment problem allows trip-makers to adjust their departure times, in addition to switching routes (or paths), in response to changes of network conditions. Indeed, the SRDUE problem admits a more realistic generalization of the static UE problem than the DUE problem, as the underlying behavioral assumption is consistent with the observation that trip-makers are more inclined to adjust their departure times than to switch paths, in order to avoid traffic congestion (e.g. Hendrickson and Plank, 1984; Mahmassani and Chang, 1985). The importance of incorporating departure time choices into a DUE model is well recognized in the literature and the SRDUE problem has been extensively studied in the past two decades.

### 2.4.1 SRDUE models and solution algorithms

Several early studies in 1980's focused on determining analytically equilibrium conditions resulting from network users' departure time choices and time-dependent equilibrium arrival and departure patterns at a single bottleneck in the morning commute. They assumed that a group of commuters must pass a bottleneck with fixed capacity in order to reach their workplaces and each traveler has certain time (i.e. preferred arrival time, PAT, or desired arrival time, DTA) at which he or she would like to pass the bottleneck in order to be at work on time. Hendrickson and Kocur (1981) were among the first to introduce the notion of schedule delay (the difference between actual and desired arrival times) to the network equilibrium model. They developed a simple analytical model that determines the time-dependent user equilibrium conditions (i.e. trip costs are equal for all travelers) and corresponding departure time patterns in a deterministic setting based on the user equilibrium concept that assumes each traveler to select the departure time (route choice is fixed) with minimum sum of travel time and schedule delay. Deterministic queueing theory was applied to study the equilibrium arrival and departure distributions at a bottleneck. De Palma et al. (1983) incorporated a probabilistic departure time choice model of the continuous logit form within the framework of Hendrickson and Kocur (1981). Newell (1987) generalized the framework of Hendrickson and Kocur (1981) to consider the more realistic situations in which non-identical travelers have different PATs and may attach different values to their schedule delays. These models were considered as pure departure time choice models because travelers' route choices were assumed fixed.

By extending Hendrickson and Kocur's framework (1981), Mahmassani and Herman (1984) proposed a network equilibrium model with joint departure time and route choices. Instead of using deterministic queueing theory in modeling arrivals and departures at a bottleneck, they applied a linear (Greenshield's) speed-density traffic flow relationship to describe congestion along routes connecting a single origin and a single destination. With the relatively realistic elementary traffic flow relationship incorporated in their model, the interrelation between user decisions and system performance is taken into account, and time-varying network performance indicators, such as link densities and speeds, can be obtained. Another network equilibrium model with joint departure time and route choices is the dynamic model of peak period traffic congestion with elastic (demand) arrival rates developed by Ben-Akiva et al. (1986). In addition to describing delays at bottlenecks by a deterministic queueing model, they used a nested logit choice hierarchical structure to model the join departure time and route choices with elastic demand. An important feature of their work is that the day-to-day adjustment of the temporal and spatial distribution of traffic is derived from a dynamic Markovian model with a set of nonlinear differential equations. The model was used to perform simulation experiments and analyze the impact of alternative pricing policies and preferential treatment of high occupancy vehicles. Arnot et al. (1990) applied a joint departure time and route choice user equilibrium model with deterministic queueing bottlenecks to systematically analyze various pricing regimes. An important finding was that step tolls generally yield greater efficiency gains than uniform tolls because the former reduce queueing delays by altering travelers' departure times (i.e. route split is not very sensitive to the choice of toll regime). Although the aforementioned models were restricted to one origin, one destination and parallel non-interacting routes, each of which has a bottleneck that causes delays, it provides valuable theoretical insights in modeling travelers' joint (or simultaneous) route and departure time choices in the traffic assignment context.

The first formulation that truly accounts for simultaneous route choice and time-departure decisions on general dynamic networks is due to Friesz et al. (1993). They defined user equilibrium conditions for simultaneous route choice and departure time decisions as the following:

> *If, for each OD pair, the actual flow unit costs from time of departure to time of arrival on utilized paths, including any early or late arrival penalties, are identical and equal to the minimum unit path cost which can be realized from among all route choice and departure time decisions, the corresponding flow pattern is said to be a simultaneous route-departure (SRD) user equilibrium.*

The SRDUE problem was formulated as an infinite-dimensional variational inequality (VI) problem because they applied the continuous time representations of path costs and path flows. In spite of being able to describe the SRDUE problem on general networks, the infinite-dimensional VI formulation provided no obvious way of designing a solution algorithm, other than solving a system of simultaneous integral equations. Moreover, applying inverse exit time function to determine link travel costs (or times) make it more difficult to solve the infinite-dimensional VI formulation. To eliminate both traffic flow propagation anomalies and the need for inverse exit time functions, Bernstein et al. (1993) proposed an alternative formulation of the SRDUE problem. The formulation was referred to as the variation control (VC) problem in their paper, because it is similar in form to an optimal control problem, the difference being that the objective function is replaced by an inequality. In this VC problem formulation, the control variables are the path departure rate, and the state variables are the arc occupancies. Rather than using inverse (link) exit time functions, each arc is modeled as a deterministic queue and associated with a delay function of this form. Algorithms for solving VI problems can be employed to solve VC problems, as the latter are just a special case of the former. In general, those algorithms approximate original VI problems and use heuristics to solve the approximate problems. Bernstein et al. (1993) presented an approximation of the VC problem in deterministic queueing networks, and described a path-swapping heuristic, which is based on the equilibration algorithm (e.g. Dafermos and Sparrow, 1969) which shifts flows from the best alternative (route-departure time combination) to the worst, for solving the approximate problem.

Unlike the above two continuous time infinite-dimensional VI or VC formulations, Wie et al. (1995) proposed a discrete time finite-dimensional VI formulation of the SRDUE problem, which utilized link exit flow functions and nested cost operators to calculate unit path costs given the departure time and route choices of network users. The advantage of this formulation is that it is computationally tractable. They also showed that, under certain regularity conditions in which link exit flow functions and schedule delay functions are nonnegative and continuous, a discrete time SRDUE is guaranteed to exist. A heuristic solution algorithm based on the route-departure time swapping rule that moves flows from all the other route-departure time combinations to the best one (see section 2.2.1 for the review of similar path-swapping algorithms) was presented and used to solve the SRDUE problem on a small test network with 2 OD pairs.

As a different approach of modeling and solving the SRDUE problem, Janson and Robles (1993) developed a link flow-based bi-level mathematic programming of the dynamic user equilibrium problem with combined path and departure time choices. The upper problem (UP) is a dynamic network loading problem that determines link flows and computed link travel times on the basis of monotonically non-decreasing functions for a given set of shortest paths obtained by solving the lower problem (LP), which is a shortest path linear program based on the link travel times generated from solving the UP. Although the bi-level formulation is non-convex over the solution space, the UP is convex with a unique global optimum for a given set of shortest paths. The solution algorithm presented in their paper iterates between UP with a fixed set of shortest paths and LP with fixed link travel times. They derived the optimality conditions of the bi-level formulation of the SRDUE problem by setting up the Lagrangian of UP with a fixed set of shortest paths. They showed that for a given set of shortest paths the bordered Hessian matrix of the Lagrangian of UP is positive definite (PD), meaning that there is a unique global optimum. The bordered Hessian matrix is PD so long as each link travel time function is a monotonically non-decreasing function of the flow on that link at a given time interval only (i.e. there are not spatial and temporal interactions considered). Note that, similar to the analytical DUE models with fixed departure times reviewed in section 2.1.1, additional flow propagation constraints were used to guarantee the first-in-first-out (FIFO) condition. Huang and Lam (2002) proposed an equivalent zero-extreme value minimization problem to the SRDUE problem on discrete time basis. They presented a time-dependent link travel time function that is able to model point queues in a network and satisfy the FIFO condition, whereas more realistic physical queue phenomena, such as queue build-up, spill-back, and dissipation, are not taken into account. A heuristic solution algorithm which simulates a day-to-day dynamic system was proposed, based on a route-departure time swapping process motivated by Smith and Wisten (1995) and similar to the method by Wie et al. (1995). Huang and Lam (2003) extended this SRDUE model to the multi-class context.

In addition to those analytical models of the SRDUE problem, the simulation-based approach was also considered in the previous studies. While still formulating the SRDUE problem using the discrete time finite-dimensional VI approach, Ziliaskopoulos and Rao (1999) proposed a simulation-based solution algorithm that uses a traffic simulator to determine link and path travel times and assumes the arrival time-based OD demands are available, i.e. travelers' PATs (or DTAs) are known and fixed a priori. In addition to the traffic simulator, the other important component of their solution scheme is a time-dependent shortest (least time) path algorithm with fixed arrival times, which computes optimal paths from all origin nodes to a destination node and for all possible arrival times. They proposed two heuristics that have the potential to meet the SRDUE equilibrium conditions: one emulates iteratively day-to-day dynamic adjustments of departure time and path choices of network user; another is a converging scheme that estimates the equilibrium arc travel times and adjusts the schedule delay penalties in each iteration, so that the system advances toward an equilibrium solution. Szeto and Lo (2004) developed a cell-based SRDUE model with elastic demand. The problem was formulated as a discrete time finite-dimensional VI problem similar to the one by Wie et al. (1995). The cell transmission model (Daganzo, 1994 and 1995a), which can capture (physical) queue spill-back and junction blockage, was encapsulated in their model to determine link travel times for a given path flow pattern. They adopted the descent direction method, developed by Han and Lo (2003) to solve the VI problem. The convergence of this descent direction method is based on the condition that the underlying link travel time mapping (i.e. function) is monotonic (or co-coercive) and the solution set is nonempty.

### 2.4.2 Schedule delay and path-finding algorithms for the SRDUE problem

Most of the aforementioned dynamic network user equilibrium models, analytical or simulation-based, with departure time and/or route choice assume paths travel disutilities (such as time and cost) are additive of link travel disutilities and feature the trip cost (or disutility) function as the weighted sum of path time, path cost, and schedule delay cost. Let $G_{odp}^{\tau}$, $TC_{odp}^{\tau}$, $TT_{odp}^{\tau}$, and $\varphi_{odp}^{\tau}(\theta)$ be the trip cost, path cost, path time, and schedule delay, respectively, experienced by a trip departing from origin $o$ at time $\tau$ to destination $d$ assigned to path $p \in P(o,d)$ with a preferred arrival time $\theta$. The trip cost function can be expressed in the following form.

$$G_{odp}^{\tau} = TC_{odp}^{\tau} + \alpha \times TT_{odp}^{\tau} + \varphi_{odp}^{\tau}(\theta), \tag{2.4.1}$$

where $TC_{odp}^{\tau} + \alpha \times TT_{odp}^{\tau}$ is generally considered as the path generalized cost perceived by trips with VOT $\alpha$. The schedule delay cost is typically defined as the following piece-wise linear function (e.g. Bernstein et al. 1993; Ziliaskopoulos and Rao (1999); Huang and Lam 2002; Szeto and Lo, 2004).

$$\varphi_{odp}^{\tau}(\theta) = \begin{cases} \beta \times [(\theta - \Delta) - (\tau + TT_{odp}^{\tau})] \text{ if } (\theta - \Delta) > (\tau + TT_{odp}^{\tau}), \\ 0 \text{ if } (\theta - \Delta) \leq (\tau + TT_{odp}^{\tau}) \leq (\theta - \Delta), \\ \lambda \times [(\tau + TT_{odp}^{\tau}) - (\theta + \Delta)] \text{ if } (\theta + \Delta) < (\tau + TT_{odp}^{\tau}), \end{cases} \qquad (2.4.2)$$

where $\alpha$, $\beta$, and $\lambda$ are value of time (VOT), value of early schedule delay (VOESD), and value of late schedule delay (VOLSD), respectively (see Figure 2.3).

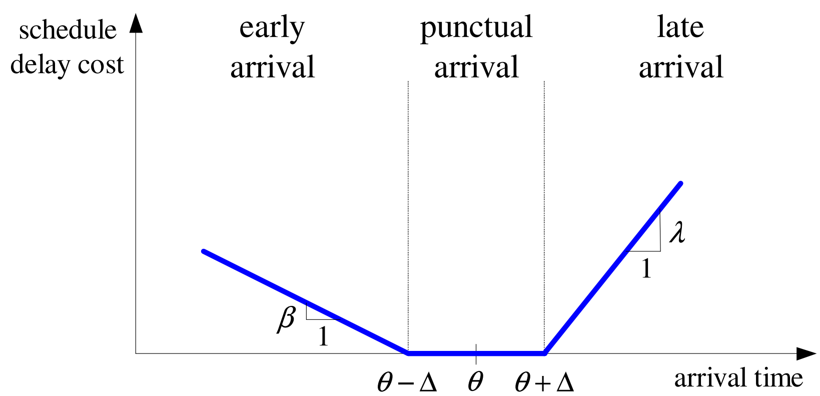



Figure 2.3 The piece-wise linear function of schedule delay cost

In general, by following the empirical results (e.g. Small, 1982) it is assumed that $\lambda > \alpha > \beta > 0$; that is, travelers value the cost of LSD higher than the costs of time and ESD. While most of the previous studies assume identical values of $\theta$, $\alpha$, $\beta$, and $\lambda$ for all travelers in a network, some have recognized that in reality different travelers usually have different $\theta$, $\alpha$, $\beta$, and $\lambda$, because of different work places, socio-economic characteristics, trip purposes, attitudes and inherent preferences. For example, Newell (1987) assumed continuously distributed values of $\theta$, $\alpha$, $\beta$, and $\lambda$ for non-identical commuters, and Szeto and Lo (2004) considered destination-based $\theta$, $\alpha$, $\beta$, and $\lambda$ in their cell-based SRDUE models.

Just as the path flow-based formulation for DUE models, SRDUE models that used path flows as decision variables require a complete set (or a representative subset) of feasible paths on which time-varying OD demands would be distributed. Most of the existing SRDUE models (e.g. Friesz et al., 1993; Bernstein et al., 1993; Wie et al., 1995; Huang and Lam, 2002, Szeto and Lo, 2004) assumed the set of feasible paths are available a priori, and focused only on model formulations and/or equilibration schemes. De Palma et al. (1990) were among the first to explicitly address the issue of finding best paths with penalties for early or late arrivals. They formulated the choice of a best path for a commuter leaving his/her home at a given time and expect to arrive at his/her destination within a given time interval. The travel times along arcs depend on arc flows and the arrival time at the upstream node of the arc. The schedule delay is taken into account by penalizing arrival time outside the desired time interval. The objective function (Z) integrating travel cost (*TC*), travel time (*TT*), and schedule delay (*ESD* or *LSD*) was expressed as the following:

$$Z = TC + \alpha \times TT + \beta \times ESD + \lambda \times LSD \qquad (2.4.3)$$

They defined this problem as the generalized shortest path problem (GSPP). It includes among others the constrained shortest path problem and the shortest path problem with time dependent travel times. The GSPP corresponds to the optimal path of a marginal commuter: the quantities appearing in the objective function are insensitive to the overall set of assignments and are determined by the choice of a commuter. The GSPP algorithm (a A* type algorithm), which was proposed in their paper and finds a path $p$ from origin $v_1$ to destination $v_n$ with minimum $Z$, is summarized in the following.

First Step: Compute for each vertex $v_k$ lower bounds on the cost and the travel time corresponding to the best path from $v_k$ to $v_n$.

For $v_1$ : Find the lower bound on the cost by applying Dijkstra backwards.
Find the lower bound on the time by applying Dijkstra forward.

For $v_k$ : For each arrival time $\tau^i$ (i.e. arrival time-based), apply Dijkstra backwards to obtain the latest departure time $\tau_k^i$ which allows to arrive at $v_n$ at $\tau^i$ . The lower bound on the travel time: $\tau^i - \tau_k^i$.

Second Step: Denote $b(i)$ the lower bound on the objective function corresponding to path $i$ and $e(i)$ estimation of the objective function corresponding to path $i$.

(1) Selection of the path with smallest estimation (i.e. $e(i)$) and test for ending.

(2) Computation of new labels: efficient paths are systematically constructed by labeling vertices which follow immediately the last vertex of a selected path. Dominated sub-paths are eliminated by using the lower bound of the objective function.

Ziliaskopoulos and Rao (1999) formulated the GSPP as a dynamic program and solve it by proposing the time-dependent shortest path algorithm with fixed arrival times (TDSP-FAT), based on the label correcting method. TDSP-FAT algorithm computes the least time paths from all nodes $i$ to destination node $d$ for all arrival time intervals $\tau \in S$. Denote $\boldsymbol{\delta}_i = \{\delta_i(\tau)\}$ the vector of labels $\delta_i(\tau)$, where $\delta_i(\tau)$ be the travel time of a path from node $i$ to destination $d$ that arrives at time interval $\tau$. For each $\delta_i(\tau)$, the corresponding departure time from node $i$ that arrives $d$ at time interval $\tau$ can be determined as $\tau - \delta_i(\tau)$. The updating equation that constitutes the building block of the TDSP-FAT is as follows:

$$\delta_i(\tau) = \min\{\delta_i(\tau), d_{ij}(t_{\max}) + \delta_j(\tau)\}, \forall j \in \Gamma(i), \forall \tau \in S, \qquad (2.4.4)$$

where $t_{\max} = \arg\max\{t + d_{ij}(t) = \tau - \delta_j(\tau), \forall t \in S\}$ and $\Gamma(i)$ is the set of successor nodes of $i$. The TDSP-FAT is based on Bellman's general principle of optimality and operates in a backward fashion: the least time paths are calculated backward by starting from the destination node $d$ and recursively applying the updating equation (2.4.4) to scan all nodes in the scan eligible (SE) list until the list empty. Scanning a node is to update the labels of all predecessor nodes for all (arrival) time intervals. Define $\Gamma^{-1}(i)$ is the set of predecessor nodes of $i$. The TDSP-FAT algorithm is formally stated as follows.

Step 1: Initialization

1.1 Initialize the label vectors as the following:

$\boldsymbol{\delta}_d = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$; $\delta_i(\tau) = \infty$, $\forall i \in N$, $\forall \lambda \in S$.

1.2 Create the SE list and insert into it the destination node $d$.

Step 2: Scanning and updating labels

2.1 If the SE list is empty, then terminate the algorithm; otherwise, select the first node $i$ from the SE list and remove it from the list.

2.2 $\forall j \in \Gamma^{-1}(i)$ and $\forall \lambda \in S$,

2.2.1 Find $t_{\max} = \arg\max\{t + d_{ji}(t) = \tau - s_i(\tau), \forall t \in S\}$

2.2.2 Update $\delta_j(\tau) = \min\{\delta_j(\tau), d_{ji}(t_{\max}) + \delta_i(\tau)\}$

2.2.3 If at least one of the $M$ labels of node $j$ is improved (i.e. updated), insert node $j$ into the SE list.

Note that the TDSP-FAT operates in a label-correcting fashion, and hence the labels ($\boldsymbol{\delta}_i$, $\forall i \in N$) are upper bounds to the least time paths until the algorithm terminates.

# Chapter 3 Reformulation and Algorithm for the Dynamic User Equilibrium Problem

## 3.1 Introduction

The user equilibrium (UE) traffic assignment problem has been studied extensively in the past five decades since the pioneering work of Beckmann et al. (1956) introduced a mathematical program whose Kuhn-Tucker conditions coincide with Wardrop's first principle (Wardrop, 1952). An important extension of the problem is the UE dynamic traffic assignment (UEDTA) problem, which addresses the dynamic nature of traffic demands and flows in road networks as well as the path choice and/or departure time decisions of network users (Boyce et al., 2005). UEDTA models have evolved substantially in the last decade, and are seeing wider use in practice for predicting dynamic traffic flow patterns or evaluating traffic control and travel demand management measures. This chapter focuses on modeling and solving the UEDTA problem with given time-varying origin-destination (OD) demands, to find a time-varying path flow pattern that satisfies the time-dependent generalization of Wardrop's first principle: that travelers with the same OD and departure time experience the same and minimum travel time (or cost) along any used path, with no unused path offering a lower travel cost. Following the terminology given by Smith (1993), and increasingly adopted in the literature, the problem is referred to as the dynamic user equilibrium (DUE) problem in this study.

The goal of this study is to develop a simulation-based DUE model that is capable of realistically capturing traffic dynamics while adhering to a time-dependent generalization of Wardrop's first principle, as well as providing the basis for an algorithm that exhibits better performance (solution quality and computational effort) than commonly used averaging schemes (e.g. the method of successive averages, MSA) on practical networks. To this end, the DUE problem is reformulated, via a gap function, as a nonlinear minimization problem (NMP) whose global solution(s) coincides with solutions of the VI problem that satisfies the DUE principle. This reformulation is then solved by a column generation-based DUE algorithmic framework, which embeds (i) a simulation-based dynamic traffic (or network loading) model to capture traffic dynamics as well as to determine experienced path costs for any given path flow pattern and (ii) a descent direction method to solve the restricted NMP defined by a subset of feasible paths. The descent direction method has the following important features. First, it applies a scaling approach, in the same manner as the inverse of second order derivatives used in Newton-type methods, to determine appropriate step sizes. The scaling approach, which normalizes path cost differences between non-shortest paths and the shortest paths, also overcomes the deficiency of using absolute path cost differences in updating path assignments. Second, to be applicable in simulation-based DTA models as well as large-scale network problems, the proposed descent direction method does not require computing the gradient of the objective function. As a result, the underlying path (or link) cost functions need not be differentiable. Last, in order to mitigate the impact of possible oscillations and speed up convergence, this method is further integrated with a mixed step size scheme and an active constraint strategy in the column generation solution framework. Moreover, to circumvent the difficulty of storing the memory-intensive path set and routing policies for large-scale network applications, a vehicle-based implementation technique is proposed to use the vehicle path set as a proxy for keeping track of the path assignment results. This memory-efficient implementation technique can be seamlessly integrated with any mesoscopic/microscopic dynamic traffic model and is considered particularly appealing for large network deployments of DTA models.

The chapter is structured as follows. Section 3.2 gives the underlying assumptions and problem statement. The DUE conditions and a conventional VI formulation are then presented in section 3.3, followed by the description of an equivalent gap function-based nonlinear minimization reformulation in section 3.4. Section 3.5 first introduces a column generation-based DUE solution framework and then details the descent direction algorithm for solving restricted nonlinear optimization sub-problems and the vehicle-based implementation technique. Extensive computational results for several networks are reported in section 3.6 to demonstrate the solution quality and effectiveness of the proposed DUE algorithm.

## 3.2 Assumptions and Problem Statement

Consider a network G = ($N$, $A$), where $N$ is a finite set of nodes and $A$ is a finite set of directed links ($i$, $j$), $i \in N$ and $j \in N$. The time period of interest (planning horizon) is discretized into a set of small time intervals, $S = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$, where $t_0$ is the earliest possible departure time from any origin node, $\sigma$ is a small time interval during which no perceptible changes in traffic conditions and/or travel cost occur, and $M$ is a large number such that the intervals from $t_0$ to $t_0 + M\sigma$ covers the planning horizon $S$. Associated with each link ($i$, $j$) is the time-varying link travel time $c_{ij}(t)$ required to traverse link ($i$, $j$) when departing at time interval $t \in S$ from node $i$. Without loss of generality, $c_{ij}(t)$ is regarded as link travel time, though it can be generalized to include travel time, out-of-pocket cost and other travel impedances that may incur when traversing link ($i$, $j$) at time $t$. Travel time and cost are used interchangeably in this chapter. Other important notation and variables are summarized below.

| | |
|---|---|
| $O$ | subset of origin nodes; $O \subseteq N$ |
| $D$ | subset of destination nodes; $D \subseteq N$. |
| $T$ | set of departure time intervals. |
| $o$ | subscript for an origin node, $o \in O$. |
| $d$ | subscript for a destination node, $d \in D$. |
| $\tau$ | superscript for a departure time interval, $\tau \in T$. |
| $P(o, d, \tau)$ | set of all feasible paths for a given triplet ($o$, $d$, $\tau$). |
| $p$ | subscript for a path $p \in P(o,d,\tau)$. |
| $q_{od}^{\tau}$ | number of trips departing from node $o$ to node $d$ in time interval $\tau$. |
| $r_{odp}^{\tau}$ | number of trips departing from $o$ to $d$ in interval $\tau$ and assigned to path $p \in P(o,d,\tau)$. |
| $r$ | time-varying path flow vector, $r$ = { $r_{odp}^{\tau}$ , $\forall o \in O$, $d \in D$, $\tau \in T$, and $p \in P(o,d,\tau)$ }. |
| $c_{odp}^{\tau}(r)$ | path travel cost (or time) for the travelers departing from $o$ to $d$ in interval $\tau$ and assigned to path $p \in P(o,d,\tau)$; $c_{odp}^{\tau}(r) = \sum_{(i,j,t)\in p} c_{ij}(t)$, and is a function of the time-varying path flow vector $r$. |
| $c(r)$ | vector of path travel costs; $c(r)$ = { $c_{odp}^{\tau}(r)$ , $\forall o \in O$, $d \in D$, $\tau \in T$, and $p \in P(o,d,\tau)$ }. |

The time-varying OD demand pattern for the entire planning horizon (i.e. $q_{od}^{\tau}$, $\forall o$, $d$, and $\tau$) is assumed known *a priori*. It is also assumed that, for each $(o,d,\tau)$, all the trips departing at time $\tau$ from $o$ to $d$ have complete and accurate information about all the available paths connecting this OD pair and their characteristics. No en-route path-switching is allowed after departure from origins. The key behavioral assumption for the path choice decision is as follows: in a disutility-minimization framework, each trip-maker is rational and chooses a path that minimizes the travel cost. Specifically, for each trip-maker in $(o,d,\tau)$, a path $p^* \in P(o,d,\tau)$ will be selected if and only if $c^{\tau}_{odp^*}(r) = \min_{p \in P(o,d,\tau)} c^{\tau}_{odp}(r)$. With these precepts, the dynamic user equilibrium (DUE), a dynamic generalization of Wardrop's first principle, is defined as follows.

**Definition 3.1: DUE**

*For each OD pair and for each departure time interval, no traveler can reduce his/her experienced path cost by unilaterally changing path. That is, each traveler is assigned to a time-dependent least cost path. More costly routes are not used.*

Given the assumptions above, the problem is to solve the DUE traffic assignment problem, with a given time-varying OD demand, to obtain a time-varying path flow pattern satisfying the DUE conditions. Specifically, the goal is to determine a DUE path flow vector (routing policies) over a vehicular network for each OD pair and each departure time interval (i.e., $r^* \equiv \{ r^{\tau}_{odp}{}^*, \forall o, d, \tau, \text{ and } p \in P(o,d,\tau) \}$.

By the above DUE definition, all trips in a network are equilibrated in terms of actual experienced path costs, so it is necessary to determine the experienced path costs $c(r)$ for a given path flow vector $r$. To this end, a simulation-based dynamic traffic (or network loading) model is used to obtain the experienced path cost vector. It should be noted that the algorithm is independent of the specific dynamic traffic model selected; any (macroscopic, microscopic or mesoscopic) dynamic traffic model capable of capturing complex traffic flow dynamics, in particular the effect of physical queuing, as well as preventing violations of the first-in-first-out property, can be embedded into the proposed solution algorithm. When a particle-based dynamic traffic model is employed to determine experienced path costs, the path cost $c_{odp}^{\tau}(r)$ for a discrete time interval should be considered as the *average* path cost of the vehicles with the same $(o, d, \tau, p)$; that is $c_{odp}^{\tau}(r) = \frac{\sum_{v=1}^{r_{odp}^{\tau}} c_{odp}^{\tau,v}(r)}{r_{odp}^{\tau}(r)}$, where $c_{odp}^{\tau,v}(r)$ is the experienced path cost of vehicle *v*, because, to respect traffic propagation rules and junction exit capacity constraints, different vehicles embarking along path $p \in P(o, d, \tau)$ in departure interval $\tau$ will normally reach their destination *d* at different times and hence experience different trip times. This, in turn, means that the definition of a DUE in this study must involve the average experienced path cost. This coincides with the definition given in Beckmann et al. (1956): "Demand refers to trips and capacity refers to flows on roads. The connecting link is found in the distribution of trips over the network according to the principle that traffic follows shortest routes in terms of *average cost*".

## 3.3 DUE conditions and a VI formulation

The time-varying path flow vector $r^* \in \Omega$ is a solution to the DUE problem if the following DUE conditions are satisfied:

$$r_{odp}^{\tau} * \times [c_{odp}^{\tau}(r^*) - \pi_{od}^{\tau}(r^*)] = 0, \ \forall o, d, \tau, \text{ and } p \in P(o, d, \tau) \quad (3.1)$$

$$c_{odp}^{\tau}(r^*) - \pi_{od}^{\tau}(r^*) \geq 0, \ \forall o, d, \tau, \text{ and } p \in P(o, d, \tau) \quad (3.2)$$

where $\pi_{od}^{\tau}(r)$ is the least travel cost, with respect to the time-varying path flow vector *r*, from *o* to *d* in departure interval $\tau$. $\Omega \equiv \{r\}$ and $\Omega \subset R^m$ ($m=\sum_o \sum_d \sum_\tau |P(o,d,\tau)|$) is a set of feasible path flow vectors satisfying the following path flow conservation and non-negativity constraints:

$$\sum_{p \in P(o,d,\tau)} r_{odp}^{\tau} = q_{od}^{\tau}, \ \forall o, d, \text{ and } \tau \tag{3.3}$$

$$r_{odp}^{\tau} \geq 0, \ \forall \ o, d, \tau, \text{ and } p \in P(o, d, \tau) \tag{3.4}$$

Due to complex temporal and spatial interactions of time-varying link flows over a network, the Jacobian matrix of link travel time functions is generally not symmetric, and hence the nonlinear minimization program of the static UE problem proposed by Beckmann et al. (1956) is not applicable to the DUE traffic assignment problem. Extending his work (Smith 1979) on the static UE traffic assignment problem, Smith (1993) proposed that solving the DUE traffic assignment problem is equivalent to solving the following discrete-time and path-based VI problem: find a time-varying path flow vector $r^* \equiv \{ r_{odp}^{\tau} {}^*, \forall o, d, \tau, \text{ and } p \in P(o,d,\tau) \} \in \Omega$ such that

$$c(r^*) \bullet (r - r^*) \geq 0, \ \forall \ r \in \Omega \tag{3.5}$$

where the symbol $\bullet$ denotes the inner product between vectors of appropriate dimensions. This equivalence can be shown by adapting the proof given in Smith (1979). Although the theoretical guarantee of properties such as existence and uniqueness of solutions to the finite-dimensional VI problem (3.5) can be analytically derived, it generally requires the path cost function, i.e. $c(r)$, to be continuous and strictly monotone with respect to path flows on the finite and convex compact set $\Omega$ (e.g. Smith 1993 and Nagurney 1998). Those properties of path cost functions might not be satisfied in general road networks with complex traffic controls. The discussion of solution existence and uniqueness are beyond the scope of the current study.

## 3.4 An equivalent minimization problem

The VI formulation (3.5) enables the modeling, analysis, and computation of traffic network equilibria for general cases where the assumption of a symmetric Jacobian matrix of cost functions is no longer needed. Typically, the finite-dimensional VI problem (3.5) is solved as a series of approximate sub-problems, and many iterative algorithms, such as projection and linearization methods, can be used to progressively find the equilibrium solutions. This study reformulates the DUE problem as a nonlinear minimization program (NMP), by using a gap function, whose global minima coincide with solutions of the finite-dimensional VI problem (3.5) and hence satisfies the DUE conditions.

Several previous studies have applied similar reformulation techniques to the static UE or DUE problems. Hearn (1982) proposed a link-based primal gap function and used it to reformulate the UETA problem as an optimization (minimization) problem. Smith (1993) gave an equivalent minimization program of the DUE problem by minimizing the path-based user objective function

$$W(r) = \sum_{o\in O}\sum_{d\in D}\sum_{\tau\in T}\sum_{\forall p,q\in P(o,d,\tau)} r_{odp}^{\tau}[c_{odp}^{\tau}(r) - c_{odq}^{\tau}(r)]_{+}\ , \tag{3.6}$$

where $[x]_+$ denotes $\max\{0,\ x\}$. In general, $W(r)$ is not convex unless the path cost functions are affine and monotone (Patriksson, 1994). Moreover, $W(r)$ is not differentiable when $r_{odp}^{\tau} \geq 0$ and $[c_{odp}^{\tau}(r) - c_{odq}^{\tau}]_{+} = 0$. Smith (1993) further suggested that every term $[c_{odp}^{\tau}(r) - c_{odq}^{\tau}]_+$ in the above summation could be squared to remove the non-differentiability of *W*(*r*). Tong and Wong (2000) applied the user objective function *W*(*r*) in a DTA model. Lo and Chen (2000) proposed a smooth gap function adapted from *W*(*r*) to reformulate the static UE problem with fixed demand. By assuming that path cost functions are convex and monotonic with respect to path flows *r*, they showed that the resulting gap function is a convex function.

According to Patriksson (1999), a function *Gap*(*r*): $R^m$ -> $R^1$ is a gap function for a VI problem if (i) *Gap*(*r*) ≥ 0, ∀ $r \in \Omega$ and (ii) *Gap*(*r**) = 0 ⇔ *r** solves that VI problem. Specifically, this study defines the gap function as the following:

$$Gap(r) = \sum_{o \in O} \sum_{d \in D} \sum_{\tau \in T} \sum_{p \in P(o,d,\tau)} r_{odp}^{\tau} [c_{odp}^{\tau}(r) - \pi_{od}^{\tau}(r)]. \tag{3.7}$$

**Proposition 3.1**: *Gap*(*r*) is a gap function for the finite-dimensional VI problem (3.5).

**Proof**: By definition, $c_{odp}^{\tau}(r) - \pi_{od}^{\tau}(r) \geq 0$, ∀*o*, *d*, $\tau$ and $p \in P(o,d,\tau)$, so *Gap*(*r*) ≥ 0, ∀ $r \in \Omega$. Moreover, it can be observed that the time-varying path flow vector *r** ∈ Ω satisfies the DUE conditions (3.1) and (3.2) if and only if *Gap*(*r**) = 0. By the equivalence between finding DUE solutions and solving the VI problem (3.5), *r** is also the solution to (3.5) and hence *Gap*(*r*) is a gap function for the VI problem (3.5). This completes the proof.

The *Gap*(*r*) can be viewed as an adaptation of Smith's user objective function *W*(*r*) or a dynamic extension of Lo and Chen's smooth gap function. Note that, *Gap*(*r*) provides a measure of the violation of the DUE conditions in terms of the difference between the total actual experienced path travel cost and the total shortest path cost evaluated at any given time-varying path flow pattern $r \in \Omega$. The difference vanishes when the time-varying path flow vector $r^*$ satisfies the DUE conditions. Thus, solving the DUE problem can be viewed as a process of finding the path flow vector $r^* \in \Omega$ such that $Gap(r^*) = 0$. With the introduction of the gap function $Gap(r)$, the proposed nonlinear minimization problem (NMP) is presented as the following.

$$\underset{r\in\Omega}{Min} \sum_{o\in O}\sum_{d\in D}\sum_{\tau\in T}\sum_{p\in P(o,d,\tau)} r^{\tau}_{odp}[c^{\tau}_{odp}(r) - \pi^{\tau}_{od}(r)] \tag{3.8a}$$

$$\text{Subject to} \quad \sum_{p\in P(o,d,\tau)} r^{\tau}_{odp} = q^{\tau}_{od}, \ \forall o, d, \text{ and } \tau \tag{3.8b}$$

$$c^{\tau}_{odp}(r^*) - \pi^{\tau}_{od}(r^*) \geq 0, \ \forall o, d, \tau, \text{ and } p\in P(o, d, \tau) \tag{3.8c}$$

$$r^{\tau}_{odp} \geq 0, \ \forall\ o, d, \tau, \text{ and } p\in P(o, d, \tau) \tag{3.8d}$$

## 3.5 DUE solution algorithm

This study adopts a hybrid approach for solving the reformulated DUE optimization problem. Specifically, integrated in the proposed optimization algorithmic framework are a (feasible) descent direction method that minimizes the objective function (i.e. the gap function) and a simulation-based dynamic traffic model that generates, for a given path assignment $r$, the resulting traffic flow pattern from which the average link travel times, intersection turn delays and average experienced path costs $c(r)$ are extracted. Since the NMP reformulation uses path-related variables, a set of feasible paths on which the OD demands are to be equilibrated is required. It is generally very difficult, if not impossible, to enumerate the complete set of feasible paths of all OD pairs for a road network of practical size. Furthermore, only a (small) fraction of paths would carry positive flows in the DUE solution, in which path travel times should be equal to the least travel time of each corresponding triplet $(o,d,\tau)$, and only the constraints in (3.8c) that correspond to these used paths are binding. To avoid explicit enumeration of all possible paths, this study uses a column generation-based approach to generate a representative subset of paths that have competitive travel times.

### 3.5.1 Column generation-based DUE solution framework

The column generation-based approach augments, in the outer loop, the subset of the feasible paths and solves, in the inner loop, the restricted NMP (RNMP) with the current subset of paths. In each outer iteration *k*, the efficient time-dependent least cost path algorithm proposed by Ziliaskopoulos and Mahmassani (1993) is applied to solve the time-dependent shortest path problem, and the new paths (if any) are generated and added to the current path set $\overline{P}(o,d,\tau) \subseteq P(o,d,\tau)$ for all triplets $(o,d,\tau)$. The algorithm terminates if there is no new path found or a preset convergence criterion is satisfied; otherwise, the RNMP is solved by a descent direction method presented in the next subsection to equilibrate the current path set before returning to the path generation step. The descent direction method proceeds iteratively and forms the inner loop (with iteration counter *l*) in the column generation-based solution framework, in a manner similar to the restricted path set equilibration scheme suggested by Larsson and Patriksson (1992). It is worth noting that, as also suggested by early studies on the diagonalization algorithm for asymmetric traffic assignment problems (see e.g. Mahmassani and Mouskos, 1988), the RNMP does not have to be solved optimally in each iteration *k*, in order to improve the overall computational efficiency and achieve satisfactory convergence. The column generation-based solution algorithm is detailed in the following.

**The Column Generation-Based DUE Solution Algorithm**

**Initialization**

1. Set the iteration counter of outer loop $k = 0$. Perform a dynamic network loading with initial path assignment $r^k$ and obtain time-dependent link travel times and path travel times $c(r^k)$ from the simulator.

**Outer Loop – Path Generation**

2. Time-Dependent Shortest Path Tree Calculation: solve the time-dependent shortest path problem to find the shortest path $p^k(o,d,\tau)$ and the corresponding travel time $\pi_{od}^{\tau,k}$ for each $(o,d,\tau)$. If $p^k(o,d,\tau)$ is not in $\overline{P}(o,d,\tau)$, then add $p^k(o,d,\tau)$ to $\overline{P}(o,d,\tau)$.
3. Calculate the value of $Gap(r^k)$ using $r^k$, $\pi^k$ and $c(r^k)$.
4. Convergence Checking: if (a) there is no new path found for all $(o,d,\tau)$ and $Gap(r^k) \leq \varepsilon_{\text{out}}$ (a preset convergent threshold) or (b) $k = K_{max}$ (maximum number of outer iterations) then stop and output the solution $r^k$; otherwise start the inner loop with $r^k$, $\pi^k$ and $c(r^k)$.

**Inner Loop – Solving Restricted NMP**

5. Set iteration counter of inner loop $l = 0$ and $Gap(r^l) = 0$, and read $r^l$, $\pi^l$ and $c(r^l)$ from step 4.
6. Update Path Assignment: determine path assignment $r^{l+1}$ by using the descent direction method. Set $l = l + 1$.
7. Dynamic Network Loading (DNL): perform a DNL with new path assignment $r^l$ and obtain link travel times and path travel times $c(r^l)$ from the simulator.
8. Find, in $\overline{P}(o,d,\tau)$, the shortest path $p^l(o,d,\tau)$ and the corresponding travel time $\pi_{od}^{\tau,l}$ for each $(o,d,\tau)$.
9. Calculate the value of $Gap(r^l)$ using $r^l$, $\pi^l$ and $c(r^l)$.
10. Convergence Checking: if $|Gap(r^l)-Gap(r^{l-1})| \leq \varepsilon_{\text{in}}$ (a preset convergent threshold) or $l = L_{max}$ (maximum number of inner iterations) then return to the outer loop (step 2) with current sets of link travel times, path travel times $c(r^l)$ and $r^l$, and set $k = k+1$; otherwise go back to step 6.

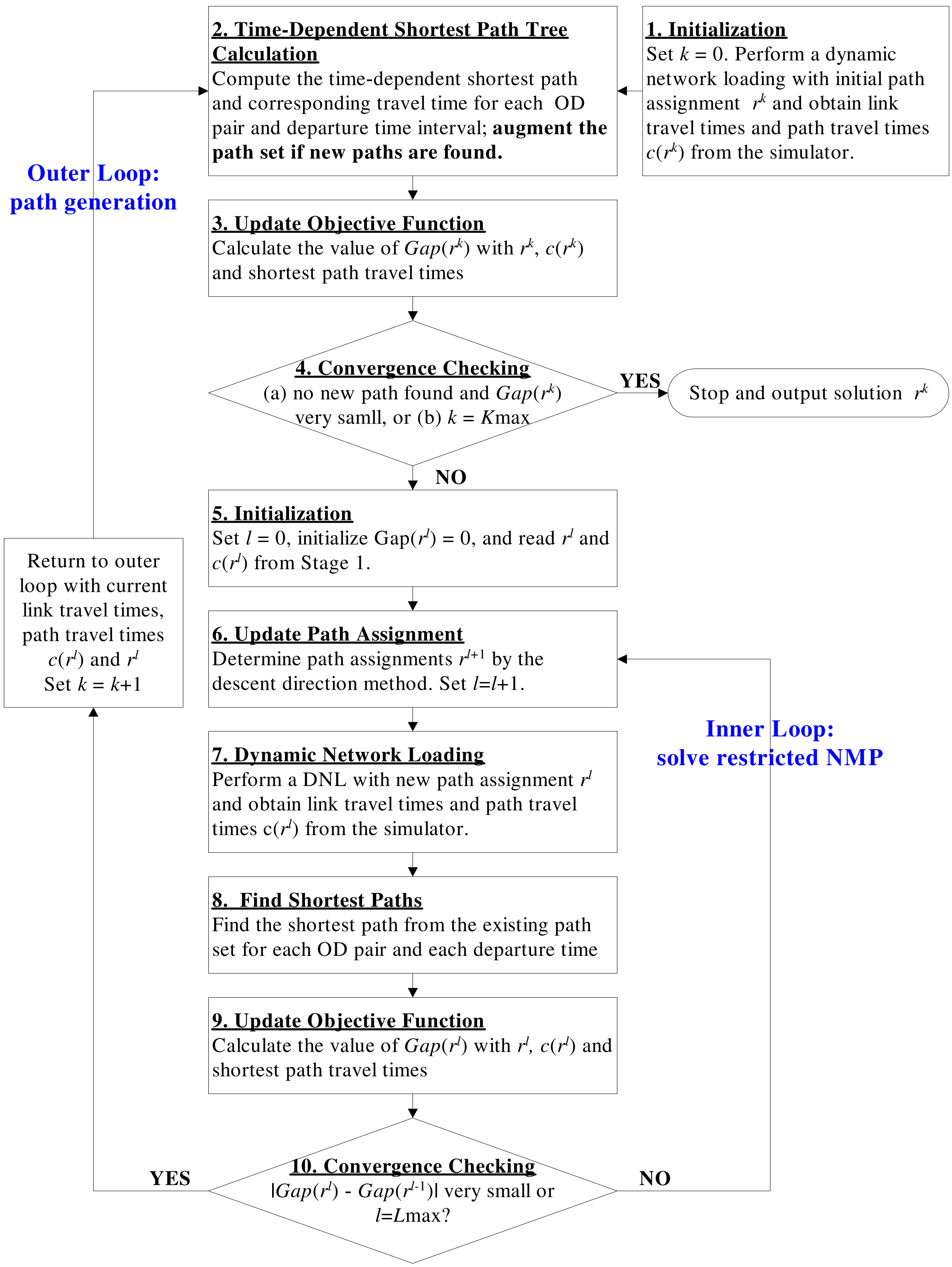



Figure 3.1 The column generation-based DUE algorithm

### 3.5.2 Solving the restricted NMP

Several conventional gradient-based algorithms (e.g. gradient projection and reduced gradient methods) for constrained nonlinear programming problems can be applied to solve the RNMP, provided that the path cost function is differentiable. With a feasible solution $r \in \Omega$, these algorithms adopt the search direction along the feasible descent direction of $Gap(r)$ at $r$, determined directly or indirectly by the gradient of $Gap(r)$, which can be written in the following vector form:

$$\nabla Gap(r) = \partial Gap(r)/\partial r = c(r) - \pi(r) + r \bullet (\partial c(r)/\partial r - \partial \pi(r)/\partial r). \quad (3.9)$$

However, because of the temporal dimension, computing partial derivatives (i.e. $\partial c(r)/\partial r$ and $\partial \pi(r)/\partial r$) is computationally intensive (or even intractable for large networks). Furthermore, when experienced path costs are obtained through a simulation-based dynamic network loading model, analytical calculations of partial derivatives are not available. Though it is possible to compute them using numerical methods, the stability and accuracy of numerically calculated derivatives are not guaranteed. Therefore, to enable the deployment of large-scale (simulation-based) DTA models, this study proposes a descent direction method to circumvent the need to calculate partial derivatives.

The proposed descent direction method is a projection type algorithm that decomposes the RNMP into many $(o, d, \tau)$ sub-problems and solves each of them by adjusting time-varying OD flows between all non-shortest paths and the shortest path(s). Given a feasible solution $r^l \in \Omega$ in an inner loop iteration $l$, the method features the following form:

$$r^{l+1} = P_\Omega[r^l - \rho^l \times Dir^l] = P_\Omega[r^l - \rho^l \times \frac{r^l \times (c(r^l) - \pi(r^l))}{c(r^l)}], \quad (3.10)$$

where $\rho^l \in (0,1)$ is the step size in iteration $l$, $-Dir^l$ is the descent direction. $P_\Omega[u]$ denotes the unique projection of vector $u \in R^{m+}$ onto $\Omega$ and is defined as the unique solution of the problem: $\min_{v\in\Omega} \| u - v \|$. Based on Eq.(3.10), the new path assignment $r^{l+1}$ is obtained by updating the current path assignment $r^l$ along the descent direction ($-Dir^l$) with a move size $\rho^l$. This path assignment updating scheme implies a natural path flow adjustment mechanism: flows on the non-cheapest paths are moved to the cheapest path and the volume moved out from a non-cheapest path $p$ is proportional to $(c^\tau_{odp}(r^l) - \pi^\tau_{od}(r^l)) / c^\tau_{odp}(r^l)$ – the relative (or scaled) difference in path cost between non-shortest paths and the shortest path, which is intuitively based on the fact that travelers farther from the equilibrium and on paths with larger flow rates are more strongly inclined to change path than those on paths with smaller flow rates and with travel cost closer to the minimal cost.

It could be noted that this scheme appears to be similar to the common path-swapping heuristics applied by several researchers in the analytical DTA arena, such as Smith (1995), Cybis (1995), Huang and Lam (2002), and Szeto and Lo (2005). However, this path assignment updating scheme is intended to deal with a deficiency similar to that of logit-based path choice models where the path choice probabilities are determined solely on the basis of absolute path-cost differences (see e.g. Sheffi, 1985). Consider the following simple example (Figure 3.2) with two OD pairs: (1, 2) and (1, 3). The travel time of each link (or path) in iteration $l$ is labeled next to that link in the figure. According to Eq.(3.10), the number of vehicles shifted from the non-shortest path to the shortest path is the same (i.e. $5\rho^l$ ) for both OD pairs. However, it could be observed that OD pair (1,3) is closer to the equilibrium than OD pair (1,2), and hence the vehicles on link 3 are less inclined to switch to the shortest path than the vehicles on link 1. Moving too many vehicles from link 3 to link 1 might change the shortest path to the non-shortest path and vice versa. This indicates that the search direction determined by the absolute path-cost difference might not be efficient in determining the path flows shifted for different OD pairs with distinct path cost magnitudes.

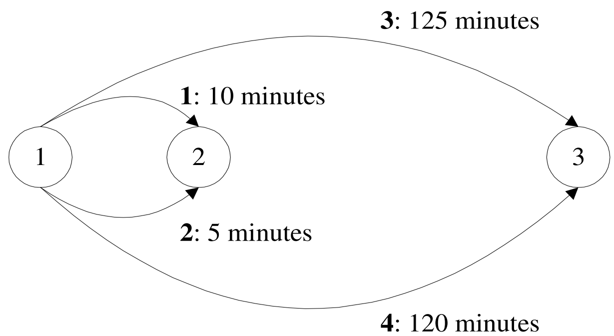



Figure 3.2 A tree-node network

To remedy the potential drawbacks of using absolute path-cost differences in the path assignment updating scheme, this study applies a scaling factor equal to the reciprocal of path cost to normalize the path-cost difference. Define $\overline{P}_{\pi}(o,d,\tau)$ = $\{ p : p \in \overline{P}(o,d,\tau)$ and $c_{odp}^{\tau}(r^l) = \pi_{od}^{\tau}(r^l) \}$. Specifically, for each $(o,d,\tau,l)$, the descent direction method leads to the following path assignment updating scheme in iteration $l$:

$$r_{odp}^{\tau,l+1} = \max\{0,\ r_{odp}^{\tau,l} \times [1 - \rho^l \times \frac{c_{odp}^{\tau}(r^l) - \pi_{od}^{\tau}(r^l)}{c_{odp}^{\tau}(r^l)}]\} \qquad \forall p \in \overline{P}(o,d,\tau) \setminus \overline{P}_{\pi}(o,d,\tau) \tag{3.11}$$

$$r_{odp}^{\tau,l+1} = r_{odp}^{\tau,l} + \frac{\psi_{od}^{\tau,l}}{|\overline{P}_{\pi}(o,d,\tau)|},\ \forall p \in \overline{P}_{\pi}(o,d,\tau), \tag{3.12}$$

where $\psi_{od}^{\tau,l} = \sum_{p \in \overline{P}(o,d,\tau) \setminus \overline{P}_{\pi}(o,d,\tau)} r_{odp}^{\tau,l} \times \frac{c_{odp}^{\tau}(r^l) - \pi_{od}^{\tau}(r^l)}{c_{odp}^{\tau}(r^l)}$ and $|\overline{P}_{\pi}(o,d,\tau)|$ denotes the number of paths in the set $\overline{P}_{\pi}(o,d,\tau)$.

### 3.5.3 Step size selection and active constraint set strategy

One of the challenges of solving the NMP reformulation is the presence of constraints (3.8c), given that both path flows $r$ and least travel times $\pi = \{ \pi_{od}^{\tau}(r), \forall o, d, \tau\}$ are decision variables and the latter also depend on path flows $r$. Specifically, if a large step size is applied in the path flow updating step in an inner loop iteration, as a consequence of path flow shifting, a current non-shortest path might become the shortest path and the current shortest path could turn out to be a non-shortest one; hence some constraints in (3.8c) would be violated. Furthermore, in the first inner loop iteration (i.e. $l = 0$), after an outer loop iteration of path generation, since the path-cost differences between those new (shortest) paths and the corresponding non-shortest paths might be large, a vast amount of flow would be shifted to those new paths and the resulting updated path flows could also violate some constraints in (3.8c), if the step size is not suitably selected. Therefore, to maintain the feasibility of the updated path flows, one has to carefully choose the step sizes and explicitly keep track of the exact change of least travel times $\pi$ and the set of active constraints.

Standard nonlinear programming theory (e.g. Bertsekas, 1995) suggests that the use of the Armijo step size rule in a line search scheme can help to identify the active constraints in a finite number of iterations. Besides, one can also use the gradient information to estimate the possible changes in the least travel times and the active constraint set. For example, to solve a reformulation of the static fixed demand traffic assignment problem, Lo and Chen (2000) determined search directions by using the gradient of a gap function in which partial derivatives were taken with respect to both path flows and least travel times. Nevertheless, in a simulation-based DTA model, performing a line search scheme and calculating the gradient of the gap function are computationally intensive (and prohibitive in real networks). To deal with the possible oscillations (i.e., least travel times and active constraints change from iteration to iteration), this study adopts a mixed scheme of step sizes, described in the following.

$$\rho^l = 1/k, \text{ if } l = 0; \ \rho^l = 1, \text{ otherwise.} \tag{3.13}$$

Recall that $k$ is the iteration counter of the outer loop. Essentially, this step size rule aims to mitigate the impact of introducing new paths to the current path set on the objective function (when $l = 0$) and uses the scaling factors $1/c(r^l)$ to take care of the selection of step sizes in the subsequent inner loop iterations (when $l > 0$). The diminishing step size $\rho^l = 1/k$ is prescribed by the method of successive average (MSA). The technique of using scaling factors to bypass the need to determine suitable step sizes was also suggested by Bertsekas and Gafni (1983) and Jayakrishnan et al. (1994), where the second derivative information was used for an automatic scaling and $\rho = 1$. The results from some pilot experiments conducted in this study have shown that this mixed step size scheme can efficiently reduce the possible oscillations.

With the mixed step size scheme (3.13), this study assumes the active constraint set, which is identified at the beginning of each inner iteration, stays fixed during an inner iteration. Furthermore, if a path flow variable violates the non-negativity constraint, that variable is set to zero. This active constraint set strategy has also been adopted by several studies, such as Smith and Winsten (1995) and Huang and Lam (2002).

### 3.5.4 Proof of the descent direction method

**Definition 2: link marginals and path marginals**

Denote by $\varphi_a^t$ the time-dependent link marginal: the travel time contribution of an additional unit of vehicular flow on link *a* in time interval *t* to the link travel time $c_a^t$. By assuming that $c_a^t$ is a monotonic (increasing) function of $x_a^t$ (the number of vehicles on link *a* in time interval *t*): $[(x_a^t + \Delta x_a^t) - x_a^t] \times [c_a^t(x_a^t + \Delta x_a^t) - c_a^t(x_a^t)] \geq 0$ (e.g. Nagurney, 1998), with $\Delta x_a^t > 0$, the following can be obtained:

$$\varphi_a^t = \lim_{\Delta x_a^t \to 0} \frac{c_a^t(x_a^t + \Delta x_a^t) - c_a^t(x_a^t)}{\Delta x_a^t} \geq 0. \tag{3.14}$$

Note that this study considers $\varphi_a^t$ as a local link marginal. Peeta (1994) gave a comprehensive discussion on global link marginals with temporal and spatial interactions. Assuming that path marginals are additive of link marginals, the path marginal of path $p \in \overline{P}(o,d,\tau)$ is:

$$\eta_{odp}^{\tau} = \sum_{a \in A(p)} \varphi_a^t \times \delta_{odpa}^{\tau,t}, \tag{3.15}$$

where *A*(*p*) is the set of links on path *p*, *t* is the first time interval in which link *a* on path *p* is reached by a vehicle assigned to that path at time $\tau$, and $\delta_{odpa}^{\tau,t}$ is the time-dependent link-path incidence indicator; $\delta_{odpa}^{\tau,t}$ =1 if vehicles going from *o* to *d* assigned to path *p* at time $\tau$ pass link $a$ in time interval $t$, and 0 otherwise. Note that this assumption may not hold strictly in reality, and as such, is used here in an approximate sense.

Let $p^* \in \overline{P}_\pi(o,d,\tau)$ be the referenced shortest path for a triplet $(o,d,\tau)$. Then constraints (3.8b) can be re-written as follows:

$$r^{\tau}_{odp^*} = q^{\tau}_{od} - \sum_{p \in \overline{P}(o,d,\tau) \setminus p^*} r^{\tau}_{odp}, \ \forall o, d, \text{ and } \tau. \tag{3.16}$$

By substituting Eq.(3.16) into the objective function (3.8a), the RNMP, in iteration $l$, becomes the following unconstrained minimization problem:

$$\underset{r \in \Omega}{Min} \ \ Gap(r) = \sum_{o \in O} \sum_{d \in D} \sum_{\tau \in T} \sum_{p \in \overline{P}(o,d,\tau) \setminus p^*} r^{\tau,l}_{odp} [c^{\tau}_{odp}(r^l) - \pi^{\tau}_{od}(r^l)] \tag{3.17}$$

Note that the constraints (3.8c) and (3.8d) are satisfied in RNMP because of the aforementioned active constraint set strategy and the projection of the updated solution onto the feasible set Ω, respectively. With this transformation and according to Eqs. (3.9) and (3.15), the first-order partial derivative of *Gap*($r$) with respect to a particular $r^{\tau}_{odp}$ is obtained as:

$$\frac{\partial Gap(r)}{\partial r^{\tau}_{odp}} = c^{\tau}_{odp}(r) - \pi^{\tau}_{od}(r) + r^{\tau}_{odp} \times \sum\nolimits_{a \in \overline{A(p) \cap A(p^*)}} \varphi^{t}_{a} \tag{3.18}$$

where $\overline{A(p) \cap A(p^*)}$ is the set of links that are on either the non-shortest path $p$ or the referenced shortest path $p^*$. The following proposition and its proof can now be given.

**Proposition 3.2**: The search direction $[r \dfrac{c(r) - \pi(r)}{c(r)}]$ is a descent direction of *Gap*($r$) at $r$.

**Proof**: To prove the vector $Dir = [r\frac{c(r)-\pi(r)}{c(r)}]$ is a descent direction of *Gap*(*r*) at *r*, it is necessary to show that the inner product $\nabla Gap(r) \bullet (-1 \times Dir) < 0$ (see e.g. Theorem 4.1.2 in Bazara'a et al. 1993). Component-wise, this is equivalent to showing that

$$\sum_{p\in \bar{P}(o,d,\tau)\setminus p^*}((-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau}) < 0,\ \forall o,\ d, \text{ and } \tau, \tag{3.19}$$

where $Dir_{odp}^{\tau} = [r_{odp}^{\tau}\times\frac{(c_{odp}^{\tau}(r)-\pi_{od}^{\tau}(r))}{c_{odp}^{\tau}(r)}]$ and $\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}$ is defined as Eq.(18).

Consider that, for a triplet $(o,d,\tau)$ and for each path *p* ($r_{odp}^{\tau} > 0$) in the path set $\bar{P}(o,d,\tau)\setminus p^*$, the cost of path *p* could be either equal to or greater than the least cost . In the first case, *p* is one of the shortest (more precisely, least cost) paths (i.e. $p\in \bar{P}_{\pi}(o,d,\tau)\setminus p^*$), then $Dir_{odp}^{\tau} = 0$ and accordingly $(-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau} = 0$. In the latter case, *p* is a non-shortest path, (i.e. $p\in \bar{P}(o,d,\tau)\setminus \bar{P}_{\pi}(o,d,\tau)$), then $Dir_{odp}^{\tau} > 0$. According to Eq.(3.14), link marginals are non-negative and $c_{odp}^{\tau}(r)-\pi_{od}^{\tau}(r)$ is positive for any non-shortest path *p*, so $\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}} > 0$ and $(-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau} < 0$ . Mathematically, for each $(o,d,\tau)$

$$\begin{aligned}
&\sum_{p\in \bar{P}(o,d,\tau)\setminus p^*}[(-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau}] \\
&= \sum_{p\in \bar{P}_{\pi}(o,d,\tau)\setminus p^*}[(-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau}] + \sum_{p\in \bar{P}(o,d,\tau)\setminus \bar{P}_{\pi}(o,d,\tau)}[(-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau}] \\
&= 0 + \sum_{p\in \bar{P}(o,d,\tau)\setminus \bar{P}_{\pi}(o,d,\tau)}[(-1)\times\frac{\partial Gap(r)}{\partial r_{odp}^{\tau}}\times Dir_{odp}^{\tau}] < 0
\end{aligned} \tag{3.20}$$

Thus the search direction *Dir* is a descent direction of *Gap*($r$) at $r$. This completes the proof.

### 3.5.5 Vehicle-based implementation technique

The above DUE algorithm is featured as the path-based approach, necessitating the explicit storage of the path set and path assignment results for each $(o,d,\tau)$ from iteration to iteration. Although it is straightforward to record all the paths and the corresponding path choice probabilities for each $(o,d,\tau)$ by using multi-dimensional arrays, computer memory requirements grow dramatically when the number of OD pairs is large, or many iterations are required to achieve convergence.

In a particle-based and simulation-based DTA system, individual vehicles are tracked and moved along their journeys from origins to destinations. Thus, vehicles have to carry their paths from iteration to iteration, and the vehicle path set implicitly reflects and stores the path set and path assignments results. This is particularly advantageous for large-scale DTA applications, as the total number of feasible paths generated by the iterative solution algorithm, after a certain number of iterations, could be significantly greater than the total number of vehicles, which is determined a priori by the OD demand table. For example, in the Portland transportation planning network (Nagel et al., 2000), there are about 1,260 traffic analysis zones (TAZ) and 1.5 million OD pairs, and the corresponding total trip-makers are 1.5 million in all time periods. Obviously, every OD pair requires more than one time-dependent shortest path for reaching the DUE conditions. Thus, storing the vehicle path set is more memory-efficient than storing the complete path set and routing policies for large-scale networks.

With this vehicle-based implementation technique, the path assignment updating scheme presented in Eq.(3.11) and Eq.(3.12) can be interpreted as the following. In iteration *l*, $\forall p \in \overline{P}(o,d,\tau) \setminus \overline{P}_{\pi}(o,d,\tau)$, the portion of the path flow $r_{odp}^{\tau}$ moved to the (referenced) least generalized cost path is $(c_{odp}^{\tau}(r^l) - \pi_{od}^{\tau}(r^l)) / c_{odp}^{\tau}(r^l)$ ; while the remaining vehicles would keep their current paths. Essentially, this implementation technique uses the vehicle path set as a proxy for the exact path set and assignment results, and the path set and routing policies of interest can be approximately recovered from the realized vehicle paths in the last iteration's simulation results.

## 3.6 Numerical experiments

Two sets of numerical experiments are conducted to examine the column generation-based DUE algorithm and the embedded descent direction method (DDM). The proposed algorithm is hence denoted as CGDDM, hereafter. The first set of experiments aims to validate the solutions found by the algorithm and is conducted on two small networks in which the computational effort for path generation is light. The second set of experiments evaluates the performance of the algorithm on several real road networks with different sizes and configurations (corridor-based and non-corridor-based).

Given a set of paths ( $P(o,d,\tau), \forall o,d,\tau$ ) and the corresponding path assignment $r \in \Omega$, the simulation-based dynamic traffic model – DYNASMART (Jayakrishnan et al., 1994a) is used to evaluate a path assignment *r* and determine experienced path costs *c*(*r*). DYNASMART adopts a mesoscopic approach to capture the dynamics of vehicular traffic flow in the simulation, where vehicles are tracked individually (or *microscopically*) and moved according to prevailing local speeds, consistently with *macroscopic* traffic flow relations between speed and concentration on links. Therefore, the experienced path cost $c(r)$ can be obtained from either averaging vehicle experienced trip costs or adding up the aggregated costs of constituent links. The objective function $Gap(r)$ is called *vehicle experienced cost* (VEC) *gap* when vehicle experienced trip costs are used, and is named *aggregated link cost* (ALC) *gap* if path costs are obtained by summing up the aggregated costs of constituent links. The $Gap(r)$ can be either the VEC gap or the ALC gap when the embedded dynamic traffic model is mesoscopic, but the VEC (ALC) gap is usually not available when a macroscopic-based (microscopic) dynamic traffic model is used. To demonstrate that the proposed CGDDM is not restricted by the choice of the dynamic traffic model, for some of the experiments, both the VEC and ALC gaps are applied as the $Gap(r)$ and the results are reported separately.

Another measure of effectiveness (MOE) is collected in the conducted experiments, in addition to the objective value $Gap(r)$. It is the average gap over all the vehicles in the network for a given path flow pattern $r$.

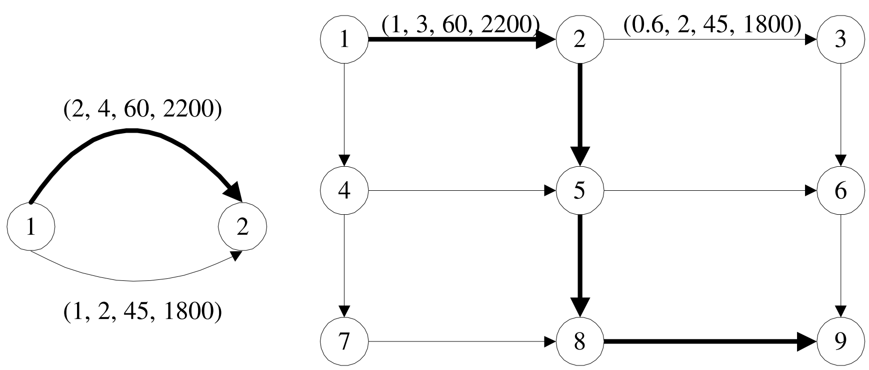
$$AGap(r) = \frac{\sum_{o\in O}\sum_{d\in D}\sum_{\tau\in T}\sum_{p\in P(o,d,\tau)} r_{odp}^{\tau}[c_{odp}^{\tau}(r) - \pi_{od}^{\tau}(r)]}{\sum_{o\in O}\sum_{d\in D}\sum_{\tau\in T}\sum_{p\in P(o,d,\tau)} r_{odp}^{\tau}} \tag{3.21}$$

$AGap(r)$ is used as a surrogate of the gap function $Gap(r)$ in this study. This MOE is independent of problem size and thus useful for examining the convergence pattern and solution quality of a DUE algorithm on different networks. The minimum of the $AGap(r)$ is zero. Essentially, the smaller the average gap, the closer the solution is to a DUE.

The proposed CGDDM algorithm is implemented using the vehicle-based approach, which uses the vehicle path set as a proxy for keeping track of the path assignment results. This memory-efficient implementation technique can be seamlessly integrated with any mesoscopic/microscopic dynamic traffic model and is considered particularly appealing for large network deployments of DTA models. The algorithm is coded and compiled by using the Compaq Visual FORTRAN 6.6 and evaluated on the Windows XP platform and a machine with an Intel Pentium IV 2.8 GHz CPU and 2GB RAM. In all experiments conducted, the following parameter settings are applied. The resolution (aggregation interval) of the time-dependent shortest path tree calculation is set to 0.1 minute, which is the same as the time step for the simulation. The OD demand assignment interval (or departure time interval) is set to 1 minute, although in some experiments it is considered as an experimental parameter and varied between 0.1 and 5 minutes. In each experiment, a 2-hour time-varying OD demand table is loaded. The (simulation) planning horizon is 150 minutes while the statistics are collected only from 10 to 100 minutes (i.e. observation period) to take into account the time for simulation warm-up and network clearance. A strict convergence criterion is used in the inner loop (solving RNMPs) of the column generation-based DUE algorithm; that is $|Gap(r^l) - Gap(r^{l-1})|/Gap(r^l) \leq 0.001$. The initial solutions of the experiments are obtained by loading time-varying OD demands to the shortest paths based on prevailing travel times.

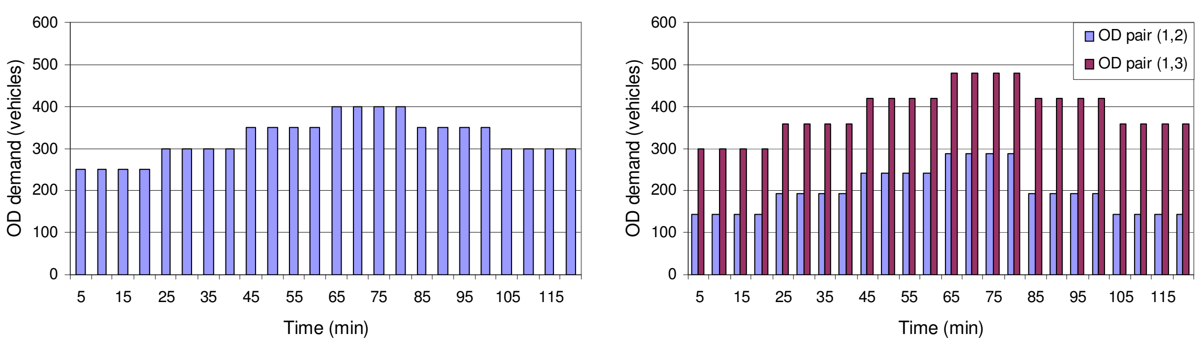
### 3.6.1 Experiments on two small test networks

The first set of experiments aims to examine the convergence pattern and to validate the solutions (i.e. whether they satisfy the DUE conditions or not). The first experiment is conducted on a two-node network (Figure 3.3(a)). There are two links (paths) connecting the only OD pair (1, 2), and each link is divided into many segments, each of which has the length of 0.2 miles. Associated with each link are the following attributes: length (miles), number of lanes, free flow speed (miles per hour), and capacity (vehicles per hour per lane). The 2-hour OD demands (7,800 vehicles) are shown in Figure 3.4(a). There are 6,100 vehicles loaded in the observation period. Note that the path generation loop of the algorithm is not activated here, as the focus of this experiment is on the effectiveness of the descent direction method (i.e. without the column generation scheme) and the only two paths used are already in the initial solution.




(a) Two-node network (b) Nine-node network

Figure 3.3 Two small test networks




(a) Two-node network (b) Nine-node network

Figure 3.4 The time-dependent OD demands of the two small networks

Depicted in Figure 3.5 are the result of performing 10 iterations of the DDM is as well as the method of successive averages or MSA (i.e. using the reciprocal of the iteration counter as the step size in updating path assignments). In the vehicle-based implementation, the MSA updates the path assignment $r^l$ in the following way (Lu et al. 2006):

$$\text{MSA: } r^{l+1} = r^l + 1/l \times (y^l - r^l) = (1-1/l) \times r^l + 1/l \times y^l, \tag{3.22}$$

where $y^l$ is the auxiliary (all-or-nothing) assignment obtained in iteration $l$. Although the use of predetermined move size from the MSA may lack search efficiency, it reduces the computational efforts required for analytically optimizing the move size. It is found that, compared with the MSA, the DDM performs relatively well in reducing the average gap and the objective value of the *vehicle experienced cost gap*. After 10 iterations, the average gap is decreased from 0.45 minutes (initial solution) to 0.01 minutes by the DDM and to 0.03 minutes by the MSA. The final objective values obtained by the DDM and the MSA are, respectively, 61.0 and 190.5 (initial value is 2762.1). The computation times of the DDM and the MSA are 10.6 and 9.5 seconds, respectively. Both the MSA and the DDM are able to reduce the average gap (and the objective value) to a satisfactory level – less than 0.05 minutes. However, with a slight increase of the computation time, the DDM gives a better DUE solution than that of the MSA. Moreover, the convergence pattern of the DDM is nearly non-increasing, while that of the MSA fluctuates significantly in the first few iterations. To validate whether or not the solution found by the DDM satisfies the DUE conditions, the (absolute) experienced travel time differences between the two paths for each 1-minute departure interval are plotted in Figure 3.6. Those travel time differences are quite significant in the initial solution, but are greatly reduced in the final solutions obtained by the DDM and the MSA. In particular, the path travel time differences in the solution of the DDM are all less than 0.1 minutes. Those (very) small time-varying path travel time differences indicate that vehicles departing at any time interval and traversing on the two paths have experienced almost the same time, validating that DDM is able to find close-to-DUE solutions.

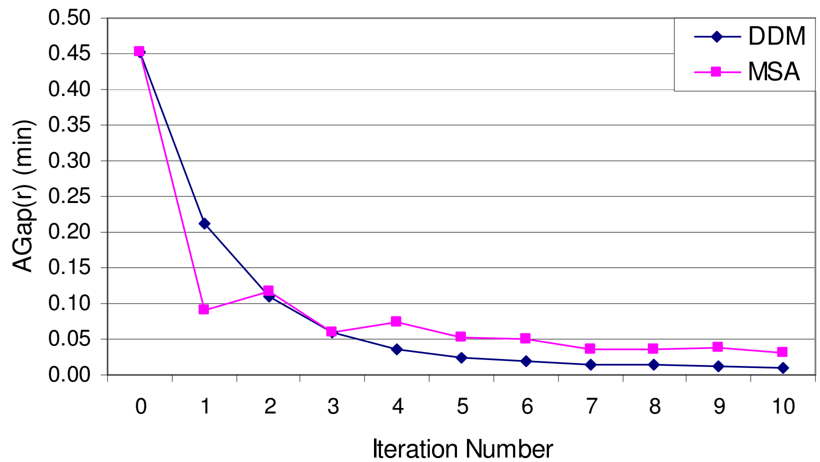



Figure 3.5 Convergence patterns of DDM and MSA on the two-node network

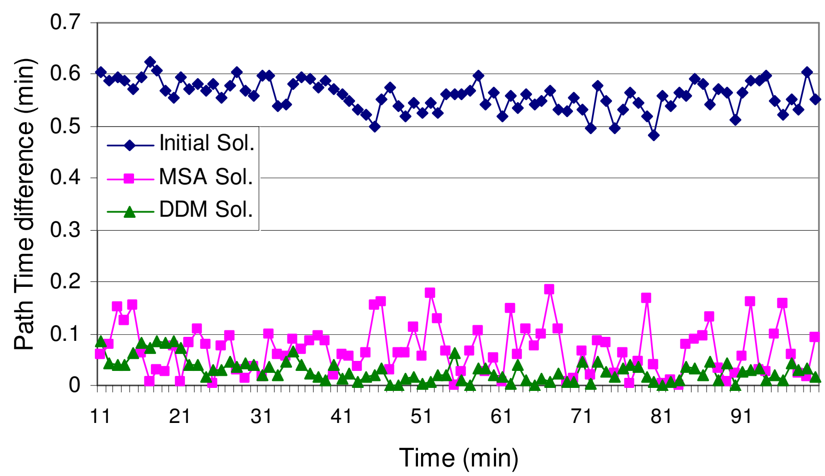



Figure 3.6 Experienced path time differences of the solutions on the two-node network

The next experiment is conducted on a nine-node network (Figure 3.3(b)) with two OD pairs (1,5) and (1,9). Links (1->2), (2->5), (5->8) and (8->9), have the same attributes, while the other links have the same attributes. The 2-hour time-varying OD demands (14,160 vehicles) are shown in Figure 3.4(b). There are 11,236 vehicles loaded in the observation period. Note that the path generation (outer) loop of the CGDDM is activated in this experiment, though enumerating paths in this network is not difficult. Three different step size rules are implemented and tested on the nine-node network to explore their impact on the performance of the CGDDM,

MSA step: $\rho^l = 1/(l+1), \forall\, l$.
Unit step: $\rho^l = 1, \forall\, l$.
Mixed step: $\rho^l = 1/k$, if $l = 0$; $\rho^l = 1$, otherwise.

In addition, the MSA presented in Eq.(3.22) is also implemented in the same column generation-based algorithmic framework (named CGMSA) to compare with the proposed CGDDM. The maximum numbers of iterations of the outer loop and the inner loop (i.e. $K_{max}$ and $L_{max}$) are set to 20 and 10, respectively.

First, the path costs (i.e. $c(r)$) are obtained by averaging vehicle experienced costs and hence the $Gap(r)$ is the VEC gap. The convergence patterns of the CGDDM with these three different step size rules and the CGMSA on the nine-node network are presented in Table 3.1. Both the CGDDM and the CGMSA reach convergence (i.e. there are no new paths found) in 15 iterations. Among the three different step size rules, the mixed step size gives a better (lower) objective value than those of the other two step sizes although it takes a few more iterations to reach convergence (identify all competitive paths). Moreover, although the average computational effort of performing one iteration of the CGMSA (21.5 seconds) is less than that of the CGDDM with the mix step size (23.7 seconds), the former takes more iterations to converge and more computation time overall (301 seconds) than the latter does (213 seconds). Note that the CGMSA requires less computation time for finishing one iteration than the CGDDM because the former does not calculate the search directions $[c(r)-\pi(r)]/c(r)$ and only uses predetermined step sizes.

The same experiment is conducted again with the path costs obtained by summing up the costs along constituent links and hence the *Gap*(*r*) is the ALC gap. The convergence patterns of the CGDDM with these three different step size rules and the CGMSA are presented in Table 3.2. Both the CGDDM and the CGMSA reach convergence (i.e. there are no new paths found) in 13 iterations. The results from Table 3.1 and Table 3.2 demonstrate the capability of the CGDDM to satisfactorily solve the DUE problem (with either the VEC gap or the ALC gap as the objective function) on a network with multiple OD pairs.

Table 3.1 Convergence patterns of CGDDM and CGMSA (experienced cost gap)

($K_{max}$ =20, $L_{max}$ =10, assign. int. =1.0-min., initial gap =3649.4, and initial average gap =0.325)

| Iteration counter *k* | vehicle experienced cost gap | | | | average vehicle experienced cost gap | | | |
|---|---|---|---|---|---|---|---|---|
| | CGDDM | | | CGMSA | CGDDM | | | CGMSA |
| | MSA step | Unit step | mix step | | MSA step | unit step | mix step | |
| 1 | 1458.0 | 307.9 | 328.4 | 1046.9 | 0.130 | 0.027 | 0.029 | 0.093 |
| 2 | 1039.6 | 262.0 | 250.3 | 865.5 | 0.093 | 0.023 | 0.022 | 0.077 |
| 3 | 756.4 | 204.6 | 192.8 | 898.4 | 0.067 | 0.018 | 0.017 | 0.080 |
| 4 | 595.8 | 145.6 | 135.0 | 720.4 | 0.053 | 0.013 | 0.012 | 0.064 |
| 5 | 448.9 | 168.0 | 148.3 | 795.1 | 0.040 | 0.015 | 0.013 | 0.071 |
| 6 | 380.9 | 143.2 | 136.0 | 679.5 | 0.034 | 0.013 | 0.012 | 0.060 |
| 7 | 314.2 | 134.6 | 126.3 | 653.5 | 0.028 | 0.012 | 0.011 | 0.058 |
| 8 | | | 112.1 | 658.3 | | | 0.010 | 0.059 |
| 9 | | | 105.3 | 568.1 | | | 0.009 | 0.051 |
| 10 | | | | 857.5 | | | | 0.076 |
| 11 | | | | 579.8 | | | | 0.052 |
| 12 | | | | 669.6 | | | | 0.060 |
| 13 | | | | 597.4 | | | | 0.053 |
| 14 | | | | 554.0 | | | | 0.049 |

Table 3.2 Convergence patterns of CGDDM and CGMSA (aggregated cost gap

($K_{max}$ = 20, $L_{max}$ = 10, assign. int. = 1.0-min., initial gap = 5869.7, and initial average gap = 0.522)

| iteration counter | aggregated link cost gap | | | | average aggregated link cost gap | | | |
|---|---|---|---|---|---|---|---|---|
| | | CGDDM | | CGMSA | | CGDDM | | CGMSA |
| *k* | MSA step | Unit step | mix step | | MSA step | unit step | mix step | |
| 1 | 1812.8 | 811.4 | 884.6 | 1247.2 | 0.161 | 0.072 | 0.079 | 0.111 |
| 2 | 1342.9 | 642.6 | 648.8 | 1394.3 | 0.120 | 0.057 | 0.058 | 0.124 |
| 3 | 1042.4 | 500.0 | 419.2 | 1244.3 | 0.093 | 0.045 | 0.037 | 0.111 |
| 4 | 876.9 | 415.0 | 356.7 | 1227.8 | 0.078 | 0.037 | 0.032 | 0.109 |
| 5 | 760.1 | 499.7 | 300.7 | 1310.4 | 0.068 | 0.044 | 0.027 | 0.117 |
| 6 | 799.5 | 520.4 | 455.1 | 1378.5 | 0.071 | 0.046 | 0.041 | 0.123 |
| 7 | 654.6 | 383.7 | 283.3 | 1191.5 | 0.058 | 0.034 | 0.025 | 0.106 |
| 8 | 630.4 | 353.3 | 199.2 | 1258.6 | 0.056 | 0.031 | 0.018 | 0.112 |
| 9 | 564.6 | 360.3 | 240.2 | 1309.3 | 0.050 | 0.032 | 0.021 | 0.117 |
| 10 | 593.0 | 323.6 | 219.9 | 1306.7 | 0.053 | 0.029 | 0.020 | 0.116 |
| 11 | | 336.1 | 198.4 | 1272.0 | | 0.030 | 0.018 | 0.113 |
| 12 | | | | 1261.3 | | | | 0.112 |
| 13 | | | | 1107.4 | | | | 0.099 |

One important feature of the proposed CGDDM algorithm is that a (current) restricted path set is fully equilibrated (in the inner loop) before it is being augmented by adding new promising paths (in the outer loop). To evaluate the performance of this restricted path set equilibration scheme, a separate set of experiments with the parameter $L_{max}$ being varied between 1 and 15 but $K_{max}$ fixed at 20 is conducted on the nine-node network. The objective values, average gap values, reductions of initial gap and computation times of testing the CGDDM with different values of $L_{max}$ are reported in Table 3.3. It can be seen from this table that the marginal contribution (in reducing the initial objective value) of extra computation times diminishes as $L_{max}$ gets larger. For example, increasing $L_{max}$ from 1 to 2 results in 11.7% more reduction of the initial objective value), but would take 17 more seconds of computation time. Increasing $L_{max}$ from 5 to 10 results in only 3.5% more reduction of the initial objective value, but would take 89 more seconds of computation time. As mentioned earlier, this observation implies that the RNMP does not have to be solved optimally in each iteration *k*.

Table 3.3 Test of different $L_{max}$ on the nine-node network

(Mix step size, $K_{max}$ = 20, assign. int. = 1.0-min., experienced cost gap, and ini. gap = 3649.4)

| $L_{max}$ | 1 | 2 | 3 | 5 | 10 | 15 |
|---|---|---|---|---|---|---|
| Objective value (gap) | 1072.9 | 647.3 | 273.6 | 232.6 | 105.3 | 106.4 |
| Average Gap (min) | 0.095 | 0.058 | 0.024 | 0.021 | 0.009 | 0.009 |
| Reduction of initial gap (%) | 70.6% | 82.3% | 92.5% | 93.6% | 97.1% | 97.1% |
| Computation time (sec) | 45 (13*) | 62 (11) | 80 (10) | 124 (10) | 213 (9) | 302 (9) |

*: number of iterations required to converge

One other experiment conducted on the nine-node network aims at investigating the impact of the departure (or assignment) interval on the performance of the CGDDM. Since the gap value that measures the deviation of the current path flow pattern from a DUE is the sum of (average) experienced path-cost differences weighted by the path flows over all the paths for each OD pair and each departure interval, the size of the departure interval would be expected to affect the magnitude of the total gap and the average gap. In this experiment, $K_{max}$ and $L_{max}$ are set to 20 and 10, respectively. The results of executing the CGDDM with four different assignment intervals are compared and presented in Table 3.4. It shows that the solution quality in terms of the objective value and average gap of the CGDDM is not sensitive to the length of the assignment interval. On the other hand, the solution quality of the CGMSA improves significantly as the length of the assignment interval gets smaller. When the finest assignment interval is used, the solution quality of the CGMSA can be close to that of the CGDDM.

Table 3.4 Test of different assignment intervals on the nine-node network

($K_{max}$ = 20, $L_{max}$ = 10, and vehicle experienced cost gap)

| Algorithm | CGDDM with mix step size | | | | CGMSA | | | |
|---|---|---|---|---|---|---|---|---|
| Assignment interval (min) | 0.1 | 1.0 | 2.0 | 5.0 | 0.1 | 1.0 | 2.0 | 5.0 |
| Initial objective value | 3280.1 | 3649.4 | 3662.6 | 3602.2 | 3280.1 | 3649.4 | 3662.6 | 3602.2 |
| Objective value | 94.4 | 105.3 | 119.1 | 124.9 | 217.9 | 554.0 | 700.3 | 1013.4 |
| Average gap (min) | 0.008 | 0.009 | 0.011 | 0.011 | 0.019 | 0.049 | 0.062 | 0.091 |
| Reduction of initial gap (%) | 97.1% | 97.1% | 96.7% | 96.5% | 93.4% | 84.8% | 80.9% | 71.9% |

To further validate whether or not the path assignment obtained by the CGDDM satisfies the DUE conditions, a solution of the CGDDM on the nine-node network is investigated and presented in Table 3.5, where for each 5-minute assignment (departure) interval the number of vehicles and travel time on all paths connecting the OD pair (1, 3) are reported. There are 6 paths connecting the OD pair (1, 3) with the following node sequences.

Path 1: 1 → 2 → 3 → 6 → 9, Path 2: 1 → 2 → 5 → 6 → 9, Path 3: 1 → 2 → 5 → 8 → 9
Path 4: 1 → 4 → 5 → 6 → 9, Path 5: 1 → 4 → 5 → 8 → 9, Path 6: 1 → 4 → 7 → 8 → 9

For any path in a given assignment interval, if there are vehicles on that path (i.e. that path is used), the path time is obtained as the average experienced trip time of all the vehicles assigned to that path; otherwise (that path is not used), the path time of that path is determined by summing up the link times of constituent links which are output from the dynamic network loading model (i.e. DYNASMART) and used in the path generation step of the CGDDM algorithm. It can be seen from Table 3.5 that for all assignment intervals, the average experienced path times on all the used paths are very close. All the unused paths have path travel times greater than or equal to those of the used paths (i.e. no new better paths can be found). Thus, the result demonstrates that the solution (path flow pattern) obtained by the CGDDM sufficiently satisfies the DUE conditions. It is important to note that this solution may not be a unique global one (see Sheffi, 1985, about the non-uniqueness of path flows)

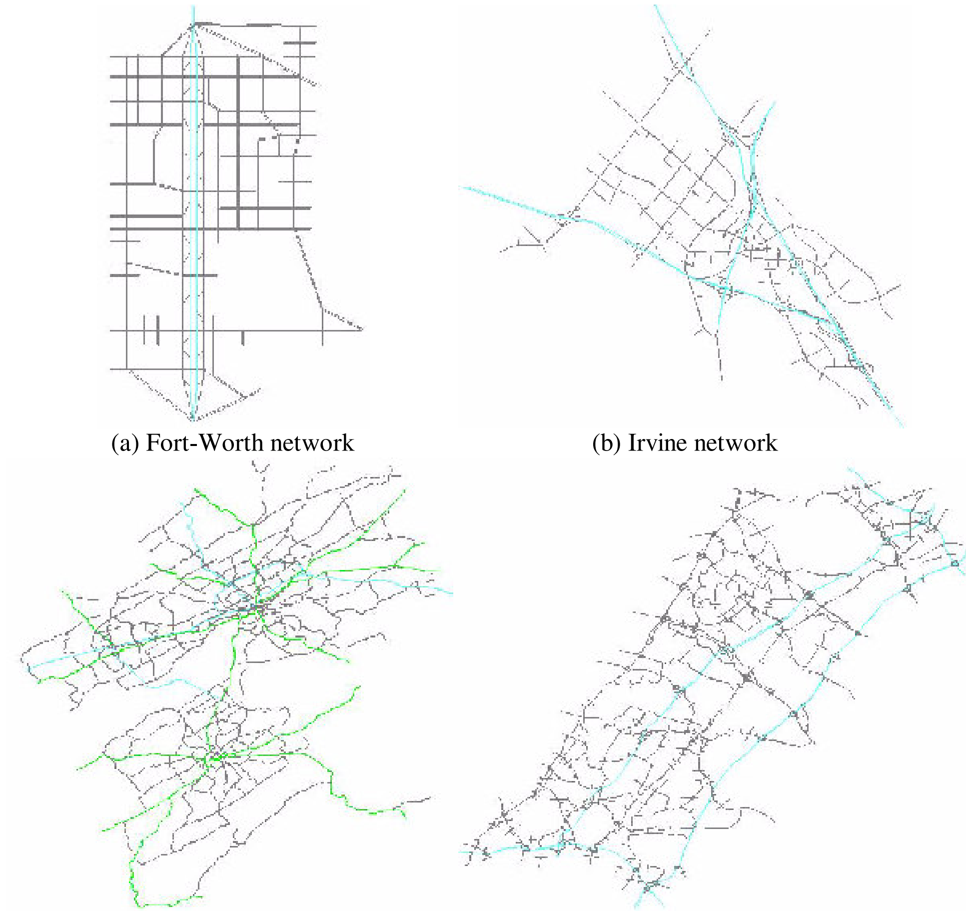
Table 3.5 Validation of a solution of CGDDM on the nine-node network

(mix step, $K_{max}$ =20, $L_{max}$ =10, assign. int. = 5-min., and vehicle experienced cost gap)

| Assignment terval (minute) | number of vehicles on each path | | | | | | path time on each path (minute) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 5 | 6 |
| (11-15) | 33 | 76 | 89 | 0 | 0 | 100 | 3.98 | 3.96 | 4.00 | 5.00 | 5.67 | 3.97 |
| (16-20) | 64 | 56 | 72 | 0 | 0 | 109 | 3.98 | 4.00 | 4.00 | 4.88 | 5.64 | 3.99 |
| (21-25) | 88 | 35 | 112 | 0 | 0 | 93 | 4.01 | 4.00 | 4.00 | 5.17 | 5.66 | 4.00 |
| (26-30) | 88 | 48 | 127 | 0 | 0 | 105 | 3.99 | 4.07 | 4.00 | 4.75 | 5.56 | 4.01 |
| (31-35) | 73 | 19 | 170 | 0 | 0 | 105 | 3.98 | 4.03 | 4.00 | 5.09 | 5.19 | 4.00 |
| (36-40) | 117 | 35 | 79 | 0 | 32 | 86 | 3.99 | 4.03 | 4.00 | 4.86 | 4.03 | 3.99 |
| (41-45) | 60 | 68 | 154 | 0 | 0 | 110 | 4.03 | 3.99 | 4.00 | 5.29 | 5.80 | 4.00 |
| (46-50) | 112 | 28 | 187 | 0 | 0 | 97 | 3.99 | 4.03 | 4.00 | 4.61 | 5.09 | 4.01 |
| (51-55) | 60 | 47 | 181 | 24 | 0 | 113 | 4.08 | 4.05 | 4.00 | 4.43 | 5.52 | 4.06 |
| (56-60) | 121 | 0 | 203 | 0 | 0 | 86 | 3.97 | 4.66 | 3.99 | 4.90 | 5.42 | 3.97 |
| (61-65) | 62 | 58 | 249 | 0 | 0 | 114 | 4.05 | 4.06 | 4.05 | 5.21 | 5.79 | 4.05 |
| (66-70) | 133 | 5 | 248 | 0 | 0 | 99 | 4.12 | 4.17 | 4.13 | 5.39 | 5.65 | 4.13 |
| (71-75) | 87 | 39 | 181 | 35 | 0 | 124 | 4.27 | 4.27 | 4.21 | 4.47 | 5.37 | 4.23 |
| (76-80) | 134 | 0 | 218 | 0 | 0 | 113 | 4.18 | 5.25 | 4.19 | 5.31 | 5.69 | 4.21 |
| (81-85) | 56 | 87 | 195 | 0 | 0 | 117 | 4.11 | 4.13 | 4.12 | 5.09 | 5.54 | 4.16 |
| (86-90) | 126 | 5 | 176 | 0 | 14 | 96 | 3.98 | 4.04 | 4.00 | 5.03 | 4.05 | 3.99 |
| (91-95) | 54 | 71 | 167 | 0 | 0 | 102 | 4.01 | 4.01 | 4.00 | 5.04 | 5.62 | 4.00 |
| (96-100) | 89 | 31 | 198 | 0 | 0 | 99 | 3.94 | 4.01 | 4.00 | 5.24 | 5.23 | 3.95 |

### 3.6.2 Experiments on real road networks

To explore the performance of the proposed CGDDM algorithm on solving DUE problems in real networks, the second set of experiments is conducted on four different real road networks with signalized intersections (see Figure 3.7); their basic characteristics are listed in Table 3.6.



(a) Fort-Worth network (b) Irvine network

(c) Knoxville network (d) Baltimore-Washington corridor network

Figure 3.7 Four real test networks

Table 3.6 Basic characteristics of the four real test networks

| Networks | # of zones | # of nodes | # of links | # of signals | # of vehicles in the observation period |
|---|---|---|---|---|---|
| Fort Worth, TX | 13 | 180 | 445 | 62 | 27,447 |
| Irvine, CA | 61 | 326 | 626 | 70 | 35,304 |
| Knoxville, TN | 106 | 1347 | 3004 | 110 | 86,483 |
| B-W, MD | 111 | 2182 | 3387 | 231 | 91,389 |

The first subset of experiments intends to compare the solutions (i.e. gap values) obtained by the CGDDM with three different step size rules and the CGMSA on these four real networks. Table 3.7(a) gives the computational results when the path costs $c(r)$ equal the sum of aggregated link costs of constituent links and the aggregated link cost gap is minimized. Table 3.7(b) presents the computational results when the path costs $c(r)$

are average vehicle experienced costs and the vehicle experienced cost gap is minimized. As shown in these two tables, the CGDDM with the mix step size performs relatively better in terms of minimizing the objective (gap) functions than the other three algorithms, although all the four algorithms are able to effectively find close-to-DUE solutions on real networks (all of them can reduce at least 70% of the initial gaps).

It is usually desirable to have sufficiently close-to-DUE solutions for large-scale network applications with some given constraints on the computational resources. If a solution with average gap less than 0.1 minutes (or 6 seconds) can be thought of as sufficiently good, then the CGDDM with mixed step requires 5 and 6 iterations to attain this level for minimizing the VEC gap and ALC gap, respectively, on the Fort-Worth network. Even for the much larger network, the B-W corridor network, the proposed algorithm only takes 5 and 4 iterations to obtain sufficiently good DUE solutions for the VEC gap and ALC gap, respectively.

Table 3.7(a) Performance of the algorithms on real networks – aggregated cost gap

($K_{max}$ = 50, $L_{max}$ = 5, and assign. int. = 1-minute)

| | | Fort-Worth | Irvine | Knoxville | B-W corridor |
|---|---|---|---|---|---|
| Algorithms | Initial gap values | 36110.0 | 3240.5 | 30384.3 | 41273.5 |
| CGDDM with Mixed step | Gap(r*) | 643.2 | 243.9 | 18.9 | 661.7 |
| | AGap(r*) | 0.023 | 0.007 | 0.000 | 0.007 |
| | Gap Reduction (%) | 98.2% | 92.5% | 99.9% | 98.4% |
| CGDDM with Unit step | Gap(r*) | 2135.4 | 526.4 | 78.0 | 1534.4 |
| | AGap(r*) | 0.078 | 0.015 | 0.001 | 0.017 |
| | Gap Reduction (%) | 94.1% | 83.8% | 99.7% | 96.3% |
| CGDDM with MSA step | Gap(r*) | 4561.7 | 627.3 | 478.7 | 4728.4 |
| | AGap(r*) | 0.166 | 0.018 | 0.006 | 0.052 |
| | Gap Reduction (%) | 87.4% | 80.6% | 98.4% | 88.5% |
| CGMSA | Gap(r*) | 7602.5 | 797.2 | 1876.4 | 10689.8 |
| | AGap(r*) | 0.277 | 0.023 | 0.022 | 0.117 |
| | Gap Reduction (%) | 78.9% | 75.4% | 93.8% | 74.1% |

Table 3.7(b) Performance of the algorithms on real networks – experienced cost gap

| | | Fort-Worth | Irvine | Knoxville | B-W corridor |
|---|---|---|---|---|---|
| Algorithms | Initial gap values | 27463.3 | 1858.3 | 18444.9 | 38629.8 |
| CGDDM with Mixed step | Gap(r*) | 524.4 | 88.8 | 27.3 | 864.0 |
| | AGap(r*) | 0.019 | 0.003 | 0.000 | 0.009 |
| | Gap Reduction (%) | 98.1% | 95.2% | 99.8% | 97.8% |
| CGDDM with Unit step | Gap(r*) | 1833.8 | 148.7 | 128.4 | 2159.2 |
| | AGap(r*) | 0.067 | 0.004 | 0.001 | 0.024 |
| | Gap Reduction (%) | 93.3% | 92.0% | 99.3% | 94.4% |
| CGDDM with MSA step | Gap(r*) | 3075.8 | 327.6 | 385.9 | 4663.3 |
| | AGap(r*) | 0.112 | 0.009 | 0.004 | 0.051 |
| | Gap Reduction (%) | 88.8% | 82.4% | 97.9% | 87.9% |
| CGMSA | Gap(r*) | 4030.0 | 543.4 | 1186.2 | 11782.1 |
| | AGap(r*) | 0.147 | 0.015 | 0.014 | 0.129 |
| | Gap Reduction (%) | 85.3% | 70.8% | 93.6% | 69.5% |

The second subset of experiments aims at examining the effect of the number of inner loop iterations (i.e. the restricted path set equilibration scheme) of the CGDDM on real networks. While $K_{max}$ is fixed at 50, $L_{max}$ is varied from 1 and 10. The assignment interval is 1.0 minute. The computational results of minimizing the aggregated link cost gap and optimizing the vehicle experienced cost gap on the Fort-Worth network are presented in Table 3.8(a) and Table 3.8(b), respectively. Without the restricted path set equilibration scheme (i.e. $L_{max}$ = 1), the performance (in terms of gap reductions) of the CGDDM and that of the CGMSA are similar. The CGMSA also has the computational advantage over the CGDDM when both have the same $L_{max}$. However, when $L_{max}$ is increased from 1 to 2, the CGDDM improves the objective values significantly and obtains very satisfactory DUE solutions (average gap is less than 0.1-minute). On the other hand, although increasing $L_{max}$ can also help CGMSA reduce the gaps, it requires more inner iterations (and hence longer computation time) to attain close-to-DUE solutions. Tables 3.9(a) and 3.9(b) give the computational results of minimizing the aggregated link cost gap and optimizing the vehicle experienced cost gap on the Irvine network, respectively. Similar observations to those on the Fort-Worth network can be found on the Irvine network.

Table 3.8(a) Test of different $L_{max}$ on the Fort-Worth network – aggregated cost gap

| | $L_{max}$ | 1 | 2 | 3 | 5 | 10 |
|---|---|---|---|---|---|---|
| CGDDM | Objective value (gap) | 12549.7 | 2048.3 | 1200.1 | 643.2 | 515.3 |
| | Average Gap (min) | 0.457 | 0.075 | 0.044 | 0.023 | 0.019 |
| | Reduction of initial gap (%) | 65.2% | 94.3% | 96.7% | 98.2% | 98.6% |
| | Computation time (hh:mm) | 03:04 | 03:28 | 03:51 | 04:29 | 05:51 |
| CGMSA | Objective value (gap) | 14877.3 | 13558.9 | 9210.8 | 7602.5 | 5364.9 |
| | Average Gap (min) | 0.542 | 0.494 | 0.336 | 0.277 | 0.195 |
| | Reduction of initial gap (%) | 58.8% | 62.5% | 74.5% | 78.9% | 85.1% |
| | Computation time (hh:mm) | 03:01 | 03:24 | 03:46 | 04:23 | 05:36 |

Table 3.8(b) Test of different $L_{max}$ on the Fort-Worth network – experienced cost gap

| | $L_{max}$ | 1 | 2 | 3 | 5 | 10 |
|---|---|---|---|---|---|---|
| CGDDM | Objective value (gap) | 12887.4 | 1466.0 | 906.8 | 524.4 | 367.0 |
| | Average Gap (min) | 0.470 | 0.053 | 0.033 | 0.019 | 0.013 |
| | Reduction of initial gap (%) | 53.1% | 94.7% | 96.7% | 98.1% | 98.7% |
| | Computation time (hh:mm) | 03:43 | 03:54 | 04:13 | 04:48 | 06:06 |
| CGMSA | Objective value (gap) | 13326.7 | 9789.0 | 8374.2 | 4030.0 | 2401.8 |
| | Average Gap (min) | 0.486 | 0.357 | 0.305 | 0.147 | 0.088 |
| | Reduction of initial gap (%) | 51.5% | 64.4% | 69.5% | 85.3% | 91.3% |
| | Computation time (hh:mm) | 03:37 | 03:48 | 04:05 | 04:38 | 05:51 |

Table 3.9(a) Test of different $L_{max}$ on the Irvine network – aggregated cost gap

| | $L_{max}$ | 1 | 2 | 3 | 5 | 10 |
|---|---|---|---|---|---|---|
| CGDDM | Objective value (gap) | 1256.2 | 496.2 | 320.2 | 243.9 | 209.9 |
| | Average Gap (min) | 0.036 | 0.014 | 0.009 | 0.007 | 0.006 |
| | Reduction of initial gap (%) | 61.2% | 84.7% | 90.1% | 92.5% | 93.5% |
| | Computation time (hh:mm) | 07:44 | 09:27 | 11:19 | 15:06 | 22:52 |
| CGMSA | Objective value (gap) | 2055.6 | 1840.8 | 1358.0 | 797.2 | 505.9 |
| | Average Gap (min) | 0.058 | 0.052 | 0.038 | 0.023 | 0.014 |
| | Reduction of initial gap (%) | 36.6% | 43.2% | 58.1% | 75.4% | 84.4% |
| | Computation time (hh:mm) | 07:30 | 09:14 | 11:03 | 14:47 | 22:21 |

Table 3.9(b) Test of different $L_{max}$ on the Irvine network – experienced cost gap

| | $L_{max}$ | 1 | 2 | 3 | 5 | 10 |
|---|---|---|---|---|---|---|
| CGDDM | Objective value (gap) | 584.3 | 217.8 | 121.2 | 88.8 | 51.1 |
| | Average Gap (min) | 0.017 | 0.006 | 0.003 | 0.003 | 0.001 |
| | Reduction of initial gap (%) | 68.6% | 88.3% | 93.5% | 95.2% | 97.3% |
| | Computation time (hh:mm) | 08:15 | 10:01 | 11:50 | 15:35 | 23:26 |
| CGMSA | Objective value (gap) | 1619.7 | 1480.1 | 782.1 | 543.4 | 343.8 |
| | Average Gap (min) | 0.046 | 0.042 | 0.022 | 0.015 | 0.010 |
| | Reduction of initial gap (%) | 12.8% | 20.4% | 57.9% | 70.8% | 81.5% |
| | Computation time (hh:mm) | 07:56 | 09:40 | 11:27 | 15:10 | 22:49 |

## 3.7 Summary

User equilibrium DTA models are used increasingly to describe and predict time-varying traffic network flow patterns, as well as to generate anticipatory and coordinated control and information supply strategies for intelligent traffic network management. The simulation-based approach has been successful at tackling many practical aspects that are essential in the application of DTA models in real networks, while the analytical approach contributes to theoretical insights about the problem and its solution. In a particular effort to improve the theoretical basis for simulation-based DTA models, this study addresses a series of critical and challenging issues in modeling and solving the UEDTA problem with known time-varying OD demands. This study proposes a reformulation of the DUE problem, via a gap function, as a nonlinear minimization problem (NMP) and then develops an efficient column generation-based optimization framework to integrate a (feasible) descent direction method that minimizes the objective function (i.e. the gap function) and a simulation-based dynamic traffic model that can generate realistic traffic flow patterns and the resulting experienced path travel times. Specifically, the column generation technique is able to avoid explicitly enumerating all feasible paths, and the descent direction method can circumvent the need for computing partial derivatives in estimating the gradient of the objective function. The adoption and integration of the above two methods, coupled with the embedded simulation-based dynamic traffic model, could enhance the development and deployment of large-scale simulation-based DTA models. Computational results on both small and large real road networks demonstrate that the proposed DUE algorithm is efficient and effective in obtaining close-to-DUE solutions.

# Chapter 4 Solving the Bi-Criterion Dynamic User Equilibrium Problem

## 4.1 Introduction

Conventional static or dynamic traffic assignment models for road pricing applications assume homogeneous perception of tolls for all trip-makers, so that every trip-maker is willing to tradeoff the same amount of money for a unit time saving, corresponding to the constant coefficients associated with the travel time and travel cost in the path generalized cost function (i.e. all trip-makers have the same value of time). However, empirical studies (e.g. Hensher, 2001; Brownstone and Small, 2005) have found that the value of time (VOT) varies significantly across individuals because of different socio-economic characteristics, trip purposes, and inherent preferences. This user heterogeneity is manifested in the fact that some trips take slower paths to avoid tolls while others choose toll roads to save time. Therefore, it is essential to explicitly recognize and represent heterogeneous users in modeling users' response to toll charges in DTA models for road pricing applications. This is especially important in assessing the feasibility of a proposed toll facility and its financial viability from the standpoint of the public or private entity that will be operating it.

Previous (static) traffic assignment studies that address user heterogeneity can be classified into two categories. The first category is the multi-class approach, in which the entire feasible VOT range is divided into several predetermined intervals according to a discrete VOT distribution or some socio-economic characteristics, such as income (Yang et al., 2002; Nagurney and Dong, 2002). The second category considers VOT to be continuously distributed across the population of trips. Leurent (1993) was among the first to propose a cost versus time network equilibrium model for road pricing applications; such equilibrium is achieved when every trip-maker is assigned a path that minimizes his/her own generalized cost. Dial (1996, 1997) developed a static bi-criterion user equilibrium traffic assignment model with continuous VOT to forecast path choice and associated total arc flows in the presence of tolled alternatives. Marcotte and Zhu (1997) considered the problem of determining an equilibrium state resulting from the interaction of infinitely many classes of customers, differentiated by a continuously distributed class-specific parameter. Solutions to the infinite dimensional VI problem were used to describe the equilibrium and obtained by an infinite dimensional extension of the Frank-Wolfe algorithm. For a thorough review and comparison of previous studies on multi-class and multi-criterion network equilibrium models readers may refer to Nagurney and Dong (2002).

This chapter presents the bi-criterion dynamic user equilibrium (BDUE) traffic assignment model which explicitly considers, in the underlying path choice model, heterogeneous trip-makers with different VOT choosing paths that simultaneously optimize the two essential path choice criteria: travel time and out-of-pocket cost. To realistically capture trip-makers' path choice decisions in response to toll charges, the VOT is assumed to be continuously distributed among trip-makers. Although this critical issue of user heterogeneity has been considered in the literature (see section 2.2), all those network equilibrium assignment models (e.g. Leurent, 1993; Dial, 1996; Marcotte and Zhu, 1997) were developed only for flat (static) road pricing schemes, rather than dynamic (or time-dependent) ones. In fact, successful design and evaluation of dynamic pricing schemes relies on a realistic representation of complex traffic dynamics and spatial and temporal vehicular interactions in traffic assignment models, hence necessitating the extension of the heterogeneous traffic assignment model from the static regime to the DTA context.

The BDUE problem is formulated as an infinite dimensional variational inequality (VI), and solved by the column generation-based algorithmic framework which embeds (i) the extreme non-dominated path finding algorithm – PAM (parametric analysis method) to obtain the breakpoints which partition the entire range of VOT into many subintervals and determine the multiple user classes, and find the least generalized cost path for each user class, (ii) the traffic simulator – DYANSMART (Jayakrishnan, et al. 1994; Mahmassani, 2001) to capture traffic dynamics and determine experienced path travel times for any given path flow pattern; and (iii) the multi-class path flow updating/equilibrating scheme to solve the restricted multi-class dynamic user equilibrium (RMDUE) problem defined by a subset of feasible paths. Moreover, to circumvent the difficulty of storing the memory-intensive path set and routing policies for large-scale network applications, a vehicle-based implementation technique using the vehicle path set as a proxy for keeping track of the path assignment results is applied.

This chapter is structured as follows. Section 4.2 presents the assumptions, definition and problem statement of the BDUE problem, followed by the infinite-dimensional VI formulation of the BDUE problem in section 4.3. In section 4.4 is the overview of a column generation-based solution algorithm for finding BDUE path flow patterns. The path-finding algorithm – PAM is presented in Section 4.5. Section 4.6 describes the RMDUE problem and the multi-class path flow updating scheme. Section

4.7 reports the results of numerical experiments which exam the path finding algorithm and illustrate the convergence behavior of the algorithm and how user heterogeneity affecting the path flow pattern and toll road usage under different dynamic road pricing scenarios. Section 4.8 summarizes this chapter.

## 4.2 Assumptions, Definition, and Problem Statement

Given a network G = ($N$, $A$), where $N$ is the set of nodes and A is the set of directed links ($i$, $j$), $i \in N$ and $j \in N$. The time period of interest (planning horizon) is discretized into a set of small time intervals, $S = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$, where $t_0$ is the earliest possible departure time from any origin node, $\sigma$ a small time interval during which no perceptible changes in traffic conditions and/or travel cost occur, and $M$ a large number such that the intervals from $t_0$ to $t_0 + M\sigma$ cover $S$. Denote $c_{ij}(t)$ and $d_{ij}(t)$ the travel cost (e.g. toll) and travel time, respectively, required for traveling on link ($i$, $j$) in time interval $t$. Note that $d_{ij}(t)$ may include both non-congested travel time and delay, while some other cost-related arc attributes can be considered in $c_{ij}(t)$. Presented below are the other important notations and variables used in this chapter.

| | |
|---|---|
| $o$ | subscript for an origin node, $o \in O \subseteq N$. |
| $d$ | subscript for a destination node, $d \in D \subseteq N$. |
| $\tau$ | superscript for a departure time interval, $\tau = 1, \ldots, T$. |
| $\alpha$ | value of time (VOT), $\alpha \in [\alpha^{\min}, \alpha^{\max}]$. |
| $P(o,d,\tau)$ | the set of feasible paths for a given triplet ($o$, $d$, $\tau$). |
| $p$ | subscript for a path $p \in P(o, d, \tau)$. |
| $h_{od}^{\tau}(\alpha)$ | the number of trips with VOT $\alpha$ departing from $o$ to $d$ in time interval $\tau$. |

$r^{\tau}_{odp}(\alpha)$ — the number of trips with VOT $\alpha$ departing from $o$ to $d$ in time interval $\tau$ that are assigned to path $p \in P(o, d, \tau)$.

$r(\alpha)$ — the class-specific time-varying path flow vector for the trips with VOT $\alpha$; i.e. $r(\alpha) \equiv \{ r^{\tau}_{odp}(\alpha), \forall o, d, \tau, \text{ and } p \in P(o, d, \tau)\}$.

$r$ — the time-varying (possibly infinite) multi-class path flow vector for the trips with all possible values of time; i.e. $r \equiv \{r(\alpha), \forall \alpha \in [\alpha^{\min}, \alpha^{\max}]\}$.

$TT^{\tau}_{odp}$ — experienced path travel time for the trips departing from $o$ to $d$ at time $\tau$ assigned to path $p \in P(o, d, \tau)$.

$\boldsymbol{TT}$ — vector of experienced path times; $\boldsymbol{TT} = \{ TT^{\tau}_{odp}, \forall o, d, \tau, \text{ and } p \in P(o, d, \tau)\}$.

$TC^{\tau}_{odp}$ — experienced path travel cost for the trips departing from $o$ to $d$ at time $\tau$ assigned to path $p \in P(o, d, \tau)$.

$\boldsymbol{TC}$ — vector of experienced path costs; $\boldsymbol{TC} = \{ TC^{\tau}_{odp}, \forall o, d, \tau, \text{ and } p \in P(o, d, \tau)\}$.

The experienced path generalized cost perceived by the trip-makers (or trips) with VOT $\alpha$ departing from $o$ to $d$ at time $\tau$ assigned to path $p \in P(o,d,\tau)$ is defined as:

$$GC^{\tau}_{odp}(\alpha) = \sum\nolimits_{(i,j,t)\in p} [c_{ij}(t) + \alpha \times d_{ij}(t)] = TC^{\tau}_{odp} + \alpha \times TT^{\tau}_{odp} , \tag{4.1}$$

where $TT^{\tau}_{odp} = \sum_{(i,j,t)\in p} d_{ij}(t)$ and $TC^{\tau}_{odp} = \sum_{(i,j,t)\in p} c_{ij}(t)$. The VOT relative to each trip represents how much money the trip-maker is willing to trade for a unit time saving. To realistically reflect heterogeneity of the population, the VOT in this study is treated as a continuous random variable distributed across the population of trip-makers, with the density function $\phi(\alpha)>0$, $\forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$ and $\int_{\alpha^{\min}}^{\alpha^{\max}} \phi(\alpha) d\alpha = 1$, where the feasible range of VOT is given by the closed interval $[\alpha^{\min}, \alpha^{\max}]$. Note that the distribution of VOT which can be estimated from survey data (e.g., Small et al., 2005) or loop detector data (e.g. Liu et al., 2004 and 2007) is assumed known, a priori. The time-dependent origin-destination (OD) demands for the entire feasible range of VOT over the planning horizon (i.e. $h_{od}^{\tau}(\alpha)$, $\forall o$, $d$, $\tau$, and $\alpha \in [\alpha^{\min}, \alpha^{\max}]$) are also assumed known, a priori (the OD pattern and the VOT distribution are considered independent of each other).

The key behavioral assumption made for the path choice decision is: each trip-maker would choose a path that minimizes the path generalized cost function, defined in Eq.(4.1). Specifically, for trips with VOT $\alpha$, a path $p^* \in P(o,d,\tau)$ will be selected if and only if $GC_{odp^*}^{\tau}(\alpha) = \min_{p \in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha)$. Based on this assumption, the bi-criterion dynamic user equilibrium (BDUE), a bi-criterion and dynamic extension of Wardrop's first principle, is defined as:

**Definition 4.1: BDUE**

*For each OD pair and for each departure time interval, every trip-maker cannot decrease the experienced path generalized cost with respect to that trip-maker's particular VOT by unilaterally changing path.*

This implies that, at BDUE, each trip-maker is assigned to a path with the least generalized cost with respect to his/her own VOT. This definition can be viewed as the dynamic extension of Dial's bi-criterion user equilibrium (1996) or Leurent's cost versus time equilibrium (1993). Since trips with different VOT (now a continuously distributed random variable) are assigned onto the same road network, the generalization of the classical dynamic user equilibrium problem (i.e. the BDUE problem) allows a large number of classes of trips to be in a simultaneous equilibrium. In the extreme case where each possible VOT corresponds to a class of trips, solving for the BDUE is equivalent to determining an equilibrium state resulting from the interactions of (possibly) infinitely many classes of trips in a network. Their interactions can be reflected by assuming the (measured or actual) time-dependent path travel time functions is a function of the time-varying multi-class path flow vector $r$ (i.e. $TT_{odp}^{\tau} = TT_{odp}^{\tau}(r)$, $\forall o, d, \tau,$ and $p \in P(o,d,\tau)$). Note that time-dependent path travel costs are assumed flow independent as link costs are considered as the input of the model from any given dynamic road pricing scheme. By definition, the path generalized cost perceived by trips with VOT $\alpha$ also depends on $r$: $GC_{odp}^{\tau}(\alpha, r) = TC_{odp}^{\tau} + \alpha \times TT_{odp}^{\tau}(r)$.

Based on the above definition, the BDUE conditions can be mathematically stated as the following: $\forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$,

$$r_{odp}^{\tau}*(\alpha)[GC_{odp}^{\tau}(\alpha, r^*) - \pi_{od}(\alpha, r^*)] = 0, \ \forall o, d, \tau, p \in P(o,d,\tau), \quad (4.2)$$

$$GC_{odp}^{\tau}(\alpha, r^*) - \pi_{od}^{\tau}(\alpha, r^*) \geq 0, \ \forall o, d, \tau, p \in P(o,d,\tau), \quad (4.3)$$

$$\sum_{p \in P(o,d,\tau)} r_{odp}^{\tau}(\alpha) = h_{od}^{\tau}(\alpha), \ \forall o, d, \tau \quad (4.4)$$

$$r_{odp}^{\tau}(\alpha) \geq 0, \ \forall o, d, \tau, p \in P(o,d,\tau), \quad (4.5)$$

where $r^* = \{r_{odp}^{\tau}*(\alpha)\}$ is a multi-class time-varying BDUE path flow vector, and $\pi_{od}^{\tau}(\alpha, r^*)$ is the time-varying minimum OD generalized travel cost, evaluated at $r^*$, for the trips with the same $(o, d, \tau, \alpha)$. Given the assumptions and definition above, this study aims at solving the BDUE problem, under a given dynamic road pricing scheme, to obtain a time-varying path flow vector satisfying the BDUE conditions. Specifically, the focus is on determining the BDUE path flows (routing policies) in a vehicular network: $r_{odp}^{\tau}(\alpha)$, $\forall o, d, \tau, p \in P(o,d,\tau)$ and $\forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$.

## 4.3 Infinite Dimensional VI Formulation of the BDUE

Let $\Omega(\alpha) \equiv \{r(\alpha)\}$ be the set of feasible class-specific path flow vectors $r(\alpha)$ satisfying the path flow conservation constraints (4.4) and non-negativity constraints (4.5). The following proposition gives the equivalent VI formulation of the BDUE problem of interest.

**Proposition 4.1**: Solving for the BDUE flow pattern $r^*$ is equivalent to finding the solution of a system of variational inequalities: $r^*(\alpha)\in\Omega(\alpha)$ such that

$$\sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha, r^*)\times(r_{odp}^{\tau}(\alpha) - r_{odp}^{\tau}(\alpha)^*) \geq 0\,,$$

$$\forall\ r(\alpha)\in\Omega(\alpha), \text{ and } \forall\alpha\in[\alpha^{\min}, \alpha^{\max}], \tag{4.6}$$

or in the following vector form for simplicity and clarity:

$$GC(\alpha, r^*)^T \circ (r(\alpha) - r^*(\alpha)) \geq 0,\ \forall\ r(\alpha)\in\Omega(\alpha), \text{ and } \forall\alpha\in[\alpha^{\min}, \alpha^{\max}], \tag{4.7}$$

where $GC(\alpha, r^*)$ is the path generalized cost vector perceived by the trips with VOT $\alpha$ and evaluated at flow pattern $r^*$, and $\circ$ denotes the inner product of the two vectors: $GC(\alpha, r^*)$ and $(r^*(\alpha) - r(\alpha))$. Since (4.6) or (4.7) is only required to hold on $[\alpha^{\min}, \alpha^{\max}]$, it can be further represented by the following infinite-dimensional VI (see e.g. Marcotte and Zhu, 1997): find $r^* \equiv \{r^*(\alpha), \forall\alpha\in[\alpha^{\min}, \alpha^{\max}]\}$ and $r^*\in\Omega$ such that

$$GC(r^*)^T \circ (r - r^*) \geq 0,\ \forall\ r\in\Omega \tag{4.8}$$

where $GC(r^*)\equiv\{GC(\alpha, r^*), \forall\alpha\in[\alpha^{\min}, \alpha^{\max}]\}$, and $\Omega=\{r\}=\{\Omega(\alpha), \forall\alpha\in[\alpha^{\min}, \alpha^{\max}]\}$. Note that $GC(r^*)$ and $r^*$ (or $r$) have the same (possibly infinite) number of elements.

**Proof of Proposition 4.1**:

Suppose $r^*$ is a BDUE path flow vector, and let $GC(r^*)$ be the corresponding path generalized cost vector. We first establish that $r^*$ is a solution to the VI problem (4.6). From the BDUE condition (4.2), the following inequalities can be obtained.

$$GC_{odp}^{\tau}(\alpha, r^*)[r_{odp}^{\tau}(\alpha) - r_{odp}^{\tau}*(\alpha)] \geq \pi_{od}(\alpha, r^*)[r_{odp}^{\tau}(\alpha) - r_{odp}^{\tau}*(\alpha)]$$
$$\forall o, d, \tau, p \in P(o,d,\tau) \text{ and } \forall \alpha \in [\alpha^{\min}, \alpha^{\max}] \tag{4.9}$$

With the path flow conservation constraints (4.4), it follows that

$$\begin{aligned}&\sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d)} GC_{odp}^{\tau}(\alpha, r^*)[r_{odp}^{\tau}(\alpha) - r_{odp}^{\tau}*(\alpha)]\\ &\geq \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\pi_{od}^{\tau}(\alpha, r^*)\{\sum_{p\in P(o,d,\tau)}[r_{odp}^{\tau}(\alpha) - r_{odp}^{\tau}*(\alpha)]\} = 0\end{aligned}, \forall \alpha \in [\alpha^{\min}, \alpha^{\max}] \tag{4.10}$$

Hence, $r^*$ is a solution to the VI problem (4.6).

We then show that a solution $r^*$ to the VI problem (4.6) is a BDUE path flow vector which satisfies conditions (4.2)–(4.5). Eq.(4.6) can be rearranged as the following:

$$\sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha, r^*)\times r_{odp}^{\tau}(\alpha) \geq \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha, r^*)\times r_{odp}^{\tau}*(\alpha),$$
$$\forall r(\alpha) \in \Omega(\alpha) \text{ and } \forall \alpha \in [\alpha^{\min}, \alpha^{\max}]. \tag{4.11}$$

It can seen from (4.11) that, $\forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$, $r*(\alpha) \in \Omega(\alpha)$ is an optimal solution to the linear program

$$Minimize \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha, r^*)\times r_{odp}^{\tau}(\alpha) \tag{4.12}$$

Subject to (4.4) and (4.5)

Let $\pi_{od}^{\tau}(\alpha, r^*)$ , $\forall o, d, \tau$ be the corresponding dual variables for the path flow conservation constraints (4.4). Then (4.2) follows from complementary slackness, (4.3) follows from dual feasibility, and (4.4) and (4.5) follow from primal feasibility. Therefore, $r^*$ is a BDUE path flow vector. This completes the proof.

Although the theoretical guarantee of properties such as existence and uniqueness of solutions to the VI problem (4.6) (or the infinite dimensional VI (4.8)) can be analytically derived, it generally requires the path generalized cost function, i.e. $GC_{odp}^{\tau}(\alpha, r)$, to be continuous and strictly monotone (see e.g. Marcotte and Zhu, 1997). Those properties of path cost functions might not be satisfied in general road networks with complex traffic controls, and hence only close-to-BDUE (multiple optima) solutions can be obtained if the condition for solution existence (uniqueness) fails to be established. The discussion of solution existence and uniqueness is beyond the scope of this study.

## 4.4 BDUE Solution Algorithm

### 4.4.1 Overview of the column generation-based algorithmic framework

Since the BDUE problem of interest seeks equilibrium network states in terms of path generalized costs of network users, a set of feasible paths on which the time-varying and heterogeneous OD demands are to be equilibrated is required for the BDUE solution algorithm. It is generally very difficult, if not impossible, to enumerate the complete set of feasible paths for all OD pairs and all possible VOT in a road network of practical size. Furthermore, only a (small) fraction of paths would carry positive flows in a BDUE solution. To avoid explicit enumeration of all possible paths, this study applies a column generation-based approach that generates a representative subset of paths with competitive generalized cost and augments the path set as needed.

The column generation-based approach augments, in the outer loop, the subset of the feasible (extreme efficient or non-dominated) paths and solves, in the inner loop, the "restricted" multi-class DUE (RMDUE) problem defined by the current subset of feasible paths. In each outer iteration *k*, the extreme non-dominated path finding algorithm – parametric analysis method (PAM) is applied to (i) obtain the breakpoints which partition the entire range of VOT into many subintervals and determine the multiple user classes, and (ii) find the least generalized cost (i.e. extreme efficient or non-dominated) path for each user class. New paths, if any, are added to the current path set. The algorithm terminates if there is not any new path found for all user classes or a preset convergence criterion is satisfied; otherwise the RMDUE problem is solved by adopting the multi-class path flow updating scheme to equilibrate time-varying and heterogeneous OD demands on the current path set, before returning to the path generation step (i.e. outer loop). This multi-class path flow updating/equilibrating scheme proceeds iteratively and forms the inner loop (with iteration counter *l*) of the column generation-based solution framework, in a manner similar to the descent direction method proposed in Chapter 3 or the restricted path set equilibration scheme suggested by Larsson and Patriksson (1992). By and large, the original BDUE problem is solved in this algorithmic framework as a series of approximate RMDUE problems to progressively find BDUE solutions. This idea of obtaining VOT breakpoints that naturally determine multiple user classes and solving the RMDUE problem by equilibrating path flows in each user class bases on the assumption that, in the disutility minimization-based path choice modeling framework with convex disutility (i.e. path generalized cost) functions, all trips would choose only among the set of extreme efficient (or non-dominated) paths, and the trips in each user class behave similarly in their path choices (e.g. Dial, 1996; Marcotte and Zhu, 1997).

It is worth noting that, as also suggested by early studies on the diagonalization algorithm for asymmetric traffic assignment problems (see e.g. Sheffi, 1985; Mahmassani and Mouskos, 1988) and the experimental results reported in Chapter 3, the RMDUE problem does not have to be solved optimally in each iteration $k$, in order to strike the balance between computational efficiency and satisfactory convergence. Also embedded in this algorithmic framework is the traffic simulator – DYNASMART (Jayakrishnan et al., 1994; Mahmassani, 2001), that performs multi-class dynamic network loadings (MDNL) to determines link travel times and experienced path generalized costs for any given path flow pattern $r$; traffic flow propagations and the vehicular spatial and temporal interactions are addressed through the traffic simulation instead of analytical calculations. The column generation-based BDUE solution algorithm is outlined below and its flow chart is presented in Figure 4.1.

**Initialization**

0. Input: (I) time-dependent OD demands for the entire feasible range of VOT over the planning horizon ( $h_{od}^{\tau}(\alpha)$ , $\forall o$, $d$, $\tau$, and $\alpha \in [\alpha^{\min}, \alpha^{\max}]$), (II) time-dependent link tolls, (III) VOT distribution function, and (IV) initial paths and path assignment.

1. Set the outer loop iteration counter $k = 0$. Perform a MDNL by the traffic simulator to evaluate the initial path assignment and obtain time-dependent link travel times and experienced path travel times and costs (i.e. ***TT*** and ***TC***).

**Outer Loop – generating extreme efficient path set**

2. Use the parametric analysis method (PAM) to obtain the set of time-dependent extreme efficient paths, their corresponding generalized costs ($\pi^k$) and breakpoints of VOT that partition the entire feasible VOT range and define the multi-user classes.

3. Convergence checking: if (a) there is not any new path found or (b) $k = K_{max}$ (maximum number of iterations) then stop; otherwise start the inner loop (step 4).

**Inner Loop – solving the RMDUE sub-problem**

4. Set the inner loop iteration counter $l = 0$; read the output of step 2: $\pi^l$ and VOT breakpoints, as well as the current path set (and ***TT*** and ***TC***) and path assignment ($r^l$).

5. Update path assignment: determine path assignment $r^{l+1}$ by using the multi-class path flow updating/equilibrating scheme. Set $l = l + 1$.

6. Perform a MDNL by the traffic simulator (DYANSMART) to evaluate the new path assignment $r^l$ and obtain experienced path travel times and costs (i.e. ***TT*** and ***TC***).

7. Convergence checking: if the preset convergent threshold is reached or $l = L_{max}$ (maximum number of inner iterations), then set $k = k+1$ and return to step 2 with current link travel times; otherwise go back to step 5.

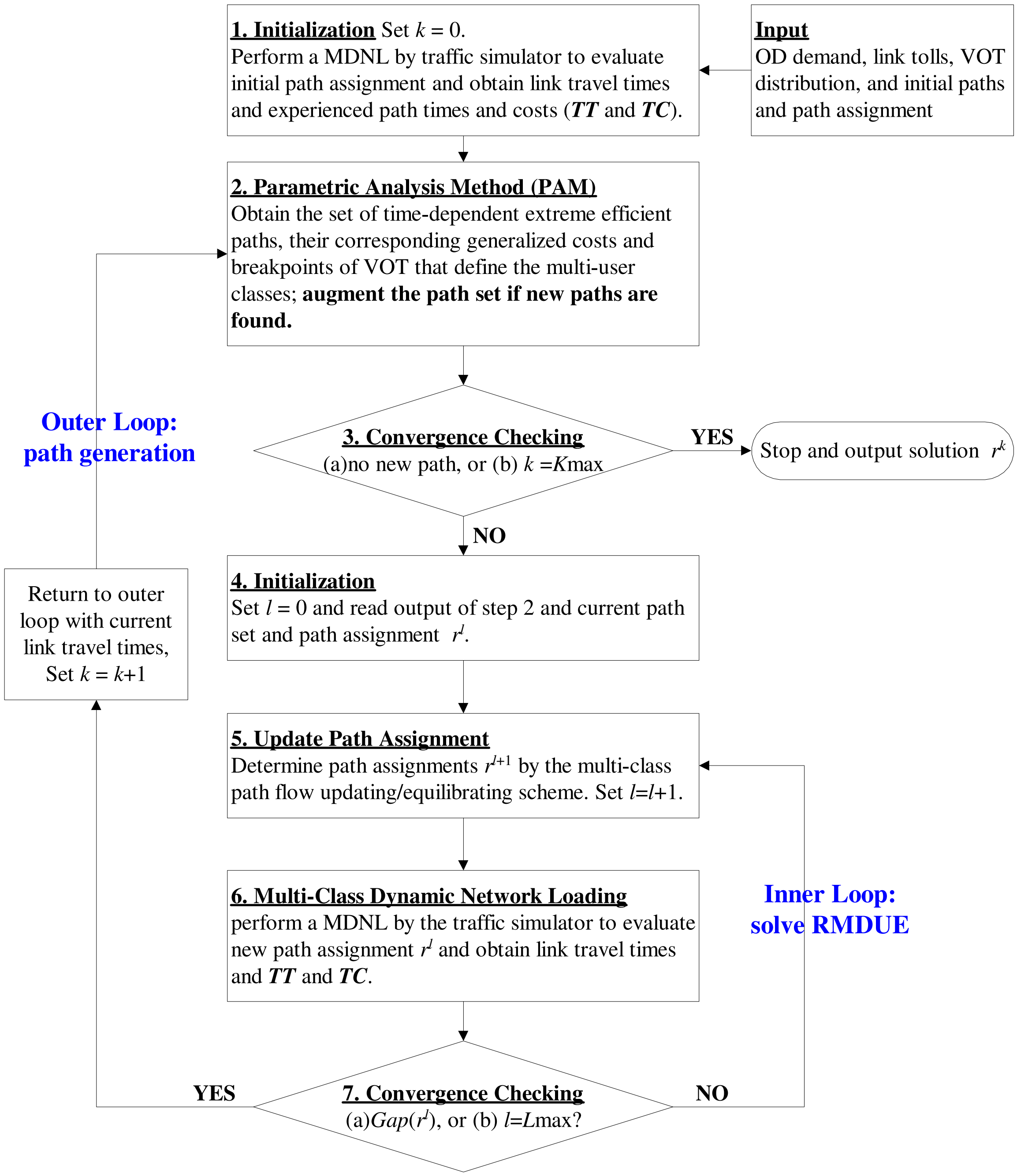



Figure 4.1 Flow chart of the BDUE solution algorithm

## 4.5 Augmenting the Extreme Efficient Path Set

The main impediment for solving the BDUE problem of interest is due largely to the relaxation of VOT from a constant to a continuous random variable and hence the need to find an equilibrium state resulting from the interactions of (possibly infinitely) many classes of trips, each of which corresponds to a class-specific VOT, in a network. If, in the extreme case, each trip-maker (or class) requires its own set of time-dependent least generalized cost paths, finding and storing such a grand path set is computationally intractable and memory intensive in (road) network applications of practical sizes. In order to circumvent the difficulty of finding and storing the least generalized cost path for each individual trip-maker with different VOT, the Parametric Analysis Method (PAM) is proposed to find the set of extreme efficient path trees, each of which minimizes the parametric path generalized cost function Eq.(4.1) for a particular VOT subinterval. The idea of finding the set of extreme efficient paths on which and heterogeneous trips are to be assigned is based on the assumption (see e.g. Dial, 1996; Marcotte and Zhu, 1997) that in the disutility minimization-based path choice modeling framework with convex disutility functions, all trips would choose only among the set of extreme efficient paths corresponding to the extreme points on the efficient frontier in the criterion space, defined in the following.

**Definition 4.1** A path $p \in P(o,d,\tau)$ is efficient (Pareto Optimal or non-dominated) if and only if it is not possible to find a different path $q \in P(o,d,\tau)$ such that $TT_{odq}^{\tau} \leq TT_{odp}^{\tau}$ and $TC_{odq}^{\tau} \leq TC_{odp}^{\tau}$ with at least one strict inequality.

An efficient path $p$ in the solution space corresponds to an efficient point $Z(p) = (TT^{\tau}_{odp}, TC^{\tau}_{odp})$ in the criterion space. The set of efficient points is denoted as Z.

**Definition 4.2** If an efficient point Z($p$) lies on the boundary of the convex hull of Z, then Z($p$) is an extreme efficient point and $p$ is an extreme efficient path; otherwise Z($p$) is a non-extreme efficient point and $p$ is a non-extreme efficient path.

### 4.5.1 Parametric analysis of VOT ($\alpha$)

This subsection presents the parametric analysis of VOT which sequentially computes a (complete) set of time-dependent extreme efficient path trees, each of which corresponds to a VOT subinterval (i.e. optimizes the path generalized cost function Eq.(4.1)) for that VOT subinterval) and consists of time-dependent least generalized cost (TDLGC) paths from a given origin node, for all departure time intervals, to all the other (destination) nodes in a network. This parametric analysis method (PAM) can be viewed as a time-dependent adaptation of the static parametric approach (e.g. Henig, 1985; Mote et al., 1991; Dial, 1997).

Relying on efficiently finding the time-dependent extreme efficient tree $T(\alpha)$ for a given VOT $\alpha$, the PAM adopts the computationally efficient time-dependent least cost path (TDLCP) algorithm, developed by Ziliaskopoulos and Mahmassani (1993). Denote the time-dependent link generalized cost function of a given arc ($i$, $j$) by the following linear form:

$$g_{ij}(t) = c_{ij}(t) + \alpha \times d_{ij}(t) \quad (4.13)$$

Each node $i \in N$ is associated with three label vectors: $\boldsymbol{\delta}_i = \{\delta_i(t)\}$, $\boldsymbol{\gamma}_i = \{\gamma_i(t)\}$, and $\boldsymbol{\eta}_i = \{\eta_i(t)\}$ $\forall t \in S$, corresponding to travel time, travel cost, and generalized cost, respectively, of paths from origin $r$ to node $i$ for each time interval $t$ in the planning horizon. The TDLCP algorithm is based on Bellman's general principle of optimality, and the least (generalized) cost paths are calculated forward, starting from the origin node (with no loss of generality). In each iteration, the algorithm selects and deletes the first node $i$, or "current node", from the scan eligible (SE) list. Then the current node $i$ is scanned and the labels of its downstream nodes are updated according to the following equation:

$$\eta_j(t+ d_{ij}(t)) = \min\{ \eta_j(t+ d_{ij}(t)), g_{ij}(t) + \eta_i(t)\}, \forall j\in \Gamma\{i\} \quad (4.14)$$

for every time $t\in S$, where $\Gamma\{i\}$ is the set of nodes that can be directly reached from $i$ (forward star). If at least one of the components of $\boldsymbol{\eta}_j$ is modified, node $j$ is inserted in the SE list, and the other three label vectors (i.e. $\boldsymbol{\delta}_j$ and $\boldsymbol{\gamma}_j$) are updated accordingly. The algorithm repeats this process and terminates when the SE list is empty. The output of the algorithm includes the time-dependent extreme efficient tree $T(\alpha)$ as well as the node vectors: $\boldsymbol{\delta}_i$, $\boldsymbol{\gamma}_i$, and $\boldsymbol{\eta}_i$ associated with each node $i$. In particular, vectors $\boldsymbol{\delta}_i$ and $\boldsymbol{\gamma}_i$ are used to calculate reduced travel times ($RT_{ij}(t) = \delta_i(t) + d_{ij}(t) - \delta_j(t')$) and reduced travel costs ($RC_{ij}(t) = \gamma_i(t) + c_{ij}(t) - \gamma_j(t')$), respectively, for all out-of-tree arc-time combinations. An arc-time combination $((i,j),t)$ is said to be out-of-tree if the following inequality holds:

$$\eta_i(t) + g_{ij}(t) - \eta_j(t + d_{ij}(t)) \geq 0. \quad (4.15)$$

These reduced link travel times and costs are essential input for the algorithm PAM.

**Algorithm**: Parametric Analysis Method (PAM)

Initialize the current value of VOT $\alpha = \alpha^{\min}$.

**WHILE** $\alpha < \alpha^{\max}$ **DO**

Update link generalized costs with current VOT $\alpha$

Apply the TDLCP algorithm to find the tree $T(\alpha)$

Initialize $\alpha^{\text{ub}} = \alpha^{\max}$

**FOR** each out-of-tree arc-time combination $((i, j), t)$ **DO**

Calculate $\alpha((i, j), t) = -RC_{ij}(t)/RT_{ij}(t)$

**IF** $\alpha((i, j), t) < \alpha^{\text{ub}}$ and $\alpha((i, j), t) > \alpha$, **THEN** $\alpha^{\text{ub}} = \alpha((i, j), t)$

**END FOR**

Set $\alpha = \alpha^{\text{ub}} + \Delta(\alpha)$, and output $\alpha$.

**END WHILE.**

**Proposition 4.1**: The PAM can find the complete set of time-dependent extreme efficient path trees, each of which optimizes the generalized path cost function for a VOT subinterval and consists of TDLGC paths from a given origin node, for all departure time intervals, to all the other (destination) nodes in a network.

**Proof of Proposition 4.1**:

The path finding algorithm is based on the following *parametric analysis* of the VOT. Consider a given VOT $\alpha$ and the corresponding time-dependent extreme efficient path tree $T(\alpha)$, consisting of the TDLGC paths from origin $r$, for each departure time interval $t$, to each node $i$. If an arc-time combination $((i, j), t)$ remains out-of-tree (i.e. non-tree arc), the corresponding reduced generalized cost should be nonnegative, leading to the inequality (4.15). For path $p(r,i,t)$, which starts from origin $r$, at time $t$, to node $i$, the node label with respect to generalized cost can be expressed as the sum of the node labels in terms of travel time and travel cost.

$$\eta_i(t) = \sum_{(k,l,\tau)\in p(r,i,t)} \left[c_{kl}(\tau) + \alpha \times d_{kl}(\tau)\right] = \gamma_i(t) + \alpha \times \delta_i(t) \tag{4.16}$$

Let $t' = t + d_{ij}(t)$; the generalized disutility for path $p(r,j,t')$ from origin $r$, at time $t'$, to node $j$ can similarly be represented as

$$\eta_j(t') = \gamma_j(t') + \alpha \times \delta_j(t') \quad (4.17)$$

Substituting Equations (4.13), (4.16) and (4.17) back into Inequality (4.15) yields

$$[\gamma_i(t) + c_{ij}(t) - \gamma_j(t')] + \alpha \times [\delta_i(t) + d_{ij}(t) - \delta_j(t')] \geq 0$$

$$\text{or} \quad RC_{ij}(t) + \alpha \times RT_{ij}(t) \geq 0 \quad (4.18)$$

Based on Inequality (4.18), the dependence of the least generalized cost path tree on the single scalar VOT can be examined. For any out-of-tree arc for which $RT_{ij}(t) \neq 0$, the following two cases determine the sensitivity range of VOT that does not violate the reduced-cost optimality conditions.

$$\text{If } RT_{ij}(t) > 0,\ \alpha > -RC_{ij}(t)/RT_{ij}(t) \quad (4.19)$$

$$\text{If } RT_{ij}(t) < 0,\ \alpha < -RC_{ij}(t)/RT_{ij}(t) \quad (4.20)$$

Collectively, we can calculate the lower and upper bounds of VOT by scanning each out-of-tree arc-time combination $((i, j), t)$,

$$\alpha^{lb} = \max_{((i,j),t) \notin Tr(\alpha)} \{-RC_{ij}(t) / RT_{ij}(t) \mid RT_{ij}(t) > 0\} \quad (4.21)$$

$$\alpha^{ub} = \min_{((i,j),t) \notin Tr(\alpha)} \{-RC_{ij}(t) / RT_{ij}(t) \mid RT_{ij}(t) < 0\} \quad (4.22)$$

The least generalized cost path tree $T(\alpha)$ remains unchanged as long as $\alpha^{lb} \leq \alpha \leq \alpha^{ub}$. In other words, the closed interval $[\alpha^{lb}, \alpha^{ub}]$ defines the (sensitivity) range of VOT for keeping tree $T(\alpha)$ optimal. The parametric analysis forms a main building block of PAM.

Starting from the minimal feasible value of VOT ($\alpha^{min}$), the PAM solves for the time-dependent extreme efficient path tree with respect to the current $\alpha$, and determines the upper bound $\alpha^{ub}$ for which the current shortest path tree $T(\alpha)$ remains unchanged, by the parametric analysis. This process continues until the maximal feasible value of VOT is reached. Based on the above parametric analysis, the algorithm is able to not only sequentially enumerate all possible time-dependent extreme efficient path trees (and all corresponding sensitivity ranges of VOT) but also directly move from one extreme efficient tree (and its sensitivity range of VOT) to the next one without redundant calculations on the non-extreme efficient solutions.

On the other hand, assume there is a time-dependent extreme efficient path tree not found by the PAM. However, by performing the parametric analysis on that tree, the sensitivity range of VOT $[\alpha^{\text{lb}},\alpha^{\text{ub}}]$ obtained can be found among the ranges already identified by the PAM, because it enumerate all the possible sensitivity ranges. That tree is actually included in the solution found by the PAM, and this contradicts the assumption. Thus, the PAM can find the complete set of time dependent extreme efficient path trees. This completes the proof.

Note that in order to move to the next VOT segment and obtain a different tree, a small positive value $\Delta(\alpha)$ needs to be added to the $\alpha^{\text{ub}}$ found in parametrically analyzing the current tree. This implies that trip-makers cannot distinguish differences in VOT below $\Delta(\alpha)$ per time unit. The value of $\Delta(\alpha)$ also implicitly sets an upper bound for the number of breakpoints generated using the PAM: $(\alpha^{\max}-\alpha^{\min})/\Delta(\alpha)$. In each iteration $k$, the PAM is applied to obtain the set of VOT breakpoints

$$\boldsymbol{\alpha} = \{\alpha^0, \alpha^1, ..., \alpha^B \mid \alpha^{\min} = \alpha^0 < \alpha^1 < ... < \alpha^b < ... < \alpha^B = \alpha^{\max}\}$$

that partitions the entire feasible range of VOT into $B$ subintervals: $[\alpha^{b-1}, \alpha^b)$, $b = 1,\ldots B$, and hence defines the $B$ master user classes of trips, each master user class $u(b)$ of which covers the trips with VOT $\alpha \in [\alpha^{b-1}, \alpha^b)$. Associated with each VOT subinterval $b$ (or master user class $u(b)$) is the time-dependent extreme efficient path trees: $Tr(b)$, which optimizes the path generalized cost function Eq.(4.1) for the corresponding VOT subinterval $[\alpha^{b-1},\alpha^{b})$ and consists of time-dependent least generalized cost (TDLGC) paths from a given origin node, for all departure time intervals, to all the other (destination) nodes in a network. If there is not any new path found for each $(o,d,\tau)$ and each user class $u(b)$, or the outer loop iteration counter $k$ equals $K_{max}$ (maximum number of outer iterations) then the algorithm terminate; otherwise it starts the inner loop with the output of the PAM: the set of VOT breakpoints ($\alpha$), as well as current path set and path assignment $r^k$.

## 4.6 Solving the RMDUE Problem

### 4.6.1 The RMDUE problem

With the set of VOT breakpoints ($\alpha$) determined by the PAM in a outer loop iteration $k$ of the column generation-based algorithmic framework, the entire population of heterogeneous trips in a network can be divided into a finite number of user classes, and hence the original (infinite-dimensional) BDUE problem of interest can be reduced to the (finite-dimensional) multi-class DUE problem, in which the equilibration within each user class is sought. Furthermore, since, in each iteration, the multi-class DUE is determined based on the current subset of feasible paths, the problem solved in the inner loop is termed the "restricted" multi-class DUE (or RMDUE) problem by following the terminology often adopted in the literature (e.g. Patriksson, 1994). Solving the RMDUE problem aims at finding a finite-dimensional multi-class path flow vector that satisfies the RMDUE definition: *for each user class, each OD pair, and each departure time interval, every trip cannot decrease the experienced path generalized cost by unilaterally* *changing paths*. The following variables and notations are defined (or redefined) for the RMDUE problem.

| | |
|---|---|
| $(b,o,d,\tau)$ | the combination of user class $u(b)$, OD pair $(o, d)$ and departure time interval $\tau$. |
| $P(b,o,d,\tau)$ | (current) subset of feasible time-dependent extreme efficient paths for a $(b,o,d,\tau)$. |
| $h_{od}^{\tau}(b)$ | number of class $u(b)$ trips departing from $o$ to $d$ in time interval $\tau$. |
| $h(b)$ | $\sum_o \sum_d \sum_\tau h_{od}^{\tau}(b)$; number of class $u(b)$ trips |
| $r_{odp}^{\tau}(b)$ | number of class $u(b)$ trips departing from $o$ to $d$ in time interval $\tau$ and assigned to path $p \in P(b,o,d,\tau)$. |
| $r_{od}^{\tau}(b)$ | $\equiv \{r_{odp}^{\tau}(b), \forall p \in P(b,o,d,\tau)\}$; path flow vector for class $u(b)$ trips departing from $o$ to $d$ in time interval $\tau$. |
| $r(b)$ | $\equiv \{r_{odp}^{\tau}(b), \forall o,d,\tau, p \in P(b,o,d,\tau)\}$; the class-specific path flow vector for the class $u(b)$ trips. |
| $r$ | $\equiv \{r(b), b = 1,..., B\}$; the multi-class path flow vector. |
| $GC_{odp}^{\tau}(b,r)$ | the path generalized cost of class $u(b)$ trips departing from $o$ to $d$ in time interval $\tau$ that are assigned to path $p \in P(b,o,d,\tau)$. |
| $GC(b,r)$ | $\equiv \{GC_{odp}^{\tau}(b,r), \forall o,d,\tau, p \in P(b,o,d,\tau)\}$ }, the class-specific path generalized cost vector perceived by the trips of class $u(b)$ and evaluated at flow pattern $r$. |
| $\pi_{od}^{\tau}(b,r)$ | least generalized cost of class $u(b)$ trips departing from $o$ to $d$ in time interval $\tau$, evaluated at the path assignment $r$. |

Let $\Omega(b) = \{r(b)\}$ be the set of feasible class-specific path flow vectors satisfying the path flow conservation and non-negativity constraints:

$$\sum_{o}\sum_{d}\sum_{\tau}\sum_{p\in P(b,o,d,\tau)} r_{odp}^{\tau}(b) = h(b), \forall b, \qquad (4.23)$$

$$r_{odp}^{\tau}(b) \geq 0, \forall b, o, d, \tau, p \in P(b,o,d,\tau) \geq 0. \qquad (4.24)$$

It can be obtained that, by adapting the result of Proposition 4.1, solving for the RMDUE flow pattern $r^*$ is equivalent to finding the solution of a system of variational inequalities: $r^*(b) \in \Omega(b)$, $b = 1,\ldots B$, such that

$$\sum_{o}\sum_{d}\sum_{\tau}\sum_{p\in P(b,o,d,\tau)} GC_{odp}^{\tau}(b,r)\times(r_{odp}^{\tau}{}^*(b) - r_{odp}^{\tau}(b)) \leq 0, \forall r(b) \in \Omega(b) \qquad (4.25)$$

or in the following vector form for simplicity and clarity:

$$GC(b,r^*)^T \circ (r^*(b) - r(b)) \leq 0, \forall r(b) \in \Omega(b), b = 1,\ldots,B \qquad (4.26)$$

where $\circ$ denotes the inner product of the two vectors: $GC(b,r^*)$ and $(r^*(b) - r(b))$.

### 4.6.2 Multi-class path flow updating/equilibrating scheme

In the inner loop of the column generation-based algorithmic framework is a multi-class path flow updating (or equilibrating) scheme to solve the RMDUE problem and to update path assignments. This multi-class path flow updating scheme is a projection type algorithm that decomposes the RMDUE problem into many $(b,o,d,\tau)$ sub-problems and solves each of them by adjusting time-varying OD flows between (all) non-least generalized cost paths and the least generalized cost path(s). Given a feasible solution $r^l$ in an inner loop iteration $l$, the scheme features the following form:

$$r^{l+1} = P_{\Omega}[r^l - \rho^l \times Dir^l] = P_{\Omega}[r^l - \rho^l \times \frac{r^l \times (GC(r^l) - \pi(r^l))}{GC(r^l)}], \qquad (4.27)$$

where $\rho^l \in (0,1)$ is the step size in iteration $l$, $-Dir^l$ is the descent direction, and $\pi(r^l)$ is the vector of least path generalized costs evaluated at $r^l$. $P_{\Omega}[u]$ denotes the unique projection of vector u onto Ω (the set of feasible multi-class path flow vectors *r*) and is defined as the unique solution of the problem: $\min_{v\in\Omega} \| u - v \|$. Based on Eq.(4.27), the new path assignment $r^{l+1}$ is obtained by updating the current path assignment $r^l$ along the descent direction ($-Dir^l$) with a move size $\rho^l$.

Let *p** be the *referenced* least generalized cost path for a $(b, o, d, \tau)$. Specifically, for each $(b, o, d, \tau)$ sub-problem, the multi-class path flow updating scheme in an inner loop iteration *l* is as follows:

$$r_{odp}^{\tau,l+1}(b) = \max\{0, r_{odp}^{\tau,l}(b) - \rho^l \times \frac{r_{odp}^{\tau,l}(b) \times [GC_{odp}^{\tau}(b, r^l) - \pi_{od}^{\tau}(b, r^l)]}{GC_{odp}^{\tau}(b, r^l)}\}$$

$$\forall p \in P(b, o, d, \tau), p \neq p^*; \tag{4.28}$$

$$r_{odp^*}^{\tau,l+1}(b) = r_{odp^*}^{\tau,l}(b) + \sum_{p \in P(b,o,d,\tau), p \neq p^*} \rho^l \times \frac{r_{odp}^{\tau,l}(b) \times [GC_{odp}^{\tau}(b, r^l) - \pi_{od}^{\tau}(b, r^l)]}{GC_{odp}^{\tau}(b, r^l)} \tag{4.29}$$

This path assignment updating scheme implies a natural path flow adjustment mechanism: flows on the non-cheapest paths are moved to the cheapest path and the volume moved out from a non-cheapest path *p* is proportional to $[GC_{odp}^{\tau}(b, r^l) - \pi_{od}^{\tau}(b, r^l)] / GC_{odp}^{\tau}(b, r^l)$, which is intuitively based on the fact that travelers farther from the equilibrium and on paths with larger flow rates are more inclined to change path than those on paths with smaller flow rates and with travel cost closer to the minimal cost.

### 4.6.3 Multi-class dynamic network loading (MDNL) using the traffic simulator

By the BDUE definition, all trips in a network are equilibrated in terms of actual experienced path generalized costs, consisting of experienced path times and path costs, so it is necessary to determine the experienced path generalized costs *G*(*r*) for a given multi-class path flow vector *r*. To this end, the simulation-based dynamic traffic (network loading) model – DYNASMART (Jayakrishnan et al., 1994; Mahmassani, 2001) is employed to evaluate a path assignment *r* and to obtain *GC*(*r*) and time-dependent link travel times used in the path generation step. DYNASMART adopts a hybrid (mesoscopic) approach to capture the dynamics of vehicular traffic flow in the simulation, whereby vehicles are moved individually according to prevailing local speeds, consistent with macroscopic flow relations on links. It should be noted that the algorithm is independent of the specific dynamic traffic model selected; any particle-based (microscopic or mesoscopic) dynamic traffic model capable of capturing complex traffic flow dynamics can be embedded into the proposed algorithm. When a particle-based dynamic traffic model is employed to determine experienced path times, the path time $TT_{odp}^{\tau}(r)$ for a discrete time interval should be considered as the *average* path time of the vehicles with the same $(o,d,\tau,p)$, because, to respect traffic propagation rules and junction exit capacity constraints, different vehicles embarking along path $p\in P(o,d,\tau)$ in departure interval $\tau$ will normally reach their destination *d* at different times and hence experience different trip times. This, in turn, means that the definition of RMDUE (or BDUE) in this study must involve the *average* experienced path generalized cost.

#### 4.6.4 Convergence checking using gap values

Several criteria for convergence checking had been considered in the literature of DTA algorithms. For instance, Peeta and Mahmassani (1995) adopted in their simulation-based DTA model a criterion based on the comparison of path assignments (or path flows) over successive iterations. This study extends the gap-based criterion (or measure) proposed in Chapter 3 for the DUE problem to the RMDUE problem and defines the multi-class version of the gap function as the following:

$$Gap(r^l) = \sum_b \sum_o \sum_d \sum_\tau \sum_{p \in P(b,o,d,\tau)} r_{odp}^{\tau,l}(b) \times [GC_{odp}^{\tau}(b, r^l) - \pi_{od}^{\tau}(b, r^l)] \tag{4.30}$$

Note that, $Gap(r^l)$ provides a measure of the violation of the RMDUE conditions in terms of the difference between the total actual experienced path generalized cost and the total least generalized cost evaluated at any given multi-class path flow pattern *r*. The difference vanishes when the path flow vector *r** satisfies the RMDUE conditions. In the proposed solution algorithm, for practical considerations, if $|Gap(r^l) - Gap(r^{l-1})| \leq \varepsilon$ (a predetermined convergent threshold), convergence is assumed and the program goes back to the outer loop (step 2).

### 4.6.5 Vehicle-based implementation technique

The above BDUE model and algorithm are featured as the path-based approach, necessitating the explicit storage of the path set and path assignment results for each $(b, o, d, \tau)$. Although it is straightforward to record all the paths and the corresponding path choice probabilities for each $(b, o, d, \tau)$ by using multi-dimensional arrays, computer memory requirements grow dramatically when the number of OD pairs is large, or many iterations are required to achieve convergence. Furthermore, the relaxation to the continuously distributed VOT allows a large number of classes of trips to be in a simultaneous equilibrium, each of which requires its own set of paths, and the number of user classes is unknown a priori and changes from iteration to iteration, making it more difficult to construct a memory efficient data structure for storing and updating the huge path set and path assignments in large-scale network applications. Essentially, as an attempt to accommodate greater behavioral and policy realism in applying DTA models for designing and evaluating dynamic pricing schemes, modeling heterogeneous users with a range of VOT as opposed to identical users exacerbates the computational complexity and memory requirement.

In a particle-based and simulation-based DTA system, vehicles carry their paths from iteration to iteration, and the vehicle path set implicitly reflects and stores the path set and path assignments results. This is particularly advantageous for large-scale DTA applications, as the total number of feasible paths generated by the iterative solution algorithm, after a certain number of iterations, could be significantly greater than the total number of vehicles, which is determined a priori by the OD demand table. For example, in the Portland transportation planning network (Nagel et al., 2000), there are about 1,260 traffic analysis zones (TAZ) and 1.5 million OD pairs, and the total trips are 1.5 millions. Obviously, every OD pair requires more than one time-dependent least generalized cost path for reaching the BDUE. Thus, storing the vehicle path set is more memory-efficient than storing the complete path set and routing policies for large-scale networks.

With this vehicle-based implementation technique, the path assignment updating scheme presented in Eq.(4.28) and Eq.(4.29) can be interpreted as the following. In iteration $l$, for each $(b,o,d,\tau)$ and for each path $p \in P(b,o,d,\tau)$, the number of vehicles moved to the least generalized cost path is $\rho^l \times \frac{r_{odp}^{\tau,l}(b) \times [GC_{odp}^{\tau}(b,r^l) - \pi_{od}^{\tau}(b,r^l)]}{GC_{odp}^{\tau}(b,r^l)}$; and the remaining vehicles would keep their current paths. Essentially, this implementation technique uses the vehicle path set as a proxy for the exact path set and path assignment results (routing policies), which can be approximately recovered from the realized vehicle paths in the last iteration's simulation results.

## 4.7 NUMERICAL EXPERIMENTS

### 4.7.1 Experiments for examining the PAM

A set of numerical experiments is conducted to examine the PAM. For convenience, and with no loss of generality, the TDLCP algorithm embedded in the PAM is implemented as a backward procedure (i.e. rooted at the destination node, from all nodes to one node). They are coded and compiled in Microsoft Visual C++ 6.0 on Windows XP platform and evaluated on a machine with an INTEL PENTIUMN III 2.0GHz CPU and 2 GB memory.

To validate solutions found by the PAM, the grid network and toll scenario (Figure 4.2) created in Dial's bi-criterion traffic assignment work (1997) are used. As shown in the figure, the grid network has 9 nodes, 12 (2-way) links, and one OD pair (1->9). The numbers next to each link are the travel time and travel cost (if any) of that link. There are four links in the network with \$1 or \$2 dollars of toll. The feasible VOT range is from \$0 to \$1 per minute. Since the toll scenario is static, to apply the PAM, the length of time interval is set equal to the planning horizon (i.e. there is only one time interval). The solution found by the PAM is identical to that given in Dial's paper (1997). As depicted in Figure 4.3, there are 5 breakpoints (1.0, 0.416, 0.208, 0.166, and 0.0) that partition the entire feasible range of VOT to 4 VOT sub-intervals, and hence there are 4 different least generalized cost path trees contained in the solution. It can also be observed that the tree (a) corresponding to a range of higher VOT values involves more toll links, while the tree (d) corresponding to a range of lower VOT values includes fewer toll links. Reflected in the path choice model of a traffic assignment model, this would intuitively have high VOT trips using more toll links to save time and low VOT trips avoid toll links to save money.

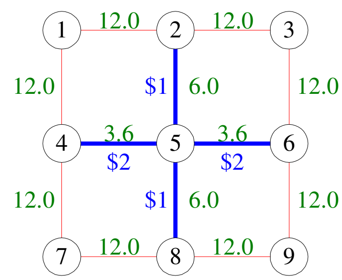



Figure 4.2 The grid network and toll scenario from Dial (1997)

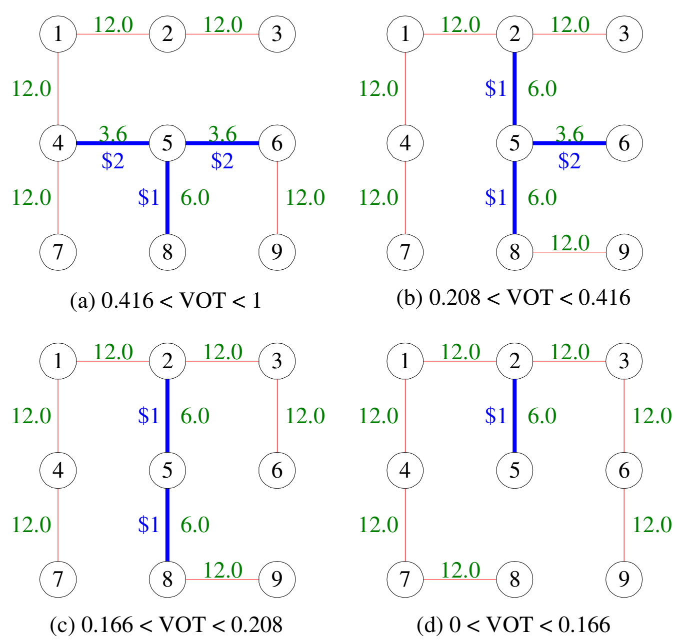



Figure 4.3 The solution of the example from Dial (1997)

To test the computational performance of the PAM with respect to several problem size attributes, the remaining experiments are conducted on three real road networks. The sizes (in terms of number of nodes and number of links) of the three real networks used in the experiments are as follows.

Fort Worth (FW), Texas: 180 nodes and 445 links
Irvine, California: 326 nodes and 626 links
Knoxville, Tennessee: 1347 nodes and 3004 links

The planning horizon is set to 120 minutes, and the time-dependent travel times are obtained from the 2-hour simulation results output from the traffic simulator DYNASMART (Jayakrishnan et al.,1994a) for each of the three networks.

The first experiment is to explore the impact of introducing the time dimension to the static bi-criterion shortest path (BSP) problem on the size of the solution set and associated computational effort; these issues are critical for developing on-line and off-line DTA models. Of particular interest is the relationship between the number of breakpoints (in the VOT range over which the TDLCP tree remains Pareto-optimal) and the number of time intervals into which the planning horizon is discretized. The number of breakpoints is selected as a figure of merit because it can serve as a surrogate for not only computational time but also size of the solution set. The length of a time interval is varied from 1 to 120. Time-dependent travel costs are randomly generated between $0.01 and $2 for every 30 minutes. The feasible range of VOT is set between $0.01 and $10.0 per minute. It is also assumed that travelers do not perceive differences in VOT below $0.01 per minute, implicitly setting the maximal number of breakpoints to 1000. To study the impact of using different root nodes for the constructed trees, 10 different destination nodes are randomly selected from the Knoxville network. The results show that the number of breakpoints varies only slightly (less than 5%) for different destination nodes.

Therefore, each data point in the following experiments reports the average value of 10 realizations, each of which is solved for a randomly selected destination.

For the three networks, Figure 4.4 shows the relationship between number of time intervals and number of breakpoints, and Figure 4.5 shows the relationship between number of time intervals and execution time. The experiment results show that the number of breakpoints is monotonically non-decreasing as the length of the aggregation time interval decreases. For example, the average number of breakpoints in the Knoxville network increases from 159.2 to 928.3 when the length of an aggregation time interval decreases from 120 minutes to 1 minute. As expected, for the same size of time intervals, a larger network has more breakpoints than a smaller network. For example, with a 1-minute time interval, the average number of breakpoints in the Knoxville network is 1.91 times that in the FW network. As shown in Figure 4.5, the computational times for the three networks, and especially for Knoxville, increase with the number of discrete time steps. For example, computing one TDLCP three takes an average of 9.15 minutes with a 1-minute aggregation time interval (120 time steps), compared to 1.02 *seconds* with a single 2-hour time interval.

The last experiment with the PAM aims to represent variable congestion pricing schemes more realistically by applying travel costs (road tolls) on only a given percentage of freeway and highway links, instead of imposing costs on all the links. The Knoxville network is used for these experiments, with an aggregation time interval of 5 minutes (24 time steps), and different travel costs generated for every 30-minute period. As shown in Figure 4.6, the higher the toll link coverage, the more breakpoints in the complete solution set. In addition, even with only 10% of the freeway and highway links (around 82 links) selected as toll links (i.e. with nonzero travel cost), the corresponding solution set is still considerably large (494.2 breakpoints, equivalent to 73% of the solution set size for 100% toll link coverage).

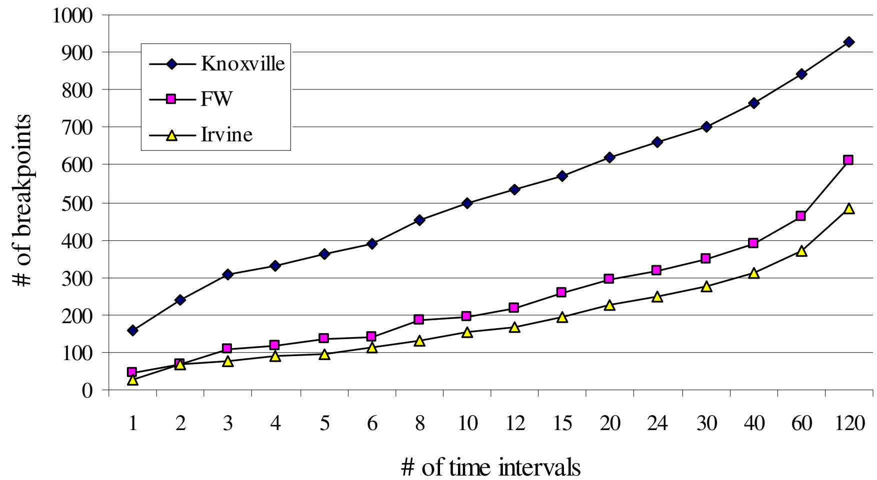



Figure 4.4 Relationship between number of time intervals and number of breakpoints

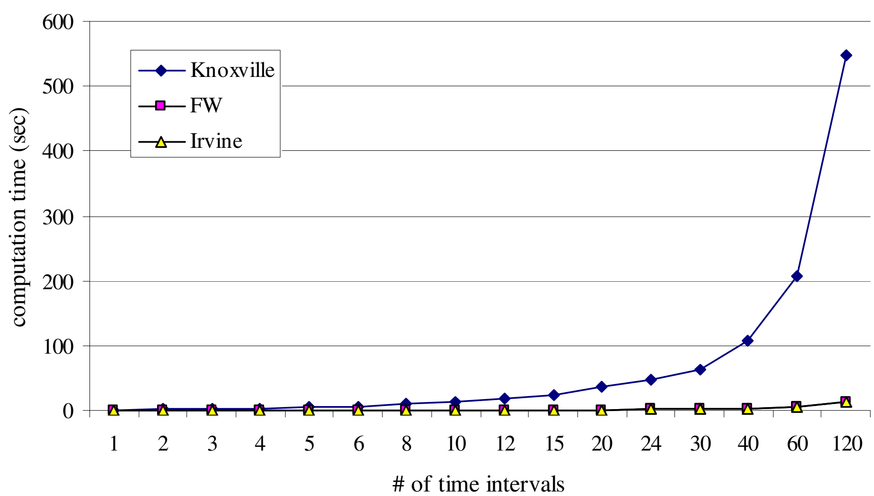



Figure 4.5 Relationship between number of time intervals and computational time

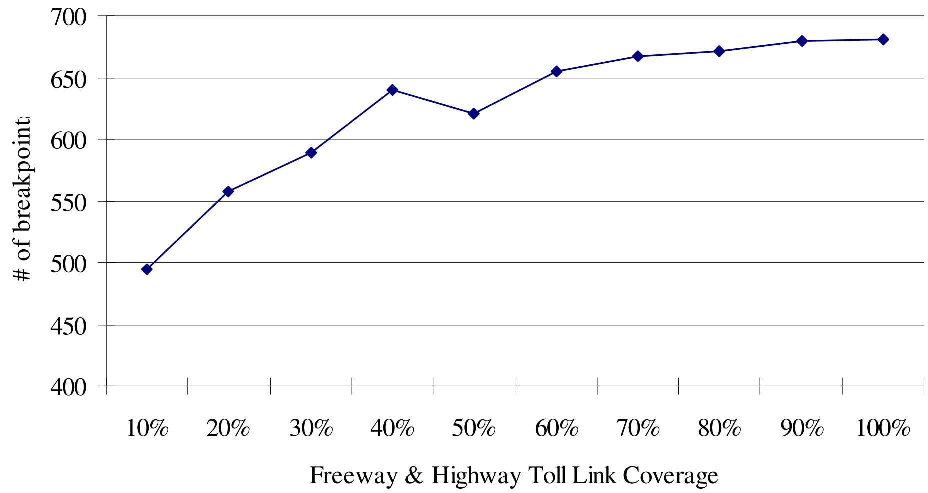



Figure 4.6 Impact of freeway and highway toll link coverage

### 4.7.2 Examine the algorithmic convergence and the solution quality

A set of numerical experiments is conducted to examine the BDUE algorithm. In addition to the algorithmic convergence property, with the explicit consideration of user heterogeneity, of particular interest is how the VOT distribution affects path flow patterns and toll road usage under dynamic toll pricing scenarios. The proposed BDUE algorithm is implemented using the aforementioned vehicle-based technique, which can be seamlessly integrated with any mesoscopic/microscopic traffic simulator and is considered particularly appealing for large network deployments of DTA models. The algorithm is coded and compiled by using the Compaq Visual FORTRAN 6.6 and evaluated on the Windows XP platform and a machine with an Intel Pentium IV 2.8 GHz CPU and 2GB RAM.

In all the experiments conducted, the following parameter settings are applied. The continuous VOT distribution considered in the experiments is a normal distribution with (mean, standard deviation) = (24, 12), denoted as N(24, 12). The parameters of this normal distribution are adapted from the estimated measurements in a value pricing experiment conducted in Southern California, USA (e.g. Lam and Small, 2001; Brownstone and Small, 2005), and the unit of VOT in this study is United States dollars (USD) per hour. The feasible range of the VOT distribution $[\alpha^{\min}, \alpha^{\max}]$ is [0.6, 180]. The resolution (aggregation interval) of the time-dependent shortest path tree calculation is set to 6-second, which is the same as the time step for the simulation. The OD demand assignment interval (or departure time interval) is set to 1 minute. A strict convergence criterion is used in the inner loop of the column generation-based algorithm; that is $|Gap(r^l) - Gap(r^{l-1})|/Gap(r^l) \leq 0.001$. The initial solutions of the experiments are obtained by loading time-varying OD demands to the (static) extreme efficient paths calculated based on prevailing travel times output from the traffic simulator.

Another measure of effectiveness (MOE) is collected in the conducted experiments, in addition to the value of $Gap(r)$. It is the average gap over all vehicles in the network for a given path flow pattern $r$.

$$AGap(r) = \frac{\sum_b \sum_o \sum_d \sum_\tau \sum_{p \in P(b,o,d,\tau)} r_{odp}^{\tau}(b) \times [GC_{odp}^{\tau}(b,r) - \pi_{od}^{\tau}(b,r)]}{\sum_b \sum_o \sum_d \sum_\tau \sum_{p \in P(b,o,d,\tau)} r_{odp}^{\tau}(b)} \tag{4.31}$$

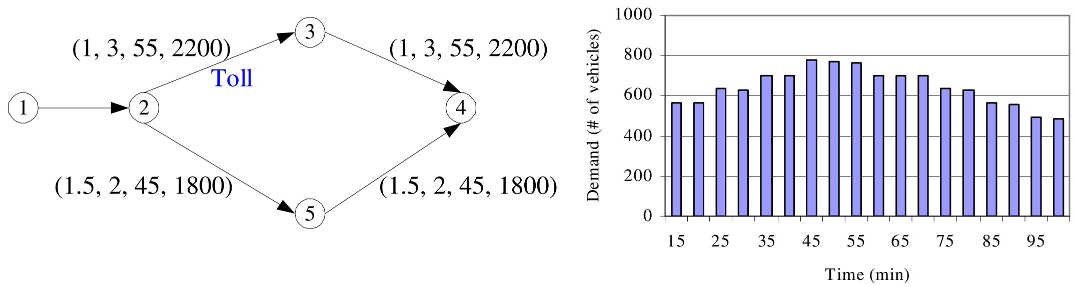
This MOE is independent of problem size and thus useful for examining the convergence pattern and solution quality of the BDUE algorithm on different networks. The minimum of the $AGap(r)$ is zero. Essentially, the smaller the average gap, the closer the solution is to the BDUE. Note that this study aims at developing a bi-criterion DTA model for evaluating dynamic pricing scenarios but not solving for a toll vector that improves local or network-wide performance. Hence, testing different dynamic toll vectors in the conducted experiments does not intend to compare their effectiveness on reducing congestion, and focuses exclusively on demonstrating what the BDUE model can accomplish and why the user heterogeneity should be addressed in evaluating dynamic road pricing scenarios.

The first set of experiments aims to examine the convergence pattern and the solution quality of the proposed BDUE algorithm in terms of $Gap(r)$ or $AGap(r)$.

4.7.2.1 The experiment on a small network

This experiment is conducted on a small test network (Figure 4.7(a)), consisting of 5 nodes and 5 links. Each link is divided into many segments, each of which has the length equal to the distance traveled by free-flowing traffic in one simulation interval. Associated with each link are the following attributes: length (miles), number of lanes, free flow speed (miles per hour), and capacity (vehicles per hour per lane). There are two paths connecting the only one OD pair (1, 4): 1→2→3→4 and 1→2→5→4. A two-hour time-varying OD demand table is loaded and there are about 11,500 vehicles loaded in the observation period (10-100 minutes), in which summary statistics are collected (Figure 4.7(b)). Note that in this experiment the outer loop (i.e. path-finding step) of the BDUE algorithm is not activated, because the only two paths have already included in the initial solution. A toll booth is installed on the entry of link (2→3) so the vehicles choosing path (1→2→3→4) have to pay (time-varying) tolls. The time-dependent (or step) pricing scenario applied on this small test network is as the following.

| Time period: | 10-30 min | 30-50 min | 50-70 min | 70-100 min |
|---|---|---|---|---|
| Toll: | \$0.20 | \$0.40 | \$0.60 | \$0.30 |




(a) (b)

Figure 4.7 The small test network and time-varying OD demand

Table 4.1 Convergence patterns of the BDUE algorithm on the test network

| | Constant VOT = 24 | | Normal VOT: N(24,12) | |
|---|---|---|---|---|
| Iteration | *Gap*(*r*) | *AGap*(*r*) | *Gap*(*r*) | *AGap*(*r*) |
| 1 | 4531.1 | 0.396 | 3346.9 | 0.292 |
| 2 | 1407.4 | 0.123 | 1168.7 | 0.102 |
| 3 | 686.8 | 0.060 | 690.7 | 0.060 |
| 4 | 501.3 | 0.044 | 421.8 | 0.037 |
| 5 | 369.0 | 0.032 | 263.1 | 0.023 |
| 6 | 306.6 | 0.027 | 187.1 | 0.016 |
| 7 | 262.2 | 0.023 | 149.2 | 0.013 |
| 8 | 200.8 | 0.018 | 109.9 | 0.010 |
| 9 | 188.3 | 0.016 | 165.6 | 0.014 |
| 10 | 184.1 | 0.016 | 139.7 | 0.012 |
| 11 | 166.7 | 0.015 | 85.3 | 0.007 |
| 12 | 152.1 | 0.013 | 73.4 | 0.006 |
| 13 | 120.1 | 0.010 | 84.7 | 0.007 |
| 14 | 111.1 | 0.010 | 81.3 | 0.007 |
| 15 | 105.6 | 0.009 | 111.5 | 0.010 |
| 16 | 95.4 | 0.008 | 63.6 | 0.006 |
| 17 | 88.9 | 0.008 | 67.9 | 0.006 |
| 18 | | | 59.7 | 0.005 |

The convergence pattern (in terms of *Gap*(*r*) and *AGap*(*r*)) of the BDUE algorithm on the small network with the time-dependent pricing scenario listed above are shown in Table 4.1. Two different VOT assumptions: constant VOT (=24) and normal distribution VOT (N(24, 12)) are considered in this experiment, in order to study the impact of the VOT distribution on the convergence pattern of the solution algorithm. It can be seen that the proposed algorithm behaves similarly (in the convergence pattern) under the two different VOT assumptions and is able to find close-to-BDUE solutions in both cases, as the final average gap values are all fairly small (less than 0.01 minutes).

4.7.2.2 The experiment on the Irvine network

The Irvine (California, USA) network depicted in Figure 4.8 consists of 326 nodes (70 of them are signalized), 626 links, and 61 traffic analysis zones (TAZ) and had been calibrated by using real-world observations from multiple-day detector data (Mahmassani et al. 2003). A 2-hour (7-9AM) morning peak time-varying OD demand table is extracted from a 6-hour (4-10AM) demand table and loaded to the test network, with 35,300 vehicles in the observation period (7:10-8:50AM). To create hypothetic dynamic road pricing scenarios, one lane of a portion (about 1 mile) of the I-405 westbound freeway is converted to the toll road, along with an additional new toll lane. The two toll lanes have the same length as the (remaining) three regular lanes but a 10-mile higher posted speed limit (and hence higher capacity) than the regular lanes. Table 4.2 lists the three simple dynamic pricing scenarios tested in the experiment conducted on the Irvine network. These three pricing scenarios have the same four pricing periods but different toll levels, each representing low, middle, and high toll scenarios, respectively.

Table 4.2 Dynamic road pricing scenarios tested on the Irvine network

| Pricing Scenario | Period 1 (7:00-7:30AM) | Period 2 (7:30-8:00AM) | Period 3 (8:00-8:30AM) | Period 4 (8:30-9:00AM) |
|---|---|---|---|---|
| 1 (Low) | $0.10 | $0.20 | $0.30 | $0.15 |
| 2 (Middle) | $0.20 | $0.30 | $0.40 | $0.25 |
| 3 (High | $0.30 | $0.40 | $0.50 | $0.35 |


Figure 4.8 Irvine network with hypothetic toll road

The convergence patterns in terms of iteration-by-iteration gap values of the BDUE algorithm under the three dynamic pricing scenarios are presented in Table 4.3. It can be found that the algorithm can effectively reduce the gap measure (as well as the average gap defined in Eq.(4.17) in all three pricing scenarios tested on the Irvine network, although the convergence patterns are not strictly monotonic decreasing. As for the solution quality, the final gap values obtained by the BDUE algorithm are 3.9% (196.3/5028.6), 4.5% (234.9/5211.2), and 5.4% (315.1/5795.7) of the initial gap values, respectively, for the three pricing scenarios. In addition, the average gap values for the three pricing scenarios, obtained by dividing these final gap values by the number of vehicles loaded in the observation period, are all less than 0.01 minutes. These small gap and average gap values indicate that the BDUE algorithm is able to find close-to-BDUE solutions for this network.

Table 4.3 Convergence patterns of the BDUE algorithm on the Irvine network

| | Gap($r$) | | | AGap($r$) | | |
|---|---|---|---|---|---|---|
| Iteration | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 1 | Scenario 2 | Scenario 3 |
| 0 | 5028.6 | 5211.2 | 5795.7 | 0.142 | 0.148 | 0.164 |
| 1 | 835.0 | 1025.6 | 851.3 | 0.024 | 0.029 | 0.024 |
| 2 | 787.1 | 892.2 | 822.7 | 0.022 | 0.025 | 0.023 |
| 3 | 452.8 | 624.6 | 546.9 | 0.013 | 0.018 | 0.015 |
| 4 | 536.9 | 505.0 | 501.4 | 0.015 | 0.014 | 0.014 |
| 5 | 590.7 | 597.7 | 407.3 | 0.017 | 0.017 | 0.012 |
| 6 | 376.1 | 415.4 | 542.2 | 0.011 | 0.012 | 0.015 |
| 7 | 409.6 | 332.2 | 419.5 | 0.012 | 0.009 | 0.012 |
| 8 | 523.4 | 342.0 | 385.8 | 0.015 | 0.010 | 0.011 |
| 9 | 316.2 | 369.4 | 366.9 | 0.009 | 0.010 | 0.010 |
| 10 | 406.5 | 357.9 | 299.1 | 0.012 | 0.010 | 0.008 |
| 11 | 372.9 | 280.1 | 460.6 | 0.011 | 0.008 | 0.013 |
| 12 | 430.7 | 294.8 | 402.2 | 0.012 | 0.008 | 0.011 |
| 13 | 335.7 | 238.9 | 237.7 | 0.010 | 0.007 | 0.007 |
| 14 | 589.1 | 256.4 | 292.6 | 0.017 | 0.007 | 0.008 |
| 15 | 274.5 | 255.4 | 320.2 | 0.008 | 0.007 | 0.009 |
| 16 | 283.4 | 252.9 | 353.9 | 0.008 | 0.007 | 0.010 |
| 17 | 271.2 | 228.3 | 249.3 | 0.008 | 0.006 | 0.007 |
| 18 | 247.1 | 268.3 | 323.7 | 0.007 | 0.008 | 0.009 |
| 19 | 258.4 | 285.3 | 313.0 | 0.007 | 0.008 | 0.009 |
| 20 | 196.3 | 234.9 | 315.1 | 0.006 | 0.007 | 0.009 |

To highlight the memory efficiency of the vehicle-based implementation technique, a grand path set version of the BDUE algorithm is also implemented by using fixed size multi-dimensional arrays to store the complete extreme efficient path set and routing policies for all iterations. With identical experimental settings, the grand path set version is found to require more than 2.83GB memory (the largest memory size available for a single 32-bit Windows application is 3.0GB), while the vehicle path set version needs about 2.14GB memory. Note that although some advanced data structures might be applied to reduce the memory usage of the grand path set version, this difference in memory usage is still proportional to the problem (or network) size and the number of iterations required to reach the convergence.

4.7.2.3 The experiment on the CHART network

To further demonstrate the capability of the BDUE algorithm for large-scale networks with dynamic road pricing scenarios, the next experiment is conducted on a recently coded large road network, the CHART network, which consists primarily of the I-95 freeway corridor between Washington, D.C. and Baltimore (Maryland, USA) and is bounded by two beltways (I-695 Baltimore Beltway on the north and I-495 Capital Beltway on the south). The CHART network has 2241 nodes (231 of them are signalized), 3459 links and 111 traffic analysis zones (TAZ), and been calibrated by using real-world observations from multiple-day detector data (Mahmassani et al. 2005a). An available 1-hour (7:30-8:30AM) morning peak time-varying OD demand (with 39,560 vehicles in the observation period from 7:40 to 8:20 AM) table is extracted and loaded to the network. To create hypothetic dynamic toll scenarios, one of the 20-mile long southbound lanes of the I-95 corridor is converted to the toll road, together with an additional new toll lane. The two toll lanes have the same length, posted speed limit, and capacity as the (remaining) three regular lanes. The two-lane toll road consists of 57 links in the coded network, and the four access/egress points to/from the toll road are interchanges with I-195, MD-100, MD-32 and MD-198, where additional on-ramps and off-ramps are added. A dynamic link toll vector generated by the method proposed by Dong et al. (2006) is used in this experiment to test the BDUE algorithm. Their method solves for a vector of time-varying link tolls so as to maintain high level of service on the toll road. Essentially, the deviations between (prevailing or predicted) link concentrations and a given set of target concentrations on toll links are calculated, and then link tolls are determined by some control regulator according to the deviations.

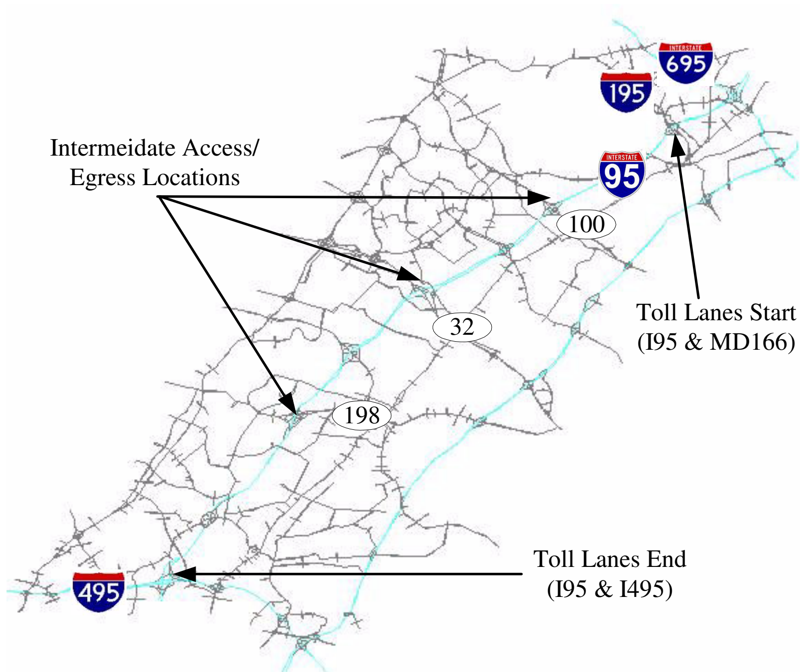



Figure 4.9 Baltimore-Washington D.C. corridor network with hypothetic toll road

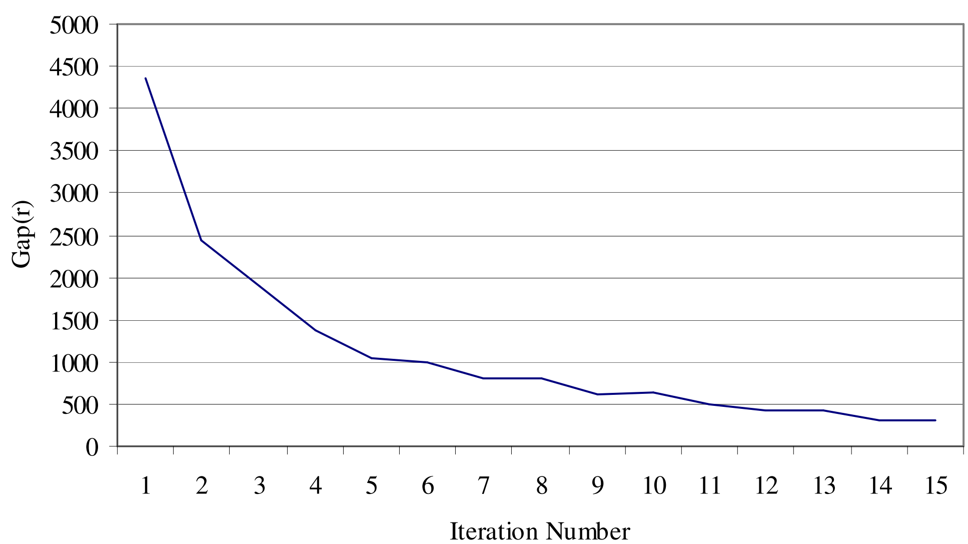



Figure 4.10 The convergence pattern of the algorithm on the BW network

The convergence pattern in terms of the gap measure of the algorithm is plotted in Figure 4.10. The initial and final gap values are 4365.4 and 300.1, respectively. The gap reduction is 93.1% (4365.4−300.1)/4365.4. The final average gap value is 0.008

(300.1/39560) minutes, which demonstrates the algorithm can find the close-to-BDUE solution in this case. The memory usage is 2.95GB and the computation time of finishing 22 iterations is about 28 hours. The grand path set version of the algorithm fails in this experiment on the 32-bit operation system as it requires more than 3.0GB memory. This experiment further illustrates the contribution of introducing the vehicle-based implementation technique in developing large-scale DTA network models for evaluating dynamic road pricing scenarios.

### 4.7.3 The impacts of VOT assumption on toll road usage

This set of experiments intends to investigate the impacts of user heterogeneity in terms of VOT on path flow patterns and toll road usage in evaluating different dynamic road pricing scenarios.

4.7.3.1 The experiment of evaluating a new express toll road

In this experiment, the scenario in which a new express toll road is constructed and operated in a small network is considered, with particular interest in investigating the impact of user heterogeneity in terms of different VOT distributions on the path flow pattern and toll road usage before and after the toll road is constructed. The original network consists of 5 nodes and 5 links (Figure 4.11(a)). Each link is divided into many segments, each of which has the length equal to the distance traveled by free-flowing traffic in one simulation interval. Associated with each link are the following attributes: length (miles), number of lanes, free flow speed (miles per hour), and capacity (vehicles per hour per lane). There are two paths connecting the only one OD pair (1, 4): (path 1) 1→2→3→4 and (path 2) 1→2→5→4. A two-hour peak period time-varying OD demand table is loaded and there are about 9,900 vehicles loaded in the observation period (10-

100 minutes). There are not any tolls collected in this network. Solving the BDUE problem on this network (Figure 4.11(a)) gives the following path flow pattern.

Path 1: path share - 58%; travel time - 4.68 minutes (free flow time 2.4 min)

Path 2: path share - 42%; travel time - 4.88 minutes (free flow time 3.6 min)

In order to alleviate the congestion in the peak hours, the local traffic management authority decides to construct an express road that connects directly node 2 and node 4 (i.e. 2→6→4) and install a toll booth in the entry of the express road. The configuration of the express road and its link attributes are provided in Figure 4.11(b), and the path (path 3) is defined as 1→2→6→4. The time-dependent (or step) pricing scenarios applied on this small test network are listed in Table 4.4.

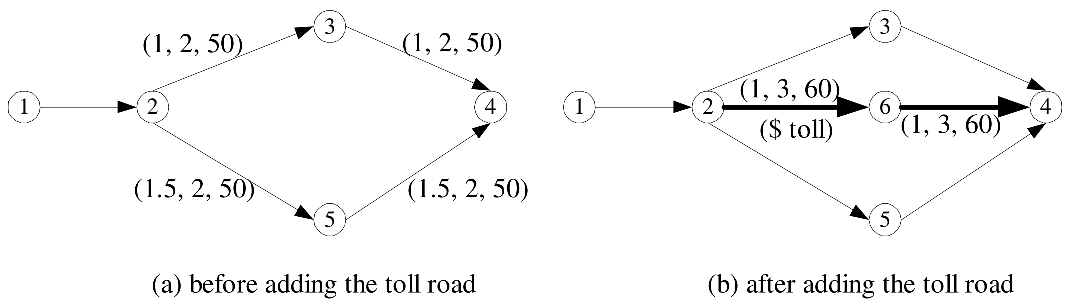



Figure 4.11 Before and after adding the toll road in a small test network

Table 4.4 Dynamic pricing scenarios tested after constructing the express road

| Pricing Scenario | Period 1 (0-30 min) | Period 2 (30-50 min) | Period 3 (50-70 min) | Period 4 (70-120 min) |
|---|---|---|---|---|
| 1 | $0.01 | $0.10 | $0.05 | $0.01 |
| 2 | $0.05 | $0.25 | $0.15 | $0.05 |
| 3 | $0.05 | $0.50 | $0.30 | $0.05 |
| 4 | $0.05 | $0.75 | $0.45 | $0.05 |
| 5 | $0.05 | $1.00 | $0.60 | $0.05 |

The convergence patterns in terms of $AGap(r)$ of applying the proposed algorithm for solving the BDUE problem on the new network (Figure 4.11(b)) under pricing scenario 1 are presented in Table 4.5. There are two VOT assumptions tested in the experiment: constant VOT = 24 and normal distribution VOT = N(24, 12). As shown in the table, the convergence patterns under the two VOT assumptions are pretty similar. Moreover, the algorithm is able to find the close-to-BDUE solutions as it can effectively reduce the $AGap(r)$ from 1.52 to 0.1 minutes in both cases.

Table 4.5 Convergence patterns of the algorithm after constructing the express road

| | Constant VOT | Normal VOT |
|---|---|---|
| Iteration | $AGap(r)$ | $AGap(r)$ |
| 0 | 1.528 | 1.522 |
| 1 | 0.832 | 0.829 |
| 2 | 0.575 | 0.571 |
| 3 | 0.411 | 0.413 |
| 4 | 0.302 | 0.321 |
| 5 | 0.226 | 0.253 |
| 6 | 0.187 | 0.198 |
| 7 | 0.163 | 0.168 |
| 8 | 0.149 | 0.143 |
| 9 | 0.141 | 0.127 |
| 10 | 0.134 | 0.117 |
| 11 | 0.131 | 0.113 |
| 12 | 0.129 | 0.112 |
| 13 | 0.125 | 0.111 |
| 14 | 0.125 | 0.107 |
| 15 | 0.118 | 0.104 |

Table 4.6 Path shares (%) under different toll scenarios

| VOT | Constant VOT | | | Normal VOT | | |
|---|---|---|---|---|---|---|
| Pricing # | Path 1 | Path 2 | Path 3 | Path 1 | Path 2 | Path 3 |
| 1 | 13 | 1 | 86 | 16 | 3 | 81 |
| 2 | 26 | 2 | 72 | 29 | 3 | 68 |
| 3 | 33 | 7 | 60 | 33 | 4 | 63 |
| 4 | 37 | 13 | 50 | 36 | 8 | 56 |
| 5 | 37 | 20 | 43 | 38 | 14 | 48 |

The resulting path flow patterns, under different pricing scenarios and for different VOT assumptions, in terms of path share (percentages of OD demand using a path) are reported in Table 4.6. Note that in this network the toll road usage can be obtained as the path share of path 3. The toll road usages under different pricing scenarios and with different VOT assumptions are plotted in Figure 4.12. As demonstrated in this figure, when the toll level is increased from scenario 1 to scenario 5, the decrease of the toll road usage (i.e. the path share of path 3) in the normal distribution VOT case is less dramatic than that for in the constant VOT case. Furthermore, it is worth noting that when the toll charge is low (scenario 1), the toll road usage predicted by the DTA model with a single constant VOT is higher than that forecasted by the BDUE model with the normal VOT distribution. Since the single VOT model assumes homogeneous users, all users with this constant VOT are willing to use the toll road when the toll charge is low. However, there are in fact a certain number of trips that have lower VOT and may not want to use the toll road even when the toll charge is not high. This phenomenon can be captured in the proposed BDUE model with continuous VOT by recognizing the existence of those low VOT users in the heterogeneous population. On the other hand, when the toll charge is high (scenario 5), the constant VOT model gives lower toll road usage than the continuous VOT model, because, in this case, it assumes that all users behave identically in response to the higher toll charge so travelers are less likely to use the toll road to save time. The BDUE model acknowledges the fact that there is a certain portion of high VOT trips that still wish to take the expensive but fast toll road.

If the results obtained by the normal distribution VOT model are considered as the benchmark, then the constant VOT model overestimates the toll road usage when the toll charge is low and underestimates the toll road usage when the toll charge is high. The experimental results also provide toll operators useful information: when the toll level changes, users' reactions are not as dramatic as what had been predicted by DTA models with the single constant VOT assumption.

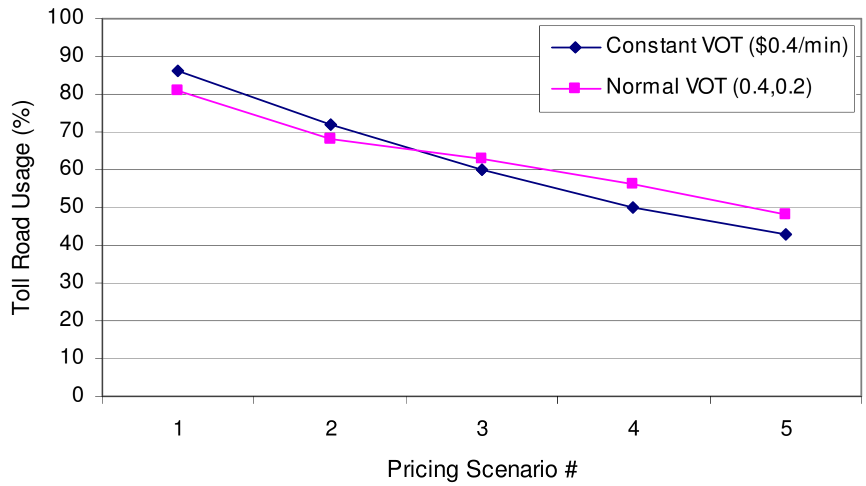



Figure 4.12 Toll road usages under different pricing scenarios

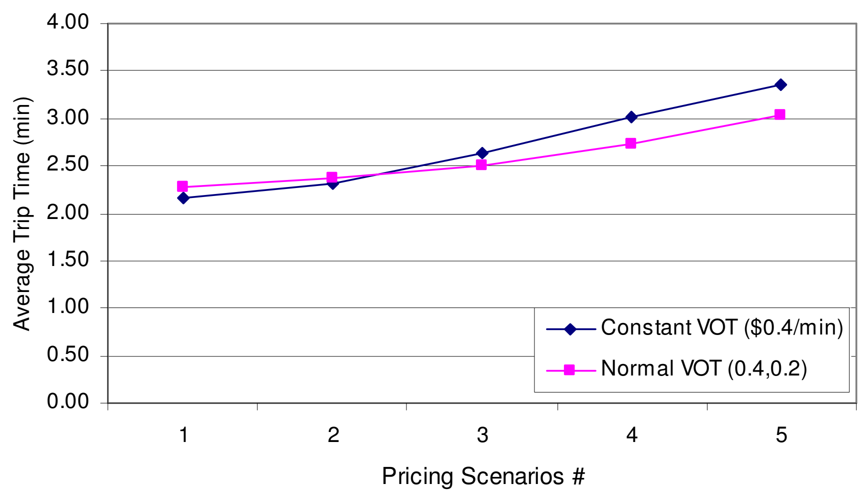



Figure 4.13 Average trip times under different pricing scenarios

The impact of user heterogeneity in terms of VOT on toll road usage is also reflected in the overall network performance. Figure 4.13 presents the network-wide average trip times under different pricing scenarios. When the toll charge is high, since the number of trips assigned to the toll road (expressway) is underestimated in the constant VOT case and more trips take the low-speed local streets, the resulting average trip time is higher than that of the BDUE model with continuous VOT distribution. Thus, the estimate/prediction of network performance, under a given pricing scenario, obtained from the constant VOT model could be biased if user heterogeneity is not realistically captured.

4.7.3.2 The experiment on the Fort Worth network

The Fort Worth (Texas, USA) network, depicted in Figure 4.14 and consisting of 180 nodes (62 of them are signalized), 445 links and 13 traffic analysis zones (TAZ), is used in this experiment. An available one-hour time-varying OD demand (23,000 vehicles) table is loaded to the network. The planning horizon is 90 minutes while the statistics are collected only from 10 to 50 minutes in order to take into account the time for simulation warm-up and network clearance. In this experiment, a portion (about 1.5 miles) of the entire I-35 northbound freeway corridor (4 lanes) is converted to the toll road, to create the hypothetical pricing scenario. Table 4.7 lists the four simple dynamic pricing scenarios tested in this set of experiments. In addition to the continuous N(24,12) VOT distribution, two other VOT assumptions are considered as well: one is a constant VOT equal to $24/hour, and the other one is a discrete VOT distribution in which the entire population is segmented into three groups according to different trip purposes (with mean VOT = $24/hour).

Group 1: commute trips, 50%, VOT = $24/hour
Group 2: business trips, 25%, VOT = $36/hour
Group 3: other trips, 25%, VOT = $12/hour

In this network, the toll road usage is obtained as the percentages of vehicles from a given (major) OD pair passing through toll links, and it is used to explore the impact of VOT distributions on network flow patterns under different dynamic pricing scenarios.

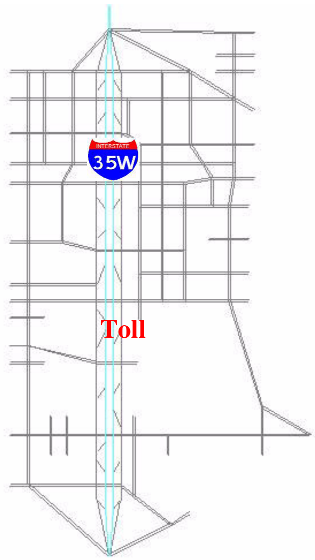



Figure 4.14 Fort Worth network with the converted toll road

Table 4.7 Dynamic road pricing scenarios tested on the Fort Worth network

| Pricing Scenario | Period 1 (0-30 minutes) | Period 2 (30-60 minutes) |
|---|---|---|
| 1 | $0.10 | $0.15 |
| 2 | $0.30 | $0.50 |
| 3 | $0.75 | $1.00 |
| 4 | $1.00 | $1.50 |

Figure 4.15 provides the toll road usage over the planning horizon of one major OD pair using the northbound of the freeway corridor predicted by the BDUE model with different VOT distributions and under different pricing scenarios. The experimental result is similar to that on the small network with the express toll road. When the toll level is increased from scenario 1 to scenario 4, the decrease of the toll road usage in the constant and discrete VOT models is more dramatic than that in the normal distribution VOT model. Besides, when the toll charge is low (scenario 1), the toll road usage predicted by the DTA model with a single constant VOT is higher than that forecasted by the BDUE model with continuous or discrete VOT distributions. On the other hand, when the toll charge is high (scenario 4), the constant VOT model gives lower toll road usage than the

continuous or discrete VOT model, because, in this case, it assumes that all users behave identically in response to the higher toll charge so travelers are less likely to use the toll road to save time. If the results obtained by the normal distribution VOT model are considered as the benchmark, then both constant VOT and discrete VOT models overestimate the toll road usage when the toll charge is low and underestimate the toll road usage when the toll charge is high.

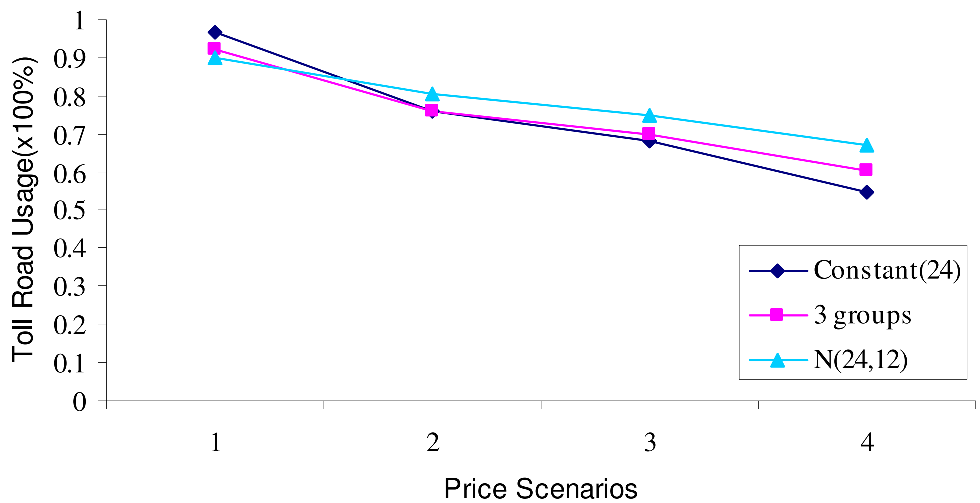



Figure 4.15 Toll road usages under different pricing scenarios

The impact of user heterogeneity in terms of VOT on toll road usage is also reflected in the overall network performance. Figure 4.16 presents the network-wide average trip times under different pricing scenarios. When the toll charge is high, since the number of trips assigned to the toll road (freeway) is underestimated in the constant VOT and discrete VOT cases and more trips take the low-speed local streets, the resulting average trip time is higher than that of the BDUE model with continuous VOT distribution. Thus, the estimate/prediction of network performance, under a given pricing scenario, obtained from the constant VOT model or discrete VOT model could be biased if user heterogeneity is not realistically captured.

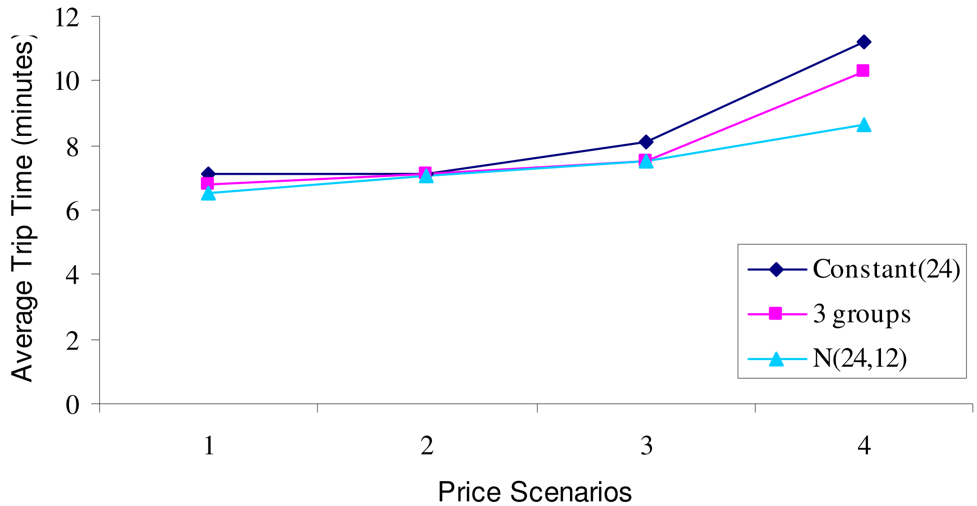



Figure 4.16 Average network trip times under different pricing scenarios

## 4.8 Summary

With increasing interest in applying dynamic road pricing strategies to alleviate peak period congestion and improve network performance, there is a need to develop an equilibrium network assignment model capable of capturing traffic dynamics and heterogeneous users' responses to time varying toll charges for the design and evaluation of time-dependent pricing schemes. This chapter proposes the bi-criterion dynamic user equilibrium (BDUE) traffic assignment model, in which the VOT is considered as a continuously distributed random variable across the population of trips, and presents its solution algorithm. The BDUE problem is formulated as an infinite dimensional VI, and solved by the column generation-based algorithmic framework presented in section 4.4. To circumvent the difficulty of storing the memory-intensive path set and routing policies for large-scale network applications, the vehicle-based implementation technique, using the vehicle path set as a proxy for keeping track of the path assignment results, is applied. Although the mathematical abstraction of the problem is a typical analytical formulation, this study adopts the simulation-based approach to tackle many practical aspects of the DTA applications.

The experimental results show that the convergence pattern of the proposed BDUE algorithm is not affected by the different VOT assumptions (i.e. constant or random VOT), and it is able to find close-to-BDUE solutions. Moreover, when the toll level is increased, the decreasing of the toll road usage for the constant and discrete VOT cases is more dramatic than that for the normal distribution VOT case. Using the random parameter model with a normal VOT distribution as a benchmark, the constant VOT model overestimates the toll road usage when the toll charge is low and underestimates the toll road usage when the toll charge is high. The impact of estimation biases in terms of the toll road usage is also reflected in the overall network performance, in terms of average trip time. The experimental results also provide toll operators useful information: when the toll level changes, users' reactions are not as dramatic as what had been predicted by DTA models with the single constant VOT assumption.

# Chapter 5 Model and Algorithm for the Multi-Criterion Dynamic User Equilibrium Problem

## 5.1 Introduction

Conventional UE traffic assignment models typically assume in the underlying path choice decision framework that every trip-maker chooses a path with the least (experienced) travel time. However, in reality, trip-makers are likely to use paths deviating from the fastest paths. Empirical researches on route choice showed that trip-makers consider numerous other criteria in finding paths. Among those criteria, the reliability of a path, in spite of being measured/defined in different ways (e.g. variance or the difference between the $80^{th}$ and the $50^{th}$ percentiles), was recognized as a critical criterion in trip-makers' path choice decisions, especially when (arrival) time constraints would impose certain penalties on individuals. Specifically, recent studies (e.g. Abdel-Aty et al. 1997; Small et al. 2005; Liu et al. 2004) have found that commuters exhibit high values of travel time and its reliability and significant heterogeneity in those values. This form of reliability, mostly adopted in empirical studies of travelers' path choice behavior, was often regarded as the (path) travel time variability (or unpredictability). The sources of travel time variability can range from regular fluctuations of travel demand in times of day, days of week, and seasons of year to random incidents, such as adverse weather, traffic accidents, vehicle breakdowns, signal failures, road works, etc (Taylor, 1999). Jackson and Jucker (1981) suggested that including travel time variability in the impedance/disutility function might improve traffic assignment models for two reasons. First, it is considered of prime importance to trip-makers. Second, a number of criteria not included in traditional disutility functions, such as the number of interchanges (or traffic signals) and the safety on a path, may be positively correlated with the variability of travel time.

Mirchandani and Soroush (1987) were among the first to consider both variable (or probabilistic) travel times and inaccurate perceptions in a traffic assignment model and proposed a generalized traffic equilibrium model to more realistically capture trip-makers' risk-taking behavior. They represented the link travel time by a nonnegative random variable and the perception of the travel time distribution differs from trip-maker to trip-maker. Each trip-maker is assumed to choose the path minimizing his/her expected disutility. On the basis of the same expected disutility approach, Boyce et al. (1998) developed a stochastic dynamic user optimal traffic assignment model where three different risk-taking path choice behaviors: risk aversion, risk neutrality, and risk seeking, each of which was associated with a particular form of disutility function, were captured. A similar expected disutility-related multi-class (static) user equilibrium traffic assignment model in stochastic networks (but without perception errors) was proposed by Ying and Ieda (2001) to explicitly consider these three risk-taking behaviors (i.e. user classes). The total expected disutility obtained by their model was used to assess performance reliability of road networks under non-recurrent congestion. The same modeling framework was extended by Ying et al. (2004) to develop a simultaneous route and departure time user equilibrium model in stochastic networks. Other forms of travel time reliability were also considered in the network modeling literature. For example, Iida (1999) defined the probability that a trip-maker can reach his destination within given time as travel time reliability. Based on this definition, Lam and Xu (1999) developed a traffic simulator to perform reliability analysis of travel times which are assumed to follow (independent) normal distributions.

Although the impact of travel time variability on trip-makers' path choices has been considered in the literature of network modeling, most (if not all) of them has adopted the expected disutility approach which could leave out the variable nature of travel times and might not be able to justify itself in terms of satisfying the Wardrop's (1952) UE principle prescribing the equilibration of travel demands based on (actual) experienced disutility. Moreover, the heterogeneity of user's response (i.e. path choice) to the travel time variability has not been addressed in existing models.

This research aims at developing a DTA model for assessing the travel time variability (or reliability) of the network flow pattern resulting from any given traffic management strategy. Attaining this goal necessitates realistically capturing trip-makers' path choices in response to the travel time variability. The travel time variability of a path in a departure time interval is defined as the variance (or standard deviation) of experienced path travel times of vehicles entering that path in that departure time interval. Each trip-maker is assumed to choose a path that minimizes the three essential path choice criteria: out-of-pocket cost (e.g. toll), travel time, and travel time variability. By following the modeling framework typically adopted in discrete time, deterministic DUE models for describing trip-makers' path choice behavior, the (experienced) path generalized cost is defined as the sum of travel cost, travel time weighted by the value of time (VOT) and travel time variability weighted by the value of reliability (VOR). This study extends BDUE model developed in chapter 3 to the multi-criterion context by explicitly considering the travel time variability in trip-makers' path choices and allowing not only the VOT but also the VOR to be continuously distributed among trip-makers. Specifically, the multi-criterion dynamic user equilibrium (MDUE) problem is formulated as an infinite dimensional variational inequality (VI), and solved by a column generation-based solution algorithm, which embeds (i) the sequential parametric analysis method (SPAM) to obtain the set of time-dependent extreme efficient (or non-dominated) paths and the corresponding breakpoint vectors of VOT and VOR that naturally define the multiple user classes, each of which corresponds to particular ranges of VOT and VOR, (ii) the traffic simulator – DYANSMART (Jayakrishnan, et al. 1994) to capture traffic dynamics and determine experienced path travel times and their travel time standard deviations for any given path flow pattern, and (iii) the multi-class path flow updating scheme to solve the restricted multi-class dynamic user equilibrium (RMDUE) problem defined by a subset of time-dependent extreme efficient paths.

This chapter is structured as follows. Section 5.2 presents the assumptions, definition and problem statement of the MDUE problem, followed by the infinite-dimensional VI formulation of the MDUE problem in section 5.3. In section 5.4 is the overview of a column generation-based solution algorithm for finding MDUE path flow patterns. The path-finding algorithm – SPAM is presented in Section 5.5. Section 5.6 describes the RMDUE problem and the multi-class path flow updating scheme.

## 5.2 Assumptions, Definition, and Problem Statement

Given a network G = ($N$, $A$), where $N$ is the set of nodes and A is the set of directed links ($i$, $j$), $i \in N$ and $j \in N$. The time period of interest (planning horizon) is discretized into a set of small time intervals, $S = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$, where $t_0$ is the earliest possible departure time from any origin node, $\sigma$ a small time interval during which no perceptible changes in traffic conditions and/or travel cost occur, and $M$ a large number such that the intervals from $t_0$ to $t_0 + M\sigma$ cover $S$. Let $c_{ij}(t)$, $d_{ij}(t)$, and $v_{ij}(t)$ be the travel cost (e.g. toll), travel time, and travel time standard deviation, respectively, for traveling on link ($i$, $j$) in time interval $t$. Denote $r_{ij}(t)$ the number of trip-makers (i.e. vehicles) entering link ($i$, $j$) in time interval $t$. In this study, $d_{ij}(t)$ is considered as the average (experienced) link travel time over $r_{ij}(t)$ vehicles, and $v_{ij}(t)$ the standard deviation of experienced link travel times of $r_{ij}(t)$ vehicles. Presented below are other important notations and variables used in this chapter.

| | |
|---|---|
| $o$ | subscript for an origin node, $o \in O \subseteq N$. |
| $d$ | subscript for a destination node, $d \in D \subseteq N$. |
| $\tau$ | superscript for a departure time interval, $\tau = 1, \ldots, T$. |
| $\alpha$ | value of time (VOT), $\alpha \in [\alpha^{\min}, \alpha^{\max}]$. |
| $\beta$ | value of travel time reliability (VOR), $\beta \in [\beta^{\min}, \beta^{\max}]$. |
| $P(o, d, \tau)$ | the set of all feasible extreme efficient paths for a given triplet ($o$, $d$, $\tau$). |
| $p$ | subscript for a path $p \in P(o, d, \tau)$. |
| $h_{od}^{\tau}(\alpha, \beta)$ | number of vehicles with VOT $\alpha$ and VOR $\beta$ from $o$ to $d$ in time interval $\tau$. |
| $r_{odp}^{\tau}(\alpha, \beta)$ | number of vehicles with VOT $\alpha$ and VOR $\beta$ departing from $o$ to $d$ in time interval $\tau$ that are assigned to path $p \in P(o, d, \tau)$. |

$r(\alpha, \beta)$ the class-specific time-varying path flow vector for the vehicles with VOT $\alpha$ and VOR $\beta$; i.e. $r(\alpha, \beta) \equiv \{ r_{odp}^{\tau}(\alpha, \beta), \forall o, d, \tau$, and $p \in P(o, d, \tau)\}$.

$r$ the time-varying (possibly infinite number) multi-class path flow vector for the trips with all possible values of time and values of reliability; i.e. $r \equiv \{r(\alpha, \beta), \forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$ and $\beta \in [\beta^{\min}, \beta^{\max}]\}$.

$TT_{odp}^{\tau}$ average experienced path travel time for the vehicles departing from $o$ to $d$ at time $\tau$ assigned to path $p \in P(o, d, \tau)$.

$\boldsymbol{TT}$ vector of path travel times; $\boldsymbol{TT} = \{ TT_{odp}^{\tau}, \forall o, d, \tau$, and $p \in P(o, d, \tau)\}$.

$TC_{odp}^{\tau}$ average experienced path travel cost for the vehicles departing from $o$ to $d$ at time $\tau$ assigned to path $p \in P(o, d, \tau)$.

$\boldsymbol{TC}$ vector of path travel costs; $\boldsymbol{TC} = \{ TC_{odp}^{\tau}, \forall o, d, \tau$, and $p \in P(o, d, \tau)\}$.

$TV_{odp}^{\tau}$ path travel time standard deviation for the vehicles departing from $o$ to $d$ at time $\tau$ assigned to path $p \in P(o, d, \tau)$.

$\boldsymbol{TV}$ $\boldsymbol{TV} = \{ TV_{odp}^{\tau}, \forall o, d, \tau$, and $p \in P(o, d, \tau)\}$.

The link generalized travel disutility perceived by a trip-maker with VOT $\alpha$ and VOR $\beta$ from node $i$ at time interval $t$ to node $j$ is defined as:

$$g_{ij}(t) = c_{ij}(t) + \alpha \times d_{ij}(t) + \beta \times v_{ij}(t) \qquad (5.1)$$

The VOT represents how much money a trip-maker is willing to trade for a unit time saving, and the VOR reflects the monetary value perceived by a trip-maker for a unit reduction in travel time variability. To realistically reflect heterogeneity of the population, VOT and VOR in this study are considered as continuous random variables distributed across the population of trip-makers, with the density functions:

$$\phi(\alpha) > 0, \forall \alpha \in [\alpha^{\min}, \alpha^{\max}] \text{ and } \int_{\alpha^{\min}}^{\alpha^{\max}} \phi(\alpha) d\alpha = 1,$$

$\phi(\beta)>0$, $\forall\beta\in[\beta^{\min},\beta^{\max}]$ and $\int_{\beta^{\min}}^{\beta^{\max}}\phi(\beta)d\beta=1$.

Note that the distributions of VOT and VOR are assumed known, and can be estimated from survey data (e.g., Small et al., 2005) or loop detector data (e.g. Liu et al., 2004). The experienced path generalized cost perceived by a trip-maker with VOT $\alpha$ and VOR $\beta$ departing from *o* to *d* at time interval $\tau$ and assigned to path $p\in P(o,d,\tau)$ is defined as:

$$GC_{odp}^{\tau}(\alpha,\beta)=\sum\nolimits_{(i,j,t)\in p}g_{ij}(t)=TC_{odp}^{\tau}+\alpha\times TT_{odp}^{\tau}+\beta\times TV_{odp}^{\tau} \quad (5.2)$$

where $TT_{odp}^{\tau}=\sum\nolimits_{(i,j,t)\in p}d_{ij}(t)$, $TC_{odp}^{\tau}=\sum\nolimits_{(i,j,t)\in p}c_{ij}(t)$, and $TV_{odp}^{\tau}=\sum\nolimits_{(i,j,t)\in p}v_{ij}(t)$. The time-dependent origin-destination (OD) demand for the entire feasible ranges of VOT and VOR over the planning horizon (i.e. $h_{od}^{\tau}(\alpha,\beta)$, $\forall o$, $d$, $\tau$, and $\forall\alpha$ and $\beta$) is also assumed given, a priori.

The key behavioral assumption made for the path choice decision is each trip-maker chooses a path that minimizes the path generalized cost function (5.2). Specifically, for trip-makers with VOT $\alpha$ and VOR $\beta$, a path $p^*\in P(o,d,\tau)$ will be selected if and only if $GC_{odp^*}^{\tau}(\alpha,\beta)=\min_{p\in P(o,d,\tau)}GC_{odp}^{\tau}(\alpha,\beta)$. Based on this assumption, the multi-criterion dynamic user equilibrium (MDUE), a multi-criterion and dynamic extension of Wardrop's first principle (1952), is defined as:

*For each OD pair and for each departure time interval, every trip-maker cannot decrease the experienced path generalized cost with respect to that trip's particular VOT and VOR by unilaterally changing paths.*

This implies that, at MDUE, each trip-maker is assigned to a path with the time-dependent least generalized cost with respect to his/her own VOT and VOR. This definition can be viewed as the multi-criterion and dynamic extension of Dial's bi-criterion user equilibrium (1996) or Leurent's cost versus time equilibrium (1993).

Since trips with different VOT and VOR (now continuously distributed random variables) are assigned onto the same road network, the generalization of the classical dynamic user equilibrium problem (i.e. the MDUE problem) allows a large number of classes of trips to be in a simultaneous equilibrium. In the extreme case where each possible combination of VOT and VOR corresponds to a class of trips, solving for the MDUE is equivalent to determining an equilibrium state resulting from the interactions of (possibly) infinitely many classes of trips in a network. Their interactions can be reflected by assuming the (measured or actual) time-dependent path travel time functions is a function of the time-varying multi-class path flow vector $r$ (i.e. $TT^{\tau}_{odp} = TT^{\tau}_{odp}(r)$, $\forall o, d, \tau$, and $p \in P(o,d,\tau)$ ). Note that time-dependent path travel costs are assumed flow independent as link costs are considered as the input of the model from any given dynamic road pricing scheme. By definition, the path generalized cost perceived by trips with VOT $\alpha$ also depends on $r$: $GC^{\tau}_{odp}(\alpha,\beta,r) = TC^{\tau}_{odp} + \alpha \times TT^{\tau}_{odp}(r) + \beta \times TV^{\tau}_{odp}(r)$.

Based on the above definition, the MDUE conditions can be mathematically stated as the following: $\forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$, and $\beta \in [\beta^{\min}, \beta^{\max}]$,

$$r^{\tau}_{odp} * (\alpha,\beta) \times [GC^{\tau}_{odp}(\alpha,\beta,r^*) - \pi_{od}(\alpha,\beta,r^*)] = 0, \ \forall o,d,\tau, p \in P(o,d,\tau) \quad (5.3)$$

$$GC^{\tau}_{odp}(\alpha,\beta,r^*) - \pi^{\tau}_{od}(\alpha,\beta,r^*) \geq 0, \ \forall o,d,\tau, p \in P(o,d,\tau), \quad (5.4)$$

$$\sum_{p \in P(o,d,\tau)} r^{\tau}_{odp}(\alpha,\beta) = h^{\tau}_{od}(\alpha,\beta), \ \forall o,d,\tau \quad (5.5)$$

$$r^{\tau}_{odp}(\alpha,\beta) \geq 0, \ \forall o,d,\tau, p \in P(o,d,\tau), \quad (5.6)$$

where $r^* = \{r_{odp}^{\tau} * (\alpha, \beta)\}$ is a multi-class time-varying MDUE path flow vector, and $\pi_{od}^{\tau}(\alpha, \beta, r^*)$ is the time-varying minimum OD generalized travel cost, evaluated at $r^*$, for the trips with the same $(o, d, \tau, \alpha, \beta)$.

Given the assumptions and the definition above, this study aims at solving the MDUE problem, under a given dynamic road pricing scheme, to obtain a time-varying path flow pattern satisfying the MDUE conditions and the corresponding experienced (link and path) travel time variances (or standard deviations). Specifically, the focus is on determining the MDUE path flows in a vehicular network: $r_{odp}^{\tau}(\alpha, \beta)$ and $TV_{odp}^{\tau}$, $\forall o, d, \tau$, $p \in P(o, d, \tau)$ and $\forall \alpha$ and $\beta$.

## 5.3 Infinite Dimensional VI Formulation of the MDUE

Let $\Omega(\alpha, \beta) \equiv \{r(\alpha, \beta)\}$ be the set of feasible class-specific path flow vectors $r(\alpha, \beta)$ satisfying the path flow conservation constraints (5.5) and non-negativity constraints (5.6). The following proposition gives the equivalent VI formulation of the MDUE problem of interest.

**Proposition 5.1**: Solving for the MDUE flow pattern $r^*$ is equivalent to finding the solution of a system of variational inequalities: $r^*(\alpha, \beta) \in \Omega(\alpha, \beta)$ such that

$$\sum_{o \in O} \sum_{d \in D} \sum_{\tau=1}^{T} \sum_{p \in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha, \beta, r^*) \times (r_{odp}^{\tau}(\alpha, \beta) - r_{odp}^{\tau}(\alpha, \beta)^*) \geq 0,$$

$$\forall\, r(\alpha, \beta) \in \Omega(\alpha, \beta), \text{ and } \forall \alpha \in [\alpha^{\min}, \alpha^{\max}] \text{ and } \beta \in [\beta^{\min}, \beta^{\max}], \quad (5.7)$$

or in the following vector form for simplicity and clarity:

$$GC(\alpha, \beta, r^*)^T \circ (r(\alpha, \beta) - r^*(\alpha, \beta)) \geq 0,$$

$$\forall\, r(\alpha, \beta)\in \Omega(\alpha, \beta), \text{ and } \forall \alpha\in [\alpha^{\min}, \alpha^{\max}] \text{ and } \beta\in [\beta^{\min}, \beta^{\max}], \tag{5.8}$$

where $GC(\alpha, \beta, r^*)$ is the path generalized cost vector perceived by the trips with VOT $\alpha$ VOR $\beta$ and evaluated at flow pattern $r^*$, and $\circ$ denotes the inner product of the two vectors: $GC(\alpha, \beta, r^*)$ and $(r^*(\alpha, \beta) - r(\alpha, \beta))$. Since (5.7) or (5.8) is only required to hold on $[\alpha^{\min}, \alpha^{\max}]$ and $[\beta^{\min}, \beta^{\max}]$, it can be further represented by the following infinite-dimensional VI (see e.g. Marcotte and Zhu, 1997): find $r^* \equiv \{r^*(\alpha, \beta), \forall \alpha \text{ and } \beta\}$ and $r^*\in \Omega$ such that

$$GC(r^*)^T \circ (r - r^*) \geq 0, \forall\, r\in \Omega \tag{5.9}$$

where $GC(r^*)\equiv\{GC(\alpha, \beta, r^*), \forall \alpha \text{ and } \beta\}$, and $\Omega \equiv\{r\}\equiv\{\Omega(\alpha, \beta),\forall \alpha \text{ and } \beta\}$. Note that $GC(r^*)$ and $r^*$ (or $r$) have the same (possibly infinite) number of elements.

**Proof of Proposition 5.1**:

Suppose $r^*$ is a MDUE path flow vector, and let $GC(r^*)$ be the corresponding path generalized cost vector. We first establish that $r^*$ is a solution to the VI problem (5.7). From the MDUE condition (5.3), the following inequalities can be obtained.

$$\begin{aligned}
& GC_{odp}^{\tau}(\alpha,\beta,r^*)[r_{odp}^{\tau}(\alpha,\beta)-r_{odp}^{\tau}{}^*(\alpha,\beta)] \\
& \geq \pi_{od}(\alpha,\beta,r^*)[r_{odp}^{\tau}(\alpha,\beta)-r_{odp}^{\tau}{}^*(\alpha,\beta)] \\
& \quad \forall o,d,\tau,p\in P(o,d,\tau) \text{ and } \forall \alpha\in[\alpha^{\min},\alpha^{\max}] \text{ and } \beta\in[\beta^{\min},\beta^{\max}]
\end{aligned} \tag{5.10}$$

With the path flow conservation constraints (5.5), it follows that

$$\begin{aligned}
& \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d)} GC_{odp}^{\tau}(\alpha,\beta,r^*)[r_{odp}^{\tau}(\alpha,\beta)-r_{odp}^{\tau}{}^*(\alpha,\beta)] \\
& \geq \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\pi_{od}^{\tau}(\alpha,\beta,r^*)\{\sum_{p\in P(o,d,\tau)}[r_{odp}^{\tau}(\alpha,\beta)-r_{odp}^{\tau}{}^*(\alpha,\beta)]\}=0 \\
& \quad \forall \alpha\in[\alpha^{\min},\alpha^{\max}] \text{ and } \beta\in[\beta^{\min},\beta^{\max}]
\end{aligned} \tag{5.11}$$

Hence, $r^*$ is a solution to the VI problem (5.7).

We then show that a solution $r^*$ to the VI problem (5.7) is a MDUE path flow vector which satisfies conditions (5.3)–(5.6). Eq.(5.7) can be rearranged as the following:

$$\sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha,\beta,r^*)\times r_{odp}^{\tau}(\alpha,\beta)$$
$$\geq \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha,\beta,r^*)\times r_{odp}^{\tau}{}^*(\alpha,\beta)$$
$$\forall r(\alpha,\beta)\in\Omega(\alpha,\beta) \text{ and, } \forall\alpha\in[\alpha^{\min},\alpha^{\max}] \text{ and } \beta\in[\beta^{\min},\beta^{\max}]. \quad (5.12)$$

It can seen from (5.12) that $r^*(\alpha,\beta)\in\Omega(\alpha,\beta)$ is an optimal solution to the linear program

$$Minimize \sum_{o\in O}\sum_{d\in D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d,\tau)} GC_{odp}^{\tau}(\alpha,\beta,r^*)\times r_{odp}^{\tau}(\alpha,\beta) \quad (5.13)$$

Subject to (5.5) and (5.6)

Let $\pi_{od}^{\tau}(\alpha,\beta,r^*)$ , $\forall o,d,\tau$ be the corresponding dual variables for the path flow conservation constraints (5.5). Then (5.3) follows from complementary slackness, (5.4) follows from dual feasibility, and (5.5) and (5.6) follow from primal feasibility. Therefore, $r^*$ is a MDUE path flow vector. This completes the proof.

Although the theoretical guarantee of properties such as existence and uniqueness of solutions to the VI problem (5.7) can be analytically derived, it generally requires the path generalized cost function to be continuous and strictly monotone (see e.g. Marcotte and Zhu, 1997). Those properties might not be satisfied in general road networks with complex traffic controls, and hence only close-to-MDUE (multiple optima) solutions can be obtained if the condition for solution existence (uniqueness) fails to be established. The discussion of solution existence and uniqueness is beyond the scope of this study.

## 5.4 MDUE Solution Algorithm

### 5.4.1 Overview of the column generation-based algorithmic framework

Since the MDUE problem of interest seeks equilibrium network states in terms of path generalized costs of network users, a set of feasible paths on which the time-varying and heterogeneous OD demands are to be equilibrated is required for the MDUE solution algorithm. It is generally very difficult, if not impossible, to enumerate the complete set of feasible paths for all OD pairs and all possible combinations of VOT and VOR in a road network of practical size. Furthermore, only a (small) fraction of paths would carry positive flows in a MDUE solution. To avoid explicit enumeration of all possible paths, this study applies a column generation-based approach that generates a representative subset of paths with competitive generalized cost and augments the path set as needed.

The column generation-based approach augments, in the outer loop, the subset of the feasible (extreme efficient or non-dominated) paths and solves, in the inner loop, the "restricted" multi-class DUE (RMDUE) problem defined by the current subset of feasible paths. In each outer iteration *k*, the extreme non-dominated path finding algorithm – sequential parametric analysis method (SPAM) is applied to (i) obtain the breakpoints which partition the feasible ranges of VOT and VOR into many subintervals and determine the multiple user classes, and (ii) find the least generalized cost (i.e. extreme efficient or non-dominated) path for each user class. New paths, if any, are added to the current path set. The algorithm terminates if there is not any new path found for all user classes or a preset convergence criterion is satisfied; otherwise the RMDUE problem is solved by adopting the multi-class path flow updating scheme to equilibrate time-varying and heterogeneous OD demands on the current path set, before returning to the path generation step (i.e. outer loop). This multi-class path flow updating/equilibrating scheme proceeds iteratively and forms the inner loop (with iteration counter $l$) of the column generation-based solution framework, in a manner similar to the descent direction method proposed in Chapter 3 or the restricted path set equilibration scheme suggested by Larsson and Patriksson (1992). By and large, the original MDUE problem is solved in this algorithmic framework as a series of approximate RMDUE problems to progressively find MDUE solutions. This idea of obtaining VOT and VOR breakpoints that naturally determine multiple user classes and solving the RMDUE problem by equilibrating path flows in each user class bases on the assumption that, in the disutility minimization-based path choice modeling framework with convex disutility (i.e. path generalized cost) functions, all trips would choose only among the set of extreme efficient (or non-dominated) paths, and the trips in each user class behave similarly in their path choices (e.g. Dial, 1996; Marcotte and Zhu, 1997).

It is worth noting that, as also suggested by early studies on the diagonalization algorithm for asymmetric traffic assignment problems (see e.g. Sheffi, 1985; Mahmassani and Mouskos, 1988) and the experimental results reported in Chapter 3, the RMDUE problem does not have to be solved optimally in each iteration $k$, in order to strike the balance between computational efficiency and satisfactory convergence. Also embedded in this algorithmic framework is the traffic simulator – DYNASMART (Jayakrishnan et al., 1994), that performs multi-class dynamic network loadings (MDNL) to determines link travel times and experienced path generalized costs for any given path flow pattern *r*; traffic flow propagations and the vehicular spatial and temporal interactions are addressed through the traffic simulation instead of analytical calculations. The column generation-based MDUE solution algorithm is outlined below and its flow chart is presented in Figure 5.1.

**Initialization**

0. Input: (i) time-dependent OD demands for the entire feasible ranges of VOT and VOR over the planning horizon ($h_{od}^{\tau}(\alpha,\beta)$, $\forall o$, $d$, $\tau$, and $\forall\alpha$ and $\beta$), (ii) time-dependent link tolls, (iii) VOT distribution function, and (iv) initial paths and path assignment.

1. Set the outer loop iteration counter $k = 0$. Perform a MDNL by the traffic simulator to evaluate the initial path assignment and obtain link/path travel times, travel time standard deviations, and costs (i.e. ***TT***, ***TV***, and ***TC***).

**Outer Loop – generating extreme efficient path set**

2. Use the sequential parametric analysis method (SPAM) to obtain the set of time-dependent extreme efficient paths, their corresponding least generalized costs ($\pi^k$) and breakpoints of VOT and VOR that define the multi-user classes.

3. Convergence checking: if (a) there is not any new path found or (b) $k = K_{max}$ (maximum number of iterations) then stop; otherwise start the inner loop (step 4).

**Inner Loop – solving the RMDUE sub-problem**

4. Set the inner loop iteration counter $l = 0$; read the output of step 2: $\pi^l$ and VOT and VOR breakpoints, as well as the current path set and path assignment ($r^l$).

5. Update path assignment: determine path assignment $r^{l+1}$ by using the multi-class path flow updating/equilibrating scheme. Set $l = l + 1$.

6. Perform a MDNL by the traffic simulator (DYANSMART) to evaluate the new path assignment $r^l$ and obtain link/path travel times, travel time standard deviations, and costs (i.e. ***TT***, ***TV***, and ***TC***).

7. Convergence checking: if the preset convergent threshold is reached or $l = L_{max}$ (maximum number of inner iterations), then set $k = k+1$ and return to step 2 with current link travel times; otherwise go back to step 5.

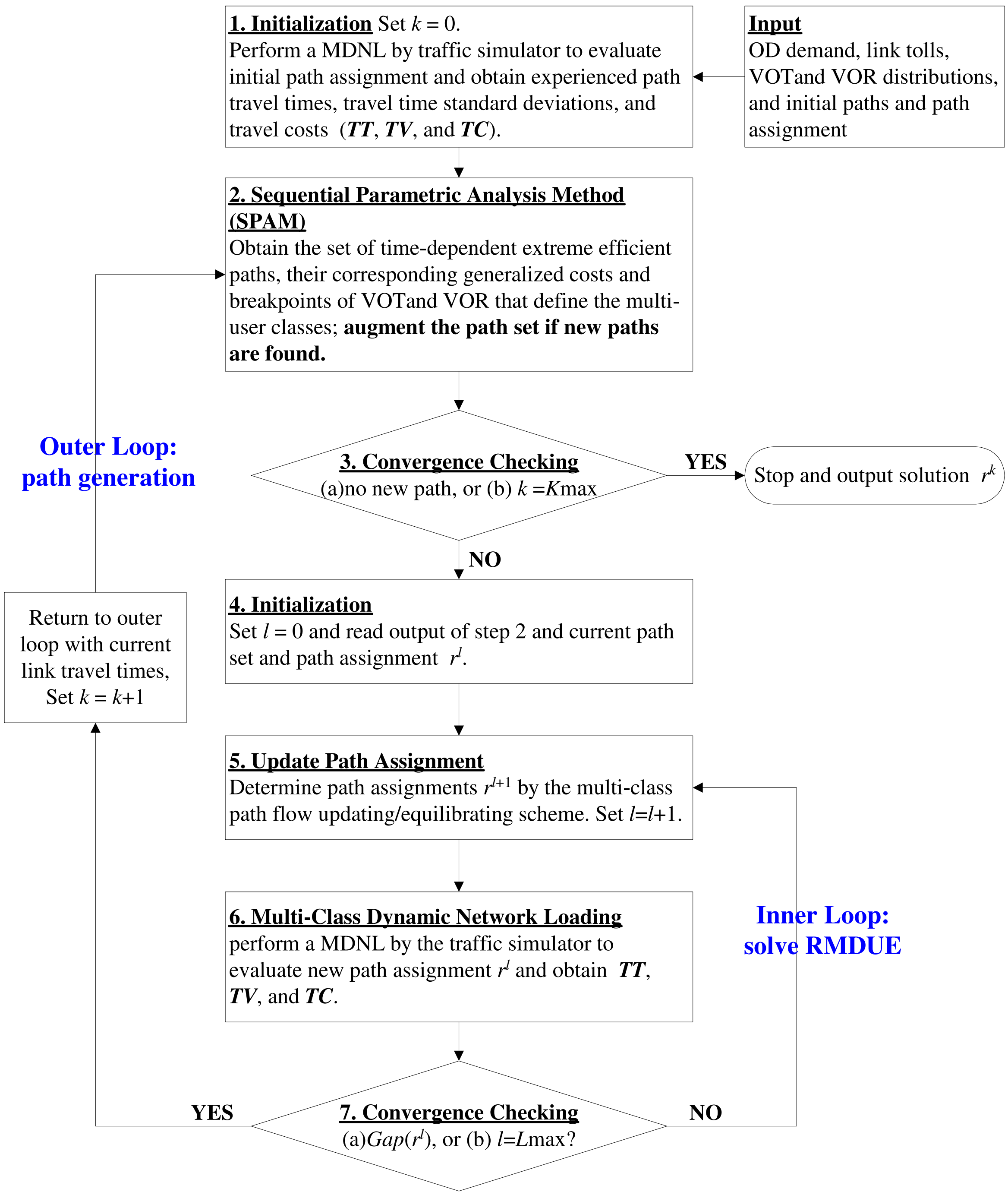



Figure 5.1 Flow chart of the MDUE solution algorithm

## 5.5 Augmenting the Extreme Efficient Path Set

The main impediment for solving the MDUE problem of interest is due largely to the relaxation of VOT and VOR from constants to continuous random variables and hence the need to find an equilibrium state resulting from the interactions of (possibly infinitely) many classes of trips, each of which corresponds to a class-specific combination of VOT and VOR, in a network. If, in the extreme case, each trip-maker (or class) requires its own set of time-dependent least generalized cost paths, finding and storing such a grand path set is computationally intractable and memory intensive in (road) network applications of practical sizes. In order to circumvent the difficulty of finding and storing the least generalized cost path for each individual trip-maker with different VOT and VOR, the Sequential Parametric Analysis Method (SPAM) is proposed to find the set of time-dependent extreme efficient (or non-dominated) path trees, each of which (i) minimizes the parametric path generalized cost function Eq.(5.2) for a particular combined VOT-VOR subinterval and (ii) consists of least generalized cost paths from a given origin to all destination nodes for all (departure) time intervals. The idea of finding the set of extreme efficient paths on which and heterogeneous trips are to be assigned is based on the assumption (see e.g. Dial, 1996; Marcotte and Zhu, 1997) that, in the disutility minimization-based path choice modeling framework with convex disutility functions, all trips would choose only among the set of extreme efficient paths corresponding to the extreme points on the efficient frontier in the criterion space.

Relying on efficiently finding the time-dependent least generalized cost path tree $Tr(\alpha,\beta)$ for given $\alpha$ and $\beta$, the SPAM adopts a computationally efficient time-dependent least cost path (TDLCP) algorithm, developed by Ziliaskopoulos and Mahmassani (1993). Each node $i \in N$ is associated with four label vectors: $\boldsymbol{\delta}_i = \{\delta_i(t)\}$, $\boldsymbol{\gamma}_i = \{\gamma_i(t)\}$, $\boldsymbol{\nu}_i = \{\nu_i(t)\}$ and $\boldsymbol{\eta}_i = \{\eta_i(t)\}$ $\forall t \in S$, corresponding to travel time, travel cost, travel time standard deviation and generalized cost, respectively, of paths from origin $r$ to node $i$ for each time interval $t$ in the planning horizon. The algorithm is based on Bellman's general principle of optimality, and the least generalized cost paths are calculated forward, starting from the origin node (in this implementation, and with no loss of generality). In each iteration, the algorithm selects and deletes the first node $i$, or "current node", from the scan eligible (SE) list. Then the current node $i$ is scanned and the labels of its downstream nodes are updated according to the following equation:

$$\eta_j(t+d_{ij}(t)) = \min\{\eta_j(t+d_{ij}(t)), g_{ij}(t) + \eta_i(t)\}, \forall t \in S, \forall j \in \Gamma\{i\}, \quad (5.14)$$

where $\Gamma\{i\}$ is the set of nodes that can be directly reached from $i$ (forward star). If at least one of the components of $\boldsymbol{\eta}_j$ is modified, node $j$ is inserted in the SE list, and the other three label vectors (i.e. $\boldsymbol{\delta}_j$, $\boldsymbol{\gamma}_j$ and $\boldsymbol{\nu}_i$) are updated accordingly. The algorithm repeats this process and terminates when the SE list is empty. The output of the algorithm includes the time-dependent least generalized cost path tree $Tr(\alpha, \beta)$ as well as the node label vectors: $\boldsymbol{\delta}_i$, $\boldsymbol{\gamma}_i$, $\boldsymbol{\nu}_i$, and $\boldsymbol{\eta}_i$ associated with each node $i$.

### 5.5.1 An example demonstrating the SPAM

Before formally presenting the SPAM, this study gives a simple example demonstrating how the SPAM works in finding the set of time-dependent extreme efficient trees and the corresponding breakpoints of VOT and VOR. As shown in Figure 5.2, starting with $\alpha^0(=\alpha^{\min})$ and $\beta^{\min}$, the SPAM first computes the least generalized cost path tree $Tr(\alpha^0, \beta^{\min})$ by using the TDLCP algorithm. With $\alpha$ fixed at $\alpha^0$, the parametric analysis of VOR $\beta$ (will be presented in the later subsection) is conducted to find the upper bound $\beta^{ub}$; the tree $Tr(\alpha^0, \beta^{\min})$ remains unchanged when $\beta^{\min} \le \beta \le \beta^{ub}$. In order to move to the next VOR segment and to obtain a different tree, a small value $\Delta_\beta$ has to be added to the bound $\beta^{ub}$. This implies that travelers cannot distinguish differences in VOR below $\Delta_\beta$ per minute. Set $\beta^1(\alpha^0) = \beta^{ub} + \Delta_\beta$. A new tree $Tr(\alpha^0, \beta^1(\alpha^0))$ can be built by applying the TDLCP algorithm, and a new upper bound $\beta^{ub}$ can be found by the parametric analysis. The same steps of tree-building and parametric analysis repeat until $\beta^{\max}$ is reached, and the set of VOR breakpoints corresponding to $\alpha^0$: $\beta(\alpha^0) = \{\beta^0(\alpha^0), \beta^1(\alpha^0), \beta^2(\alpha^0), \beta^3(\alpha^0)\}$ is obtained. The tree $Tr(\alpha^0, \beta^{\min})$ is then revisited and $\alpha^1$ is set as $\alpha^{ub} + \Delta_\alpha$, where $\alpha^{ub}$ is found by the parametric analysis of $\alpha$ (with $\beta$ fixed at $\beta^{\min}$) and $\Delta_\alpha$ represents that travelers cannot distinguish differences in VOT below $\Delta_\alpha$ per minute. With $\alpha$ fixed at $\alpha^1$, the corresponding set of VOR breakpoints $\beta(\alpha^1) = \{\beta^0(\alpha^1), \beta^1(\alpha^1), \beta^2(\alpha^1), \beta^3(\alpha^1), \beta^4(\alpha^1)\}$ is obtained by using the same process for finding $\beta(\alpha^0)$. Then, the SPAM revisits tree $Tr(\alpha^1, \beta^{\min})$ and obtains $\alpha^2$ and $\beta(\alpha^2)$. This same process of tree-building and sequential parametric analysis continues until $\alpha^{\max}$ is reached.

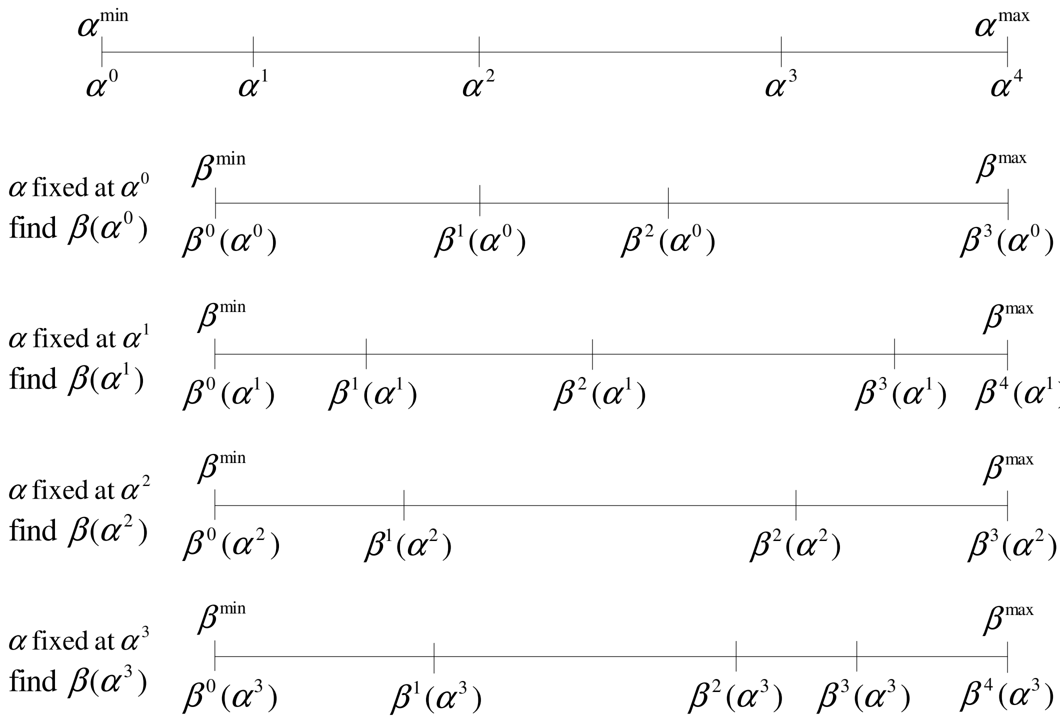



Figure 5.2 An example of the SPAM

### 5.5.2 Parametric analysis of VOR (with VOT fixed at α)

Given VOT $\alpha$ and VOR $\beta$ and the corresponding time-dependent least generalized cost path tree $Tr(\alpha, \beta)$, consisting of the time-dependent least generalized cost paths from origin $r$ to each node $i$, for each departure time interval $t$. According to the TDLCP and Eq.(5.14), if an arc-time combination $((i,j),t)$ is out-of-tree (i.e. non-tree arc), the corresponding reduced generalized cost should be nonnegative, leading to the following inequality.

$$\eta_i(t) + g_{ij}(t) - \eta_j(t + d_{ij}(t)) \geq 0. \qquad (5.15)$$

For path $p(r, i, t)$, which starts from origin $r$, at time $t$, to node $i$ the node label with respect to generalized cost can be expressed as the following:

$$\eta_i(t) = \gamma_i(t) + \alpha \times \delta_i(t) + \beta \times \upsilon_i(t) \qquad (5.16)$$

Let $t' = t + d_{ij}(t)$. Similarly, the generalized cost for path $p(r, j, t')$ from origin $r$, at time $t'$, to node $j$ can be represented as

$$\eta_j(t') = \gamma_j(t') + \alpha \times \delta_j(t') + \beta \times \upsilon_j(t') \tag{5.17}$$

Substituting Equations (5.1), (5.16) and (5.17) back into inequality (5.15) yields

$$\begin{aligned} &[\gamma_i(t) + c_{ij}(t) - \gamma_j(t')] + \alpha \times [\delta_i(t) + d_{ij}(t) - \delta_j(t')] \\ &+ \beta \times [\upsilon_i(t) + v_{ij}(t) + \upsilon_j(t')] \geq 0 \end{aligned} \tag{5.18}$$

Let $RT_{ij}(t) = \delta_i(t) + d_{ij}(t) - \delta_j(t')$, $RC_{ij}(t) = \gamma_i(t) + c_{ij}(t) - \gamma_j(t')$, and $RV_{ij}(t) = \upsilon_i(t) + v_{ij}(t) - \upsilon_j(t')$. Eq.(5.18) can be re-stated as the following:

$$RC_{ij}(t) + \alpha \times RT_{ij}(t) + \beta \times RV_{ij}(t) \geq 0 \tag{5.19}$$

Based on inequality (5.19), the dependence of the time-dependent least generalized cost path tree $Tr(\alpha, \beta)$ on the single scalar VOR $\beta$ (with VOT fixed at $\alpha$) can be examined. For any out-of-tree arc-time combination for which $RV_{ij}(t) \neq 0$, the following two cases determine the sensitivity range of $\beta$ that does not violate the reduced generalized disutility optimality conditions.

$$\text{If } RV_{ij}(t) > 0,\ \beta \geq -(RC_{ij}(t) + \alpha \times RT_{ij}(t)) / RV_{ij}(t) \tag{5.20}$$

$$\text{If } RV_{ij}(t) < 0,\ \beta \leq -(RC_{ij}(t) + \alpha \times RT_{ij}(t)) / RV_{ij}(t) \tag{5.21}$$

Collectively, we can calculate the lower and upper bounds of $\beta$ by scanning each out-of-tree arc-time combination $((i, j), t)$,

$$\beta^{lb} = \max_{((i,j),t)\notin Tr(\alpha,\beta)} \{-(RC_{ij}(t) + \alpha \times RT_{ij}(t)) / RV_{ij}(t) \mid RV_{ij}(t) > 0\} \tag{5.22}$$

$$\beta^{ub} = \min_{((i,j),t)\notin Tr(\alpha,\beta)} \{-(RC_{ij}(t) + \alpha \times RT_{ij}(t)) / RV_{ij}(t) \mid RV_{ij}(t) < 0\} \tag{5.23}$$

The time-dependent extreme efficient tree $Tr(\alpha, \beta)$ remains unchanged as long as $\beta^{lb} \le \beta \le \beta^{ub}$. In other words, the closed interval $[\beta^{lb}, \beta^{ub}]$ defines the sensitivity range of β for keeping tree $Tr(\alpha, \beta)$ optimal.

### 5.5.3 Parametric Analysis of VOT (with VOR fixed at $\beta$)

Similarly, based on inequality (5.15), the dependence of the time-dependent least generalized cost path tree $Tr(\alpha, \beta)$ on the single scalar VOT $\alpha$ (with VOR fixed at $\beta$) can be examined. For any out-of-tree arc-time combination for which $RT_{ij}(t) \neq 0$, the following two cases determine the sensitivity range of $\alpha$ that does not violate the reduced generalized disutility optimality conditions.

$$\text{If } RT_{ij}(t) > 0,\ \alpha \ge -(RC_{ij}(t) + \beta \times RV_{ij}(t)) / RT_{ij}(t) \tag{5.24}$$

$$\text{If } RT_{ij}(t) < 0,\ \alpha \le -(RC_{ij}(t) + \beta \times RV_{ij}(t)) / RT_{ij}(t) \tag{5.25}$$

Collectively, we can calculate the lower and upper bounds of $\alpha$ by scanning each out-of-tree arc-departure time combination $((i, j), t)$,

$$\alpha^{lb} = \max_{((i,j),t)\notin Tr(\alpha,\beta)} \{-(RC_{ij}(t) + \beta \times RV_{ij}(t)) / RT_{ij}(t) \mid RT_{ij}(t) > 0\} \tag{5.26}$$

$$\alpha^{ub} = \min_{((i,j),t)\notin Tr(\alpha,\beta)} \{-(RC_{ij}(t) + \beta \times RV_{ij}(t)) / RT_{ij}(t) \mid RT_{ij}(t) < 0\} \tag{5.27}$$

The time-dependent extreme efficient tree $Tr(\alpha, \beta)$ remains unchanged as long as $\alpha^{lb} \le \alpha \le \alpha^{ub}$. In other words, the closed interval $[\alpha^{lb}, \alpha^{ub}]$ defines the sensitivity range of $\alpha$ for keeping tree $Tr(\alpha, \beta)$ optimal.

### 5.5.4 Sequential parametric analysis method (SPAM)

Based on the aforementioned parametric analyses of VOT and VOR, the SPAM is now presented as follows.

**Algorithm**: Sequential Parametric Analysis Method (SPAM)

Initialize $\alpha = \alpha^{min}$ and $\beta = \beta^{min}$

**WHILE** $\alpha < \alpha^{max}$ **DO**

  **WHILE** $\beta < \beta^{max}$ **DO**

    Update link generalized costs with current VOT $\alpha$ and VOR $\beta$

    Apply TDLCP algorithm to find the tree $Tr(\alpha, \beta)$

    Initialize $\beta^{ub} = \beta^{max}$, and perform parametric analysis of VOR

    **FOR** each out-of-tree arc-time combination $((i, j), t)$ **DO**

      Calculate $\beta((i, j), t) = -(RC_{ij}(t) + \alpha \times RT_{ij}(t)) / RV_{ij}(t)$

      **IF** $\beta((i, j), t) < \beta^{ub}$ and $\beta((i, j), t) > \beta$, **THEN** $\beta^{ub} = \beta((i, j), t)$

    **END FOR**

    Set $\beta = \beta^{ub} + \Delta_\beta$, and output $\beta$

  **END WHILE**

  Set $\beta = \beta^{min}$

  Update link generalized costs with current VOT $\alpha$ and VOR $\beta$

  Apply TDLCP algorithm to find the tree $T(\alpha,\beta)$

  Initialize $\alpha^{ub} = \alpha^{max}$, and perform parametric analysis of VOT

  **FOR** each out-of-tree arc-time combination $((i, j), t)$ **DO**

    Calculate $\alpha((i, j), t) = -(RC_{ij}(t) + \beta \times RV_{ij}(t)) / RT_{ij}(t)$

    **IF** $\alpha((i, j), t) < \alpha^{ub}$ and $\alpha((i, j), t) > \alpha$, **THEN** $\alpha^{ub} = \alpha((i, j), t)$

  **END FOR**

  Set $\alpha = \alpha^{ub} + \Delta_\alpha$, and output $\alpha$

**END WHILE**

In each (outer loop) iteration *k*, the SPAM is applied to obtain the set of VOT breakpoints (k is dropped from the following notation for the ease of presentation):

$$\boldsymbol{\alpha} = \{\alpha^0, \alpha^1, ..., \alpha^B \mid \alpha^{\min} = \alpha^0 < \alpha^1 < ... < \alpha^b < ... < \alpha^B = \alpha^{\max}\}$$

that partitions the entire feasible range of VOT into $B$ subintervals: $[\alpha^{b-1}, \alpha^{b})$, $b$ = 1,…$B$, and hence defines the $B$ master user classes of trips, each master user class $u(b)$ of which covers the trips with VOT $\alpha \in [\alpha^{b-1}, \alpha^{b})$. Associated with each VOT subinterval $b$ (or master user class $u(b)$) is the set of VOR breakpoints

$$\boldsymbol{\beta}(\boldsymbol{b}) = \{\beta^{0}, \beta^{1}, ..., \beta^{M(b)} \mid \beta^{\min} = \beta^{0} < \beta^{1} < ... < \beta^{m(b)} < ... < \beta^{M(b)} = \beta^{\max}\}$$

that partitions the entire feasible range of VOR into $M(b)$ subintervals: $[\beta^{m-1}, \beta^{m})_{b}$, and defines the multiple user classes $u(b, m(b))$, $m(b)$ = 1,…$M(b)$. Each user class covers the trips with VOT $\alpha \in [\alpha^{b-1}, \alpha^{b})$ and VOR $\beta \in [\beta^{m-1}, \beta^{m})$. Associated with each $u(b, m(b))$ is the time-dependent least generalized cost path trees: $Tr(b, m(b))$, which optimizes the path generalized cost function Eq.(5.2) for the corresponding VOT subinterval $[\alpha^{b-1}, \alpha^{b})$ and VOR subinterval $[\beta^{m-1}, \beta^{m})_{b}$. If there is not any new path found for each $(o, d, \tau)$ and each user class, or the outer loop iteration counter $k$ equals $K_{max}$ (maximum number of outer iterations) then the algorithm terminate; otherwise it starts the inner loop with the output of the SPAM as well as the current path set and path assignment.

**Proposition 5.2**

The SPAM determines the *finite* number of time-dependent extreme efficient trees: $Tr(b, m(b))$ and defines the (finite) multiple user classes of trips $u(b, m(b))$, $m(b)$ = 1,…$M(b)$ and $b$ = 1,…$B$.

**Proof of Proposition 5.2**:

In the worst case scenario, the SPAM partitions the feasible range of VOT to $B^{max}$ = $(\alpha^{max} - \alpha^{min})/\Delta_{\alpha}$ subintervals. For each VOT subinterval, there will be at most $M^{max}$ =

$(\beta^{max} - \beta^{min})/\Delta_\beta$ VOR subintervals. Therefore, in the worst case scenario, the SPAM computes and defines $B^{max} \times M^{max}$ (finite) time-dependent extreme efficient trees and user classes. This completes the proof.

## 5.6 Solving the RMDUE Problem

### 5.6.1 The RMDUE problem

With the set of VOT and VOR breakpoints determined by the SPAM in a outer loop iteration $k$ of the column generation-based algorithmic framework, the entire population of heterogeneous trips in a network can be divided into a finite number of user classes, and hence the original (infinite-dimensional) MDUE problem of interest can be reduced to the (finite-dimensional) multi-class DUE problem, in which the equilibration within each user class is sought. Furthermore, since, in each iteration, the multi-class DUE is determined based on the current subset of feasible paths, the problem solved in the inner loop is termed the “restricted” multi-class DUE (or RMDUE) problem by following the terminology often adopted in the literature (e.g. Patriksson, 1994). Solving the RMDUE problem aims at finding a (finite-dimensional) multi-class time-varying path flow vector that satisfies the RMDUE definition: *for each user class, each OD pair, and each departure time interval, every trip cannot decrease the experienced path generalized cost by unilaterally changing paths*. The following variables and notations are defined (or redefined) for the RMDUE problem.

$(b, m, o, d, \tau)$ the combination of user class $u(b, m(b))$, OD pair $(o, d)$ and departure time interval $\tau$; note this is a simplified notation for $(b, m(b), o, d, \tau)$.

$P(b,m,o,d,\tau)$ (current) subset of feasible time-dependent extreme efficient paths for a $(b,m,o,d,\tau)$.

$h_{od}^{\tau}(b,m)$ number of class $u(b,m(b))$ trips departing from *o* to *d* in time interval $\tau$.

$h(b,m)$ $\sum_o \sum_d \sum_\tau h_{od}^{\tau}(b,m)$; number of class $u(b,m(b))$ trips

$r_{odp}^{\tau}(b,m)$ number of class $u(b,m(b))$ trips departing from *o* to *d* in time interval $\tau$ and assigned to path $p \in P(b,m,o,d,\tau)$.

$r_{od}^{\tau}(b,m)$ $\equiv \{r_{odp}^{\tau}(b,m), \forall p \in P(b,m,o,d,\tau)\}$; path flow vector for class $u(b,m(b))$ trips departing from *o* to *d* in time interval $\tau$.

$r(b,m)$ $\equiv \{r_{odp}^{\tau}(b,m), \forall o,d,\tau, p \in P(b,m,o,d,\tau)\}$; the class-specific path flow vector for the class $u(b,m(b))$ trips.

$r$ $\equiv \{r(b,m), m = 1,...M(b), b = 1,..., B\}$; the multi-class path flow vector.

$GC_{odp}^{\tau}(b,m,r)$ the path generalized cost of class $u(b,m(b))$ trips departing from *o* to *d* in time interval $\tau$ that are assigned to path $p \in P(b,m,o,d,\tau)$.

$GC(b,m,r)$ $\equiv \{GC_{odp}^{\tau}(b,m,r), \forall o,d,\tau, p \in P(b,m,o,d,\tau)\}$ }, the class-specific path generalized cost vector perceived by the trips of class $u(b,m(b))$ and evaluated at flow pattern *r*.

$\pi_{od}^{\tau}(b,m,r)$ least generalized cost of class $u(b,m(b))$ trips departing from *o* to *d* in time interval $\tau$, evaluated at the path assignment *r*.

Let $\Omega(b,m) = \{r(b,m)\}$ be the set of feasible class-specific path flow vectors satisfying the path flow conservation and non-negativity constraints:

$$\sum_o \sum_d \sum_\tau \sum_{p \in P(b,o,d,\tau)} r_{odp}^{\tau}(b,m) = h(b,m), \forall m = 1,...M(b), \forall b = 1,..., B \quad (5.28)$$

$$r_{odp}^{\tau}(b,m) \geq 0, \forall m,b,o,d,\tau, p \in P(b,m,o,d,\tau) \geq 0. \quad (5.29)$$

It can be obtained that, by adapting the result of Proposition 5.1, solving for the RMDUE flow pattern $r^*$ is equivalent to finding the solution of a system of variational inequalities: find $r^*(b,m) \in \Omega(b,m)$, $\forall m = 1,...M(b), \forall b = 1,...,B$, such that

$$\sum_o \sum_d \sum_\tau \sum_{p \in P(b,o,d,\tau)} GC_{odp}^{\tau}(b,m,r) \times (r_{odp}^{\tau} * (b,m) - r_{odp}^{\tau}(b,m)) \le 0, \forall r(b,m) \in \Omega(b,m) \tag{5.30}$$

or in the following vector form for simplicity and clarity:

$$GC(b,m,r^*)^T \circ (r^*(b,m) - r(b,m)) \le 0, \forall r(b,m) \in \Omega(b,m) \tag{5.31}$$

where ∘ denotes the inner product of the two vectors: $GC(b, m, r^*)$ and $(r^*(b, m) - r(b, m))$.

### 5.6.2 Multi-class path flow updating/equilibrating scheme

In the inner loop of the column generation-based algorithmic framework is a multi-class path flow updating (or equilibrating) scheme to solve the RMDUE problem and to update path assignments. This multi-class path flow updating scheme is a projection type algorithm that decomposes the RMDUE problem into many $(b,m,o,d,\tau)$ sub-problems and solves each of them by adjusting time-varying OD flows between (all) non-least generalized cost paths and the least generalized cost path(s). Given a feasible solution $r^l$ in an inner loop iteration $l$, the scheme features the following form:

$$r^{l+1} = P_\Omega[r^l - \rho^l \times Dir^l] = P_\Omega[r^l - \rho^l \times \frac{r^l \times (GC(r^l) - \pi(r^l))}{GC(r^l)}], \tag{5.32}$$

where $\rho^l \in (0,1)$ is the step size in iteration $l$, $-Dir^l$ is the descent direction, and $\pi(r^l)$ is the vector of least path generalized costs evaluated at $r^l$. $P_\Omega[u]$ denotes the unique projection of vector u onto Ω (the set of feasible multi-class path flow vectors $r$) and is defined as the unique solution of the problem: $\min_{v \in \Omega} \| u - v \|$. Based on Eq.(5.32), the new path assignment $r^{l+1}$ is obtained by updating the current path assignment $r^l$ along the descent direction ($-Dir^l$) with a move size $\rho^l$.

Let $p^*$ be the *referenced* least generalized cost path for a $(b, m, o, d, \tau)$. Specifically, for each $(b, m, o, d, \tau)$ sub-problem, the multi-class path flow updating scheme in an inner loop iteration $l$ is as follows:

$$r_{odp}^{\tau,l+1}(b,m) = \max\{0, r_{odp}^{\tau,l}(b,m) - \rho^l \times \frac{r_{odp}^{\tau,l}(b,m) \times [GC_{odp}^{\tau}(b,m,r^l) - \pi_{od}^{\tau}(b,m,r^l)]}{GC_{odp}^{\tau}(b,m,r^l)}\}$$

$$\forall p \in P(b,m,o,d,\tau), p \neq p^*; \tag{5.33}$$

$$r_{odp^*}^{\tau,l+1}(b,m) = r_{odp^*}^{\tau,l}(b,m) + \sum_{p \in P(b,m,o,d,\tau), p \neq p^*} \rho^l \times \frac{r_{odp}^{\tau,l}(b,m) \times [GC_{odp}^{\tau}(b,m,r^l) - \pi_{od}^{\tau}(b,m,r^l)]}{GC_{odp}^{\tau}(b,m,r^l)} \tag{5.34}$$

This path assignment updating/equilibrating scheme implies a natural path flow adjustment mechanism: flows on the non-cheapest paths are moved to the cheapest path and the volume moved out from a non-cheapest path $p$ is proportional to $[GC_{odp}^{\tau}(b,m,r^l) - \pi_{od}^{\tau}(b,m,r^l)] / GC_{odp}^{\tau}(b,m,r^l)$, which is intuitively based on the fact that travelers farther from the equilibrium and on paths with larger flow rates are more inclined to change path than those on paths with smaller flow rates and with travel cost closer to the minimal cost.

### 5.6.3 Multi-class dynamic network loading (MDNL) using the traffic simulator

By the MDUE definition, all trips in a network are equilibrated in terms of actual experienced path generalized costs, consisting of experienced path times and path costs, so it is necessary to determine the experienced path generalized costs $G(r)$ for a given multi-class path flow vector $r$. To this end, the simulation-based dynamic traffic (network loading) model – DYNASMART (Jayakrishnan et al., 1994) is employed to evaluate a path assignment $r$ and to obtain $GC(r)$ and time-dependent link travel times used in the path generation step. DYNASMART adopts a hybrid (mesoscopic) approach to capture the dynamics of vehicular traffic flow in the simulation, whereby vehicles are moved individually according to prevailing local speeds, consistent with macroscopic flow relations on links. It should be noted that the algorithm is independent of the specific dynamic traffic model selected; any particle-based (microscopic or mesoscopic) dynamic traffic model capable of capturing complex traffic flow dynamics can be embedded into the proposed algorithm. When a particle-based dynamic traffic model is employed to determine experienced path times, the path time $TT_{odp}^{\tau}(r)$ for a discrete time interval should be considered as the *average* path time of the vehicles with the same $(o,d,\tau,p)$, because, to respect traffic propagation rules and junction exit capacity constraints, different vehicles embarking along path $p \in P(o,d,\tau)$ in departure interval $\tau$ will normally reach their destination $d$ at different times and hence experience different trip times. This, in turn, means that the definition of RMDUE (or MDUE) in this study must involve the *average* experienced path generalized cost.

### 5.6.4 Convergence checking using gap values

Several criteria for convergence checking had been considered in the literature of DTA algorithms. For instance, Peeta and Mahmassani (1995) adopted in their simulation-based DTA model a criterion based on the comparison of path assignments (or path flows) over successive iterations. This study extends the gap-based criterion (or measure) proposed in Chapter 3 for the DUE problem to the RMDUE problem and defines the multi-class version of the gap function as the following:

$$Gap(r^l) = \sum_b \sum_{m(b)} \sum_o \sum_d \sum_\tau \sum_{p \in P(b,m,o,d,\tau)} r_{odp}^{\tau,l}(b,m) \times [GC_{odp}^{\tau}(b,m,r^l) - \pi_{od}^{\tau}(b,m,r^l)] \quad (5.35)$$

Note that, $Gap(r^l)$ provides a measure of the violation of the RMDUE conditions in terms of the difference between the total actual experienced path generalized cost and the total least generalized cost evaluated at any given multi-class path flow pattern *r*. The difference vanishes when the path flow vector $r^*$ satisfies the RMDUE conditions. In the proposed solution algorithm, for practical considerations, if $|Gap(r^l) - Gap(r^{l-1})| \leq \varepsilon$ (a predetermined convergent threshold), convergence is assumed and the program goes back to the outer loop (step 2).

### 5.6.5 Vehicle-based implementation technique

The above MDUE model and algorithm are featured as the path-based approach, necessitating the explicit storage of the path set and path assignment results for each $(b,m,o,d,\tau)$. Although it is straightforward to record all the paths and the corresponding path choice probabilities for each $(b,m,o,d,\tau)$ by using multi-dimensional arrays, computer memory requirements grow dramatically when the number of OD pairs is large, or many iterations are required to achieve convergence. Furthermore, the relaxation to the continuously distributed VOT and VOR allows a large number of classes of trips to be in a simultaneous equilibrium, each of which requires its own set of paths, and the number of user classes is unknown a priori and changes from iteration to iteration, making it more difficult to construct a memory efficient data structure for storing and updating the huge path set and path assignments in large-scale network applications. Essentially, as an attempt to accommodate greater behavioral and policy realism in applying DTA models for designing and evaluating dynamic pricing schemes, modeling heterogeneous users with a range of VOT as opposed to identical users exacerbates the computational complexity and memory requirement.

In a particle-based and simulation-based DTA system, vehicles carry their paths from iteration to iteration, and the vehicle path set implicitly reflects and stores the path set and path assignments results. This is particularly advantageous for large-scale DTA applications, as the total number of feasible paths generated by the iterative solution algorithm, after a certain number of iterations, could be significantly greater than the total number of vehicles, which is determined a priori by the OD demand table. Thus, storing the vehicle path set is more memory-efficient than storing the complete path set and routing policies for large-scale networks.

With this vehicle-based implementation technique, the path assignment updating scheme presented in Eq.(5.33) and Eq.(5.34) can be interpreted as the following. In iteration *l*, for each $(b,m,o,d,\tau)$ and for each path $p \in P(b,m,o,d,\tau)$, the number of vehicles that are moved to the (referenced) least generalized cost path is $\rho^l \times \frac{r_{odp}^{\tau,l}(b) \times [GC_{odp}^{\tau}(b,r^l) - \pi_{od}^{\tau}(b,r^l)]}{GC_{odp}^{\tau}(b,r^l)}$; and the remaining vehicles would keep their current paths. Essentially, this implementation technique uses the vehicle path set as a proxy for the exact path set and path assignment results, which can be approximately recovered from the realized vehicle paths in the last iteration's simulation results.

# Chapter 6 Solving the Multi-Criterion Simultaneous Route and Departure Time User Equilibrium Problem

## 6.1 Introduction

The BDUE problem addressed in Chapter 4 assumes the time-varying OD demands for the entire feasible range of VOT and over the planning horizon are known and fixed, a priori; or equivalently trip-makers' departure times are fixed. However, in general, a trip-maker facing a toll road with time-varying charges would not only change path (or route) but also adjust departure time so as to minimize his/her total trip cost. Some analytical studies (e.g., Arnot et al., 1990, who applied a joint departure time and route choice UE model with deterministic queueing bottlenecks to systematically analyze various pricing regimes) further found that time-varying tolls generally yield greater efficiency gains than static tolls because the former reduce queueing delays by altering travelers' departure times rather than paths. Therefore, a realistic generalization of the BDUE problem is to allow trip-makers to make departure time choices, in addition to path choices, in response to time-varying toll charges.

This chapter presents the model and solution algorithm for this important extension of the BDUE problem – the multi-criterion simultaneous route and departure time user equilibrium (MSRDUE) problem, which explicitly considers heterogeneous trips (or trip-makers) with different values of time (VOT) and values of (early or late) schedule delay (VOESD or VOLSD) simultaneously choosing departure times and paths that minimize the set of trip attributes: travel time, out-of-pocket cost, and schedule delay cost (or arrival time cost defined in Janson and Robles, 1993), where *schedule delay* is determined by the difference between actual and preferred arrival times (PAT). By following the modeling framework typically adopted in discrete time, deterministic SRDUE models for describing trip-makers' joint departure time and path choice behavior, each trip-maker is assumed to choose the alternative, a combination of departure time (interval) and path, which minimizes his/her *trip cost*, defined as the sum of travel cost, travel time weighted by VOT, and early or late schedule delay weighted by VOESD or VOLSD (e.g. Ziliaskopoulos and Rao, 1999; Huang and Lam, 2002; Szeto and Lo, 2004).

With the above assumption on modeling trip-makers' departure time and path choice behavior, it is necessary for SRDUE algorithms to construct a set of feasible alternatives on which trip-makers are to be equilibrated. While some studies (e.g. Huang and Lam, 2002; Szeto and Lo, 2004) focusing on investigating theoretical insights or equilibration methods of the problem assumed the set of feasible alternatives known and fixed, a priori, time-dependent shortest path algorithms are often applied in column generation-based DTA algorithms to generate representative subset of feasible paths (or alternatives). Because the trip's schedule delay can not be determined until the arrival at the destination, applying time-dependent least cost path algorithm to compute the least generalized cost path for each departure time interval does not guarantee to find the least trip cost path for an OD pair. Furthermore, it is impossible to assume the trip cost is the sum of generalized costs of its constituent links, due to the inclusion of schedule delay cost. This non-additive nature of trip cost prohibits the direct application of existing departure time-based, time-dependent shortest path algorithms which are often adopted in DTA algorithms for determining feasible descent directions. While few studies had attempted to solve for commuters' best paths with penalties for early or late arrivals (e.g.

De Palma et al., 1990), this study develops an algorithm for computing time-dependent least cost paths for all possible arrival time intervals by considering each trip-maker chooses the alternative with the least trip cost, where an alternative is the combination of *arrival time interval* and the corresponding least generalized cost path (that arrives the destination at that arrival time interval).

For a given PAT interval and for each origin-destination (OD) pair, this modeling approach would facilitate finding the least trip cost path(s), because, given all possible (early or late) schedule delays, the least trip cost path can be found by computing the least generalized cost paths for all possible arrival time intervals. Note that the least trip cost represents the best combination of (or compromise between) path generalized cost and schedule delay cost, where *path generalized cost* is the sum of travel cost and travel time weighted by the trip's VOT. Once the best alternative (arrival time interval and the path associated with it) is selected, the corresponding departure time can be readily determined by subtracting the path travel time from that arrival time (interval). Therefore, modeling trip-makers' selections of arrival time interval is equivalent to modeling their departure time choices. A similar approach was adopted in the SRDUE model developed by Ziliaskopoulos and Rao (1999), where time-dependent least time paths for all arrival time intervals were sought.

The MSDUE problem is formulated as an infinite dimensional variational inequality (VI) problem, and solved by the column generation-based algorithmic framework which embeds (i) the (extreme non-dominated) alternative finding algorithm – SPAM (sequential parametric analysis method) to obtain the VOT, VOESD, and VOLSD breakpoints that define multiple user classes, and determine the least trip cost alternative for each user class, (ii) the traffic simulator - DYANSMART (Jayakrishnan, et al. 1994) to capture traffic dynamics and determine experienced travel times; and (iii) the multi-class alternative flow updating (or equilibrating) scheme to solve the restricted multi-class SRDUE (RMC-SRDUE) problem defined by a subset of feasible alternatives. Although the mathematical abstraction of the problem is a typical analytical formulation, this study adopts the simulation-based approach to tackle many practical aspects of the DTA applications.

This chapter is structured as follows. Section 6.2 presents the assumptions, definition and problem statement of the MSRDUE problem, followed by the infinite-dimensional VI formulation of the MSRDUE problem in section 6.3. In section 6.4 is the overview of the column generation-based MSRDUE solution algorithm. The sequential parametric analysis method (SPAM) is presented in section 6.5, and section 6.6 details the multi-class alternative flow equilibration scheme. Section 6.7 reports the experimental results illustrating the convergence behavior of the algorithm and how user heterogeneity affecting the departure time and path flow patterns and toll road usage under different dynamic road pricing scenarios. Section 6.8 summarizes this chapter.

## 6.2 Assumptions, Definition, and Problem Statement

Consider a network G = ($N$, $A$), where $N$ is the set of nodes and A is the set of directed links ($i$, $j$), $i \in N$ and $j \in N$. The time period of interest (planning horizon) is discretized into a set of small time intervals, $S = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$, where $t_0$ is the earliest possible departure time from any origin node, $\sigma$ a small time interval during which no perceptible changes in traffic conditions and/or travel cost occur, and $M$ a large number such that the intervals from $t_0$ to $t_0+M\sigma$ cover $S$. Without loss of generality, associated with each arc $(i, j)$ and time interval $t$ are two essential time-dependent arc travel impedances: time ($d_{ij}(t)$) and cost ($c_{ij}(t)$), which are required to travel from node $i$, in time interval $t$, to node $j$. Note that $d_{ij}(t)$ may include both non-congested travel time and delay, while some other cost-related arc attributes can be considered in $c_{ij}(t)$. The link generalized travel cost perceived by a trip-maker with VOT $\alpha$ from node $i$ in time interval $t$ to node $j$ is defined as:

$$g_{ij}(t)= c_{ij}(t) + \alpha \times d_{ij}(t) \tag{6.1}$$

The VOT represents how much money a trip-maker is willing to trade for a unit time saving. Presented below are the other notations and variables used in this chapter.

| | |
|---|---|
| $o$ | index for an origin node, $o = 1,\ldots, O$. |
| $d$ | index for a destination node, $d = 1,\ldots, D$. |
| $\tau$ | index for an arrival time interval, $\tau = 1,...,T1$. |
| $\theta$ | index for a preferred arrival time (PAT) interval, $\theta = 1,...,T2$. |
| $\alpha$ | value of time (VOT), $\alpha \in [\alpha^{\min}, \alpha^{\max}]$. |
| $\beta$ | value of early schedule delay (VOESD), $\beta \in [\beta^{\min}, \beta^{\max}]$. |
| $\lambda$ | value of late schedule delay (VOLSD), $\lambda \in [\lambda^{\min}, \lambda^{\max}]$. |
| $P(o,d)$ | the set of feasible paths for a given $(o,d)$ pair. |
| $p$ | index for a path $p \in P(o,d)$. |
| $h_{od}(\theta,\alpha,\beta,\lambda)$ | the number of trips with VOT $\alpha$, VOESD $\beta$, and VOLSD $\lambda$, traveling from $o$ to $d$, and expecting to arrive in time interval $\theta$; they are given as the input. |
| $r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)$ | the number of trips with VOT $\alpha$, VOESD $\beta$, and VOLSD $\lambda$, traveling from $o$ to $d$ by alternative $(\tau, p)$ and expecting to arrive in PAT interval $\theta$; those are the unknown decision variables. |

| | |
|---|---|
| $r(\theta,\alpha,\beta,\lambda)$ | the class-specific alternative flow vector for a given combination of $\theta$, $\alpha$; $\beta$, and $\lambda$; $r(\theta,\alpha,\beta,\lambda)=\{r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda),\forall(o,d),(\tau,p)\}$. |
| $r$ | the multi-class alternative flow vector for all OD pairs and all possible values of $\theta$, $\alpha$; $\beta$, and $\lambda$; $r=\{r(\theta,\alpha,\beta,\lambda),\forall\theta,\alpha,\beta,\lambda\}$. |
| $TT_{odp}^{\tau}$ | average (or unit) experienced travel time for the trips traveling from $o$ to $d$ by alternative $(\tau,p)$. |
| $\boldsymbol{TT}$ | vector of experienced travel times; $\boldsymbol{TT}$ =\{ $TT_{odp}^{\tau}$ ,$\forall o$, $d$, $\tau$, and $p\in P(o,d)$ \}. |
| $TC_{odp}^{\tau}$ | average (or unit) experienced travel cost (i.e. road toll) for the trips traveling from $o$ to $d$ by alternative $(\tau,p)$. |
| $\boldsymbol{TC}$ | vector of experienced travel costs; $\boldsymbol{TC}$ =\{ $TC_{odp}^{\tau}$ ,$\forall o$, $d$, $\tau$, and $p\in P(o,d)$ \}. |
| $\varphi_{odp}^{\tau}(\theta)$ | experienced schedule delay (or arrival cost) for the trips traveling from $o$ to $d$ by alternative $(\tau,p)$ with PAT interval $\theta$. |

The schedule delay $\varphi_{odp}^{\tau}(\theta)$ is determined according to the piece-wise linear function:

$$\varphi_{odp}^{\tau}(\theta)=\begin{cases}\beta\times(\theta^{lb}-\tau^{mid}),\text{if }\theta^{lb}>\tau^{mid},\\ 0,\text{if }\theta^{lb}\le\tau^{mid}\le\theta^{ub},\\ \lambda\times(\tau^{mid}-\theta^{ub}),\text{if }\theta^{ub}<\tau^{mid},\end{cases}\qquad(6.2)$$

where $[\theta^{lb},\theta^{ub}]$ is the range of a PAT interval $\theta$ and $\tau^{mid}$ is the middle point of an arrival time interval $\tau$. This schedule delay cost function assumes travelers incur no arrival time cost if their arrival times are in $[\theta^{lb},\theta^{ub}]$. The (average or unit) trip cost perceived and experienced by the travelers with the same ($\theta$, $\alpha$, $\beta$, $\lambda$) traveling from origin $o$ to destination $d$ by alternative $(\tau, p)$ is defined as the sum of perceived path generalized cost ($TC^{\tau}_{odp} + \alpha \times TT^{\tau}_{odp}$) and schedule delay (Eq.(6.2)) of that alternative:

$$G^{\tau}_{odp}(\theta,\alpha,\beta,\lambda) = TC^{\tau}_{odp} + \alpha \times TT^{\tau}_{odp} + \varphi^{\tau}_{odp}(\theta), \tag{6.3}$$

where $TT^{\tau}_{odp} = \sum_{(i,j,t)\in p} d_{ij}(t)$ and $TC^{\tau}_{odp} = \sum_{(i,j,t)\in p} c_{ij}(t)$. It is clear that Eq.(6.3) can be expanded as the following, by incorporating Eq.(6.2).

$$G^{\tau}_{odp}(\theta,\alpha,\beta,\lambda) = TC^{\tau}_{odp} + \alpha \times TT^{\tau}_{odp} + \beta \times ESD^{\tau}_{odp}(\theta) + \lambda \times LSD^{\tau}_{odp}(\theta) \tag{6.4}$$

where $ESD^{\tau}_{odp}(\theta) = \max\{0, \theta^{lb} - \tau^{mid}\}$ and $LSD^{\tau}_{odp}(\theta) = \max\{0, \tau^{mid} - \theta^{ub}\}$ are the early and late schedule delays, respectively, with respect to the PAT interval $\theta$.

To explicitly consider heterogeneity of the population, VOT ($\alpha$), VOESD ($\beta$), and VOLASD ($\lambda$) in this study are assumed as continuous random variables distributed across the population of trips, with the probability density functions:

$\phi(\alpha)>0$, $\forall \alpha \in [\alpha^{\min}, \alpha^{\max}]$ and $\int_{\alpha^{\min}}^{\alpha^{\max}} \phi(\alpha)d\alpha = 1$,

$\phi(\beta)>0$, $\forall \beta \in [\beta^{\min}, \beta^{\max}]$ and $\int_{\beta^{\min}}^{\beta^{\max}} \phi(\beta)d\beta = 1$, and

$\phi(\lambda)>0$, $\forall \lambda \in [\lambda^{\min}, \lambda^{\max}]$ and $\int_{\lambda^{\min}}^{\lambda^{\max}} \phi(\lambda)d\lambda = 1$ .

Note that the distributions of VOT, VOESD, and VOLSD, which could be estimated from survey data (e.g., Small et al., 2005) or loop detector data (e.g. Liu et al., 2004), are assumed known and given a priori. In general, by following the empirical results (e.g. Small, 1982) it is assumed that $\lambda > \alpha > \beta > 0$, for all trip-makers in a network; that is, trip-makers value the cost of LSD higher than the costs of time and ESD. Additionally, this study allows each trip to have its own PAT interval $\theta$ by assuming the PAT pattern follows a given discrete distribution with the probability mass function:

$$\varpi(\theta)>0,\ \forall\theta = 1, \ldots, T, \text{ and } \sum_{\theta=1}^{T2}\varpi(\theta) = 1$$

Later in this chapter, the extension to a continuous distribution of PAT will be presented. Additionally, the origin-destination (OD) demands for each OD pair ($o$, $d$), every PAT $\theta$ and the entire ranges of VOT, VOESD, and VOLSD over the planning horizon (i.e. $h_{od}(\theta,\alpha,\beta,\lambda)$) are assumed known and given a priori.

The behavioral assumption made in this study is: each trip-maker would choose the alternative (i.e. combination of arrival time and path) that minimizes his or her trip cost, defined in Eq.(6.4). Specifically, for trips with the same ($o$, $d$, $\theta$, $\alpha$, $\beta$, $\lambda$), an alternative $(\tau^*, p^*)$ will be selected if and only if

$$(\tau^*, p^*) = \arg\min_{\forall(\tau,p)} G_{odp}^{\tau}(\theta,\alpha,\beta,\lambda).$$

Note that once the arrival time interval $\tau^*$ is selected, the corresponding departure time can be readily determined by subtracting the corresponding path travel time $TT_{odp}^{\tau}$ from $\tau^*$. Thus, as previously mentioned, modeling trip-makers' selections of arrival time interval is equivalent to modeling their departure time choices. Based on this assumption, the multi-criterion simultaneous route and departure time user equilibrium (MSRDUE), a multi-criterion and dynamic extension of Wardrop's first principle (Wardrop, 1952), is defined as the following.

**Definition 6.1: MSRDUE**

*For each OD pair, every trip cannot decrease the experienced trip cost with respect to that trip's particular VOT, VOESD, VOLSD, and PAT interval by unilaterally changing departure time and/or path.*

This implies that, at MSRDUE, each trip-maker is assigned to the alternative that has the least trip cost with respect to his/her own PAT, VOT, VOESD, and VOLSD. This definition can be viewed as the heterogeneous (or multi-criterion) generalization of the simultaneous route and departure time user equilibrium (SRDUE) in the literature (e.g. Freisz et al. 1993; Zilliaskopoulos and Rao, 1999). Since trips with different VOT, VOESD, VOLSD (now continuously distributed random variables), and PAT are assigned onto the same road network, the heterogeneous generalization of the classical SRDUE problem allows a large number of classes of trips to be in a simultaneous equilibrium. In the extreme case where each possible combination of ($\theta$, $\alpha$, $\beta$, $\lambda$) corresponds to a class of trips, solving for the MSRDUE is equivalent to determining an equilibrium state resulting from the interactions of (possibly infinite) many classes of trips in a network. Their interactions can be reflected by assuming the (measured or actual) time-dependent (path) travel time function is a function of the (possibly infinite) multi-class alternative flow vector *r* (i.e. $TT^{\tau}_{odp} = TT^{\tau}_{odp}(r)$, $\forall o$, $d$, $\tau$, and $p \in P(o,d)$). Note that time-dependent (path) travel costs are assumed flow independent as link costs (or tolls) are considered as the input of the model from any given dynamic road pricing scheme. By definition (in Eq.(6.4)), the trip costs also depend on *r*: $G^{\tau}_{odp}(\theta,\alpha,\beta,\lambda)$ = $G^{\tau}_{odp}(r;\theta,\alpha,\beta,\lambda)$.

Based on the above definition, the MSRDUE conditions can be mathematically stated as the following: $\forall\theta$, $\alpha$, $\beta$, $\lambda$, and $\forall o$, $d$

$$r_{odp}^{\tau}*(\theta,\alpha,\beta,\lambda)[G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)-\pi_{od}(r^*;\theta,\alpha,\beta,\lambda)]=0\,,\ \forall(\tau,p)\,, \tag{6.5}$$

$$G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)-\pi_{od}(r^*;\theta,\alpha,\beta,\lambda)\geq 0\,,\ \forall(\tau,p)\,, \tag{6.6}$$

$$\sum_{\tau=1}^{T}\sum_{p\in P(o,d)} r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)=h_{od}(\theta,\alpha,\beta,\lambda)\,, \tag{6.7}$$

$$r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)\geq 0\,,\ \forall(\tau,p)\,, \tag{6.8}$$

where $r^*=\{r^*(\theta,\alpha,\beta,\lambda),\forall\theta,\alpha,\beta,\lambda\}$ is a multi-class MSRDUE alternative flow vector, and $\pi_{od}(r^*;\theta,\alpha,\beta,\lambda)$ is the minimum OD trip cost, evaluated at $r^*$, for the trips with the same $(o,d,\theta,\alpha,\beta,\lambda)$.

Given the assumptions and definition above, this study aims at solving the MSRDUE problem, under a given set of time-varying link tolls and given heterogeneous OD demands, to obtain temporal splits (among departure times) and spatial distributions (over paths) satisfying the MSRDUE conditions. Specifically, the focus is on determining the MSRDUE alternative flows: $r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)$, $\forall o$, $d$, $\theta$, $\alpha$, $\beta$, $\lambda$, and $(\tau,p)$ in a vehicular network.

## 6.3 Infinite-Dimensional VI Formulation of the MSRDUE

Let $\Omega(\theta,\alpha,\beta,\lambda)\equiv\{\,r(\theta,\alpha,\beta,\lambda)\,\}$ be the set of feasible class-specific alternative flow vectors satisfying the OD flow conservation constraints (6.7) and non-negativity constraints (6.8). The following proposition gives the equivalent VI formulation of the MSRDUE problem.

**Proposition 6.1**:

Solving for the MSRDUE alternative flow pattern $r^*$ is equivalent to finding the solution of a system of variational inequalities: $\forall \theta, \alpha, \beta,$ and $\lambda$,

find $r^*(\theta,\alpha,\beta,\lambda) \in \Omega(\theta,\alpha,\beta,\lambda)$, such that

$$\sum_{o=1}^{O}\sum_{d=1}^{D}\sum_{\tau=1}^{T1}\sum_{p\in P(o,d)} G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)[r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda) - r_{odp}^{\tau}{}^*(\theta,\alpha,\beta,\lambda)] \geq 0 \qquad (6.9)$$

or in the following vector form for simplicity and clarity,

$$G(r^*;\theta,\alpha,\beta,\lambda)\circ[r(\theta,\alpha,\beta,\lambda) - r^*(\theta,\alpha,\beta,\lambda)] \geq 0,$$
$$\forall r(\theta,\alpha,\beta,\lambda) \in \Omega(\theta,\alpha,\beta,\lambda), \qquad (6.10)$$

where $\circ$ denotes the inner product of two vectors with the same size. Eq. (6.10) can be further restated by the following infinite-dimensional VI (see e.g. Marcotte and Zhu, 1997): find $r^* \equiv \{ r^*(\theta,\alpha,\beta,\lambda), \forall \theta, \alpha, \beta,$ and $\lambda$ } and $r^* \in \Omega$ such that

$$G(r^*)^T \circ (r - r^*) \geq 0, \forall\ r \in \Omega \qquad (6.11)$$

where $G(r^*) \equiv \{ G(r^*;\theta,\alpha,\beta,\lambda), \forall \theta,\alpha,\beta,$ and $\lambda$ } and $\Omega = \{r\}$, the set of feasible multi-class alternative flow vectors. Note that the vectors: $G(r^*)$ and $r^*$ (or $r$) have the same (possibly infinite) number of elements.

**Proof of Proposition 6.1**:

Suppose $r^*$ is a MSRDUE alternative flow vector, and let $G(r^*)$ be the corresponding trip cost vector. We first establish that $r^*$ is a solution to the VI problem (6.9). From the MSRDUE condition (6.6), the following inequalities can be obtained.

$$G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)[r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda) - r_{odp}^{\tau}{}^*(\theta,\alpha,\beta,\lambda)]$$
$$\geq \pi_{od}(r^*;\theta,\alpha,\beta,\lambda)[r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda) - r_{odp}^{\tau}{}^*(\theta,\alpha,\beta,\lambda)]$$
$$\forall o,d,\tau, p \in P(o,d) \text{ and } \forall \theta,\alpha,\beta, \text{and } \lambda \qquad (6.12)$$

With the OD flow conservation constraints (6.7), it follows that

$$\sum_{o=1}^{O}\sum_{d=1}^{D}\sum_{\tau=1}^{T1}\sum_{p\in P(o,d)}G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)[r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)-r_{odp}^{\tau}*(\theta,\alpha,\beta,\lambda)]$$
$$\geq\sum_{o=1}^{O}\sum_{d=1}^{D}\pi_{od}(r^*;\theta,\alpha,\beta,\lambda)\{\sum_{\tau=1}^{T1}\sum_{p\in P(o,d)}[r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)-r_{odp}^{\tau}*(\theta,\alpha,\beta,\lambda)]\}=0$$
$$\forall\theta,\alpha,\beta,\text{and }\lambda \quad (6.13)$$

Hence, $r^*$ is a solution to the VI problem (6.9).

We then show that a solution $r^*$ to the VI problem (6.9) is a MSRDUE alternative flow vector which satisfies conditions (6.5)–(6.8). Eq.(6.9) can be rearranged as the following: $\forall r(\theta,\alpha,\beta,\lambda)\in\Omega(\theta,\alpha,\beta,\lambda)$

$$\begin{aligned}&\sum_{o=1}^{O}\sum_{d=1}^{D}\sum_{\tau=1}^{T1}\sum_{p\in P(o,d)}G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)\times r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda)\\&\geq\sum_{o=1}^{O}\sum_{d=1}^{D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d)}G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)\times r_{odp}^{\tau}*(\theta,\alpha,\beta,\lambda)\end{aligned},\ \forall\theta,\alpha,\beta,\text{and }\lambda \quad (6.14)$$

It can seen from (6.14) that, for each possible $(\theta,\alpha,\beta,\lambda)$ combination, $r^*(\theta,\alpha,\beta,\lambda)$ $\in\Omega(\theta,\alpha,\beta,\lambda)$ is an optimal solution to the linear program

$$Minimize\sum_{o=1}^{O}\sum_{d=1}^{D}\sum_{\tau=1}^{T}\sum_{p\in P(o,d)}G_{odp}^{\tau}(r^*;\theta,\alpha,\beta,\lambda)\times r_{odp}^{\tau}(\theta,\alpha,\beta,\lambda) \quad (6.15)$$

Subject to (6.7) and (6.8)

Let $\pi_{od}(r^*;\theta,\alpha,\beta,\lambda)$, $\forall o,d$ be the corresponding dual variables for the OD flow conservation constraints (6.7). Then (6.5) follows from complementary slackness, (6.6) follows from dual feasibility, and (6.7) and (6.8) follow from primal feasibility. Therefore, $r^*$ is a MSRDUE alternative flow vector. This completes the proof.

Although the theoretical guarantee of properties such as existence and uniqueness of solutions to the VI problem (6.9) (or the infinite dimensional VI (6.11)) can be analytically derived, it generally requires the (path) travel time function $TT_{odp}^{\tau}(r)$ (and hence trip cost function $G_{odp}^{\tau}(r;\theta,\alpha,\beta,\lambda)$) be continuous and strictly monotone (see e.g. Marcotte and Zhu, 1997). Those properties of travel time functions might not be satisfied in general road networks with complex traffic controls, and thus only close-to-BDUE (multiple optima) solutions can be obtained if the condition for solution existence (uniqueness) fails to be established. The discussion of solution existence and uniqueness is beyond the scope of this study.

## 6.4 MSRDUE Solution Algorithm

### 6.4.1 Overview of the column generation-based algorithmic framework

Since the MSRDUE problem of interest seeks network equilibrium in terms of (alternative) trip costs of network users, a set of feasible alternatives on which the given heterogeneous OD demands are to be equilibrated is required for the MSRDUE solution algorithm. It is generally very difficult, if not impossible, to enumerate the complete set of feasible alternatives for all OD pairs and all possible PAT, VOT, VOESD, and VOLSD in a road network of practical size. Furthermore, only a (small) fraction of alternatives would carry positive flows in MSRDUE solutions. To avoid explicit enumeration of all possible alternatives, this study applies a column generation-based approach that generates and augments a representative subset of alternatives with competitive trip cost.

The column generation-based approach augments, in the outer loop, the subset of feasible alternatives and solves, in the inner loop, the "restricted" multi-class simultaneous route and departure time user equilibrium (RMC-SRDUE) problem defined by the (current) subset of feasible alternatives. In each outer loop iteration *k*, the sequential parametric analysis method (SPAM) is applied to (i) obtain the breakpoints which partition the entire ranges of VOT, VOESD, and VOLSD into many subintervals and determine the multiple user classes, and (ii) find least trip cost (i.e., extreme efficient or non-dominated) alternative for each user class. New alternatives, if any, are added to the current alternative set. The algorithm terminates if there is not any new alternative found for all user classes or a preset convergence criterion is satisfied; otherwise the RMC-SRDUE problem is solved to equilibrate the heterogeneous OD demands on the current alternative set before returning to the alternative generation step (i.e. outer loop). Solving the RMC-SRDUE problem forms the inner loop (with iteration counter *l*) of the column generation-based solution framework, which features the multi-class alternative flow updating (or equilibrating) scheme that proceeds iteratively to equilibria, in a manner similar to the descent direction method proposed in Chapter 3 and the restricted path set equilibration scheme suggested by Larsson and Patriksson (1992). A particle-based (or vehicle-based) implementation technique for the multi-class alternative flow updating scheme is also proposed to facilitate the alternative flow updating scheme. By and large, the original MSRDUE problem is solved in this algorithmic framework as a series of approximate RMC-SRDUE problems to progressively find MSRDUE solutions. This idea of obtaining VOT, VOESD, and VOLSD breakpoints that naturally determine multiple user classes and solving the RMC-SRDUE problem by equilibrating alternative flows in each user class is based on the assumption that, in the disutility minimization-based departure time and path choice modeling framework with convex disutility (i.e. trip cost) functions, all trips would choose only among the set of extreme efficient (or non-dominated) alternatives, and the trips in each user class behave similarly in their departure time and path choices (e.g. Dial, 1996; Marcotte and Zhu, 1997).

It is worth noting that, as also suggested by early studies on the diagonalization algorithm for asymmetric traffic assignment problems (see e.g. Sheffi, 1985; Mahmassani and Mouskos, 1988) and experimental results reported in Chapter 3, the RMC-SRDUE problem does not have to be solved optimally in the inner loop, in order to strike the balance between computational efficiency and satisfactory convergence. Also embedded in this algorithmic framework is the traffic simulator – DYNASMART (Jayakrishnan et al., 1994; Mahmassani, 2001), that performs multi-class dynamic network loadings (MDNL) to determines link travel times and experienced (path) travel times and travel costs for any given multi-class alternative flow pattern *r*; traffic flow propagations and the vehicular spatial and temporal interactions are addressed through the traffic simulation instead of analytical calculations. The column generation-based MSRDUE solution algorithm is outlined below and its flow chart is presented in Figure 6.1.

**MSRDUE Solution Algorithm**

**Initialization**

0. Input: (1) heterogeneous demands for each OD pair ($o$, $d$), every PAT $\theta$ and the entire ranges of VOT, VOESD, and VOLSD over the planning horizon (i.e. $r_{od}(\theta, \alpha, \beta, \lambda)$), (2) time-dependent link tolls, (3) VOT, VOESD, and VOLSD distributions, and (4) initial alternatives and assignment.
1. Set the outer loop iteration counter $k$ = 0. Perform network loading with the traffic simulator to evaluate the initial assignment and obtain time-dependent link travel times and experienced (path) travel times and costs (i.e. ***TT*** and ***TC***).

**Outer Loop – augmenting the alternative set**

2. Use the sequential parametric analysis method (SPAM) to obtain the set of (extreme efficient) alternatives, their corresponding least trip costs ($\pi^k$) and breakpoints of VOT, VOESD, and VOLSD that define the multi-user classes.
3. Convergence checking: if (a) there is not any new alternative found or (b) $k = K_{max}$ (maximum number of outer loop iterations) then stop; otherwise start the inner loop (go to step 4).

**Inner Loop – solving the RMC-SRDUE problem**

4. Set the inner loop iteration counter $l$ = 0; read the output of step 2: $\pi^l$ and VOT, VOESD, and VOLSD breakpoints, as well as the current alternative set (and ***TT*** and ***TC***) and assignment ($r^l$).
5. Update assignment: determine assignment $r^{l+1}$ by using multi-class alternative flow updating (or equilibrating) scheme. Set $l = l + 1$.
6. Perform a MDNL by the traffic simulator to evaluate the new assignment $r^l$ and obtain experienced path travel times and costs (i.e. ***TT*** and ***TC***).
7. Convergence checking: If the preset convergent threshold is reached or $l = L_{max}$ (maximum number of inner iterations), then set $k = k+1$ and return to step 2 with current link travel times; otherwise go back to step 5.

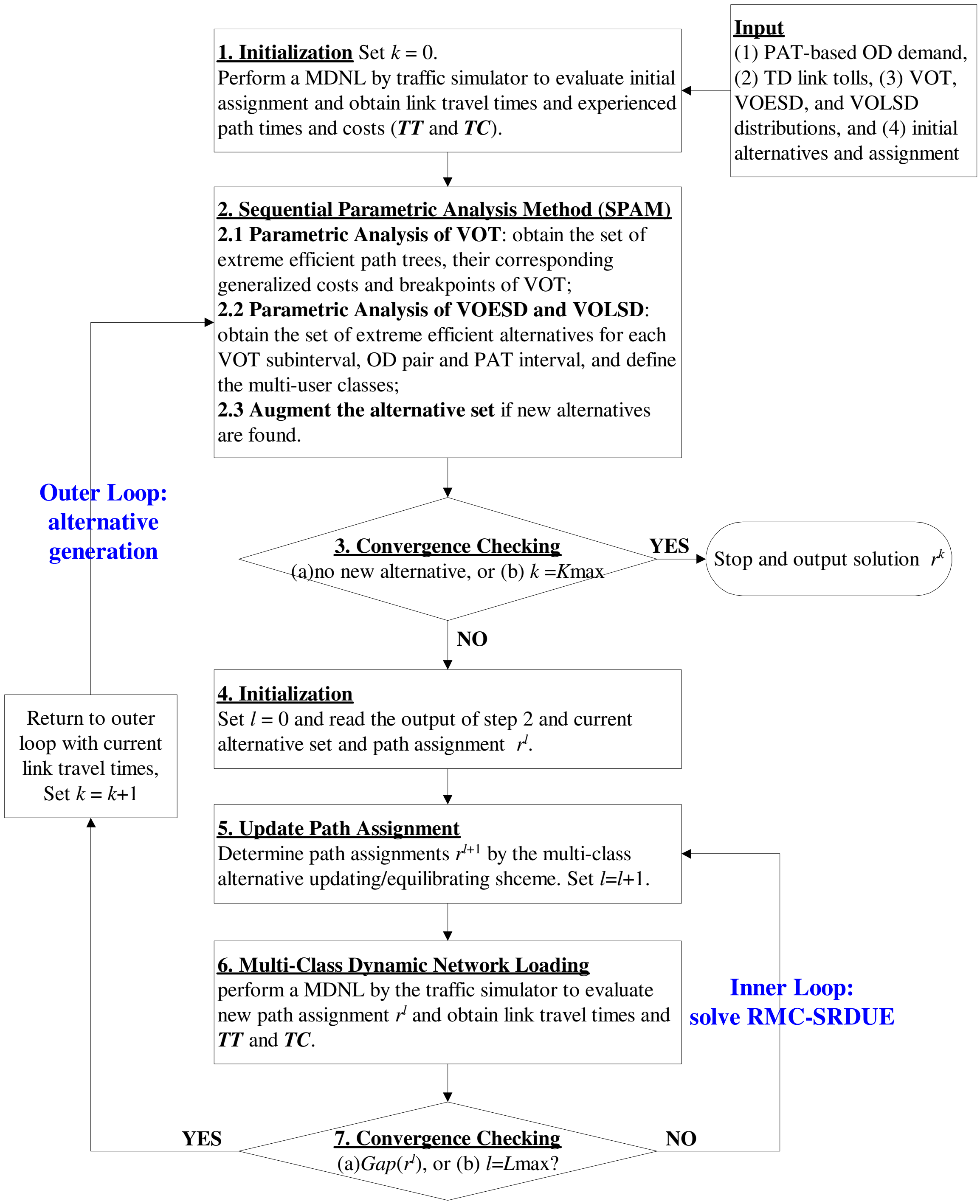



Figure 6.1 Flow chart of the MSRDUE solution algorithm

## 6.5 Augmenting the Alternative Set by SPAM

The major hurdle of solving the MSRDUE problem of interest is due to the relaxation of PAT, VOT, VOESD, and VOLSD from constants to discrete or continuous random variables and hence the need to find an equilibrium state resulting from the interactions of (possibly infinitely) many classes of trips, each of which corresponds to a class-specific combination of ($\theta$, $\alpha$, $\beta$, $\lambda$), in a network. If, in the extreme case, each trip-maker (or class) requires its own set of feasible alternatives for all OD pairs, finding and storing such a grand alternative set is computationally intractable and memory intensive in (road) network applications of practical sizes. In order to circumvent the difficulty of finding and storing the least trip cost alternative for each individual trip-maker with different PAT, VOT, VOESD, and VOLSD, the Sequential Parametric Analysis Method (SPAM) is proposed to find the set of extreme efficient (or non-dominated) alternative trees, each of which minimizes the parametric trip cost function Eq.(6.4) for a particular PAT interval and certain subintervals of VOT, VOESD, and VOLSD. The idea of finding the set of extreme efficient alternatives on which heterogeneous trip-makers are to be assigned is based on the assumption (e.g. Dial, 1996; Marcotte and Zhu, 1997) that in the disutility minimization-based path choice modeling framework with convex disutility functions, all trips would choose only among the set of extreme efficient paths corresponding to the extreme points on the efficient frontier in the criterion space.

The sequential parametric analysis method (SPAM) consists of two stages: (i) parametric analysis of VOT ($\alpha$) and (ii) parametric analyses of VOESD ($\beta$) and VOLSD ($\lambda$) for a given VOT subinterval (Figure 6.2). Since an alternative refers to a combination of arrival time and path in this study, in the first stage of the SPAM embeds the path searching algorithm which computes, for each VOT subinterval determined by the parametric analysis of VOT, the time-dependent least (generalized) cost path tree from all origin nodes to a destination node for all possible arrival times. In the second stage, each of those so-obtained trees (rooted at the same destination but corresponding to different VOT subintervals) is then parametrically analyzed with respect to VOESD and VOLSD to determine VOESD and VOLSD subintervals and least trip cost alternatives for those subintervals. The above two-stage process is repeated for each destination node. In each iteration *k*, the SPAM is performed to find the set of extreme efficient alternatives for all OD pairs and the corresponding breakpoints that partition the feasible ranges of VOT, VOESD, and VOLSD and define the multiple user classes for the RMC-SRDUE problem solved in the inner loop of the column generation-based MSRDUE solution algorithm. Note that the iteration counter *k* is dropped from the notations in this section for simplicity and clarity.

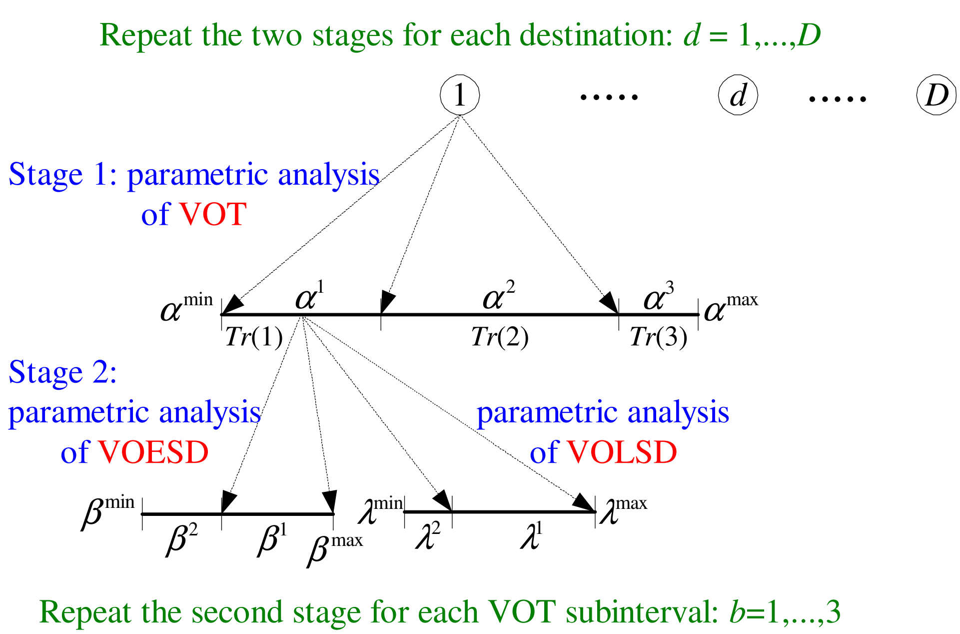



Figure 6.2 Sequential Parametric Analysis Method

### 6.5.1 Time-dependent least cost paths with fixed arrival times (TDLCP-FAT)

This study develops a TDLCP-FAT algorithm that computes the least (generalized) cost paths from all nodes $i$ to a destination node $d$ for all arrival time intervals $\tau \in S$. Each node $i \in N$ is associated with three label vectors: $\boldsymbol{\eta}_i = \{\eta_i(\tau), \forall \tau \in S\}$, $\boldsymbol{\delta}_i = \{\delta_i(\tau), \forall \tau \in S\}$, and $\boldsymbol{\gamma}_i = \{\gamma_i(\tau), \forall \tau \in S\}$, where $\eta_i(\tau)$, $\delta_i(\tau)$, and $\gamma_i(\tau)$ are the generalized cost, travel time, and travel cost, respectively, of a path from node $i$ to destination $d$ that arrives at time interval $\tau$. For a path from node $i$ that arrives $d$ at time interval $\tau$, the corresponding departure time from node $i$ can be determined as $\tau - \delta_i(\tau)$.

The TDLCP algorithm is based on Bellman's general principle of optimality, and the least (generalized) cost paths are calculated in a backward fashion (i.e., starting from the destination node). In each iteration, the algorithm selects and deletes the first node $j$, or "current node", from the scan eligible (SE) list. Then the current node $j$ is scanned and the labels of its downstream nodes are updated according to the following equation:

$$\eta_i(\tau) = \min\{\eta_i(\tau), g_{ij}(t_{\min}) + \eta_j(\tau)\}, \forall i \in \Gamma^{-1}(j), \forall \tau \in S \qquad (6.16)$$

where $t_{\min} = \arg\min\{g_{ij}(t) + \eta_j(\tau), \forall t \in \Psi(i,j,\tau)\}$, and

$\Psi(i,j,\tau) = \{t \mid t \in S, t + d_{ij}(t) = \tau - \delta_j(\tau)\}$, $\Gamma^{-1}\{j\}$ is the set of predecessor nodes of $j$ (backward star). If at least one of the components in $\boldsymbol{\eta}_i$ is modified, node $i$ is inserted in the SE list. The algorithm repeats this process and terminates when the SE list is empty. Note that the TDLCP-FAT algorithm operates in a label-correcting fashion, and hence the labels ($\boldsymbol{\eta}_i$, $\forall i \in N$) are upper bounds to the least generalized cost paths until the algorithm terminates. The TDLCP-FAT algorithm is stated as follows.

Step 1: Initialization

1.1 Initialize the label vectors as the following:

$\eta_d(\tau) = 0, \forall \lambda \in S$; $\eta_i(\tau) = \infty, \forall i \in N, \forall \lambda \in S$.

$\gamma_d(\tau) = 0, \forall \lambda \in S$; $\gamma_i(\tau) = \infty, \forall i \in N, \forall \lambda \in S$.

$\boldsymbol{\delta}_d = \{t_0, t_0 + \sigma, t_0 + 2\sigma, \ldots, t_0 + M\sigma\}$; $\delta_i(\tau) = \infty, \forall i \in N, \forall \lambda \in S$.

1.2 Create the SE list and insert into it the destination node $d$.

Step 2: Scanning and updating labels

2.1 If the SE list is empty, then terminate the algorithm; otherwise, select the first node $j$ from the SE list and remove $j$ from the list.

2.2 $\forall i \in \Gamma^{-1}(j)$ and $\forall \lambda \in S$,

2.2.1 Determine $\Psi(i, j, \tau) = \{t \mid t \in S, t + d_{ij}(t) = \tau - \delta_j(\tau)\}$

2.2.2 Find $t_{\min} = \arg\min\{g_{ij}(t) + \eta_j(\tau), \forall t \in \Psi(i, j, \tau)\}$

2.2.3 Update $\eta_i(\tau) = \min\{\eta_i(\tau), g_{ij}(t_{\min}) + \eta_j(\tau)\}, \forall i \in \Gamma^{-1}(j), \forall \tau \in S$

2.2.4 If $\eta_i(\tau)$ is updated in 2.2.3, $\delta_i(\tau) = d_{ij}(t_{min}) + \delta_j(\tau)$ and $\gamma_i(\tau) = c_{ij}(t_{min}) + \gamma_j(\tau)$

2.2.5 If at least one of the $M$ labels of node $i$ is improved (i.e. updated), insert node $i$ into the SE list. Go to Step 2.1.

### 6.5.2 Parametric analysis of VOT (α)

This subsection presents the parametric analysis of VOT (for a given destination) which sequentially computes a set of time-dependent extreme efficient path trees, each of which corresponds to a VOT subinterval (i.e. optimizes the path generalized cost function $TC_{odp}^{\tau} + \alpha \times TT_{odp}^{\tau}$ for that VOT subinterval) and consists of time-dependent least generalized cost paths from all origin nodes to a given destination node for all arrival time intervals. This parametric analysis method (PAM) can be viewed as a time-dependent adaptation of the static parametric approach (e.g. Henig, 1985; Mote et al., 1991; Dial, 1997).

Relying on efficiently finding the time-dependent extreme efficient path tree $T(\alpha)$ for a given VOT $\alpha$, the PAM adopts the TDLCP-FAT algorithm, presented in the last subsection. The output of the TDLCP-FAT algorithm includes the time-dependent extreme efficient tree $T(\alpha)$ as well as the node label vectors: $\boldsymbol{\delta}_i$, $\boldsymbol{\gamma}_i$, and $\boldsymbol{\eta}_i$ associated with each node *i*. In particular, vectors $\boldsymbol{\delta}_i$ and $\boldsymbol{\gamma}_i$ are used to calculate reduced link travel time $RT_{ij}(t) = \delta_j(\tau) + d_{ij}(t) - \delta_i(\tau)$ and reduced link travel cost $RC_{ij}(t) = \gamma_j(\tau) + c_{ij}(t) - \gamma_i(\tau)$, respectively, for all out-of-tree arc-time combinations. An arc-time combination $((i,j),t)$ is said to be out-of-tree if the following inequality holds:

$$\eta_j(\tau) + g_{ij}(t) - \eta_i(\tau) \geq 0. \tag{6.17}$$

These reduced link travel times and costs are essential input for the algorithm PAM.

**Algorithm**: Parametric Analysis Method (PAM)

Initialize the current value of VOT $\alpha = \alpha^{\min}$.

**WHILE** $\alpha < \alpha^{\max}$ **DO**

Update link generalized costs with current VOT $\alpha$

Apply the TDLCP-FAT algorithm to find the tree $Tr(\alpha)$

Initialize $\alpha^{\mathrm{ub}} = \alpha^{\max}$

**FOR** each out-of-tree arc-time combination $((i, j), t)$ **DO**

Calculate $\alpha((i, j), t) = -RC_{ij}(t)/RT_{ij}(t)$

**IF** $\alpha((i, j), t) < \alpha^{\mathrm{ub}}$ and $\alpha((i, j), t) > \alpha$, **THEN** $\alpha^{\mathrm{ub}} = \alpha((i, j), t)$

**END FOR**

Set $\alpha = \alpha^{\mathrm{ub}} + \Delta(\alpha)$, and output $\alpha$.

**END WHILE.**

**Proposition 6.2**: The PAM can find the complete set of time-dependent extreme efficient path trees each of which optimizes the generalized path cost function for a VOT subinterval and consists of time-dependent least generalized cost paths from all origin nodes to a given destination node for all arrival time intervals in a network.

**Proof of Proposition 6.2**:

The PAM is based on the following *parametric analysis* of the VOT. Consider a given VOT $\alpha$ and the corresponding time-dependent extreme efficient path tree $T(\alpha)$, consisting of the time-dependent least generalized cost paths from all origin nodes to the destination node $d$ for all possible arrival time intervals $\lambda \in S$. If an arc-time combination $((i, j), t)$ remains out-of-tree (i.e. non-tree arc), the corresponding reduced generalized cost should be nonnegative, leading to the inequality (6.17). For path $p(i, d, \lambda)$, which starts from origin $i$ and arrives node $d$ at time $\lambda$, the node label with respect to generalized cost can be expressed as the sum of the node labels of travel time and travel cost.

$$\eta_i(\tau) = \gamma_i(\tau) + \alpha \times \delta_i(\tau) \tag{6.18}$$

Substituting Equations (6.1) and (6.18) back into inequality (6.17) yields

$$[\gamma_j(\tau) + c_{ij}(t) - \gamma_i(\tau)] + \alpha \times [\delta_j(\tau) + d_{ij}(t) - \delta_i(\tau)] \geq 0$$

$$\text{or} \quad RC_{ij}(t) + \alpha \times RT_{ij}(t) \geq 0 \tag{6.19}$$

Based on Inequality (6.19), the dependence of the least generalized cost path tree on the single scalar VOT can be examined. For any out-of-tree arc for which $RT_{ij}(t) \neq 0$, the following two cases determine the sensitivity range of VOT that does not violate the reduced-cost optimality conditions.

If $RT_{ij}(t) > 0$, $\alpha > -RC_{ij}(t)/RT_{ij}(t)$ (6.20)

If $RT_{ij}(t) < 0$, $\alpha < -RC_{ij}(t)/RT_{ij}(t)$ (6.21)

Collectively, we can calculate the lower and upper bounds of VOT by scanning each out-of-tree arc-time combination $((i, j), t)$,

$$\alpha^{lb} = \max_{((i,j),t)\notin Tr(\alpha)} \{-RC_{ij}(t)/RT_{ij}(t) \mid RT_{ij}(t) > 0\} \quad (6.22)$$

$$\alpha^{ub} = \min_{((i,j),t)\notin Tr(\alpha)} \{-RC_{ij}(t)/RT_{ij}(t) \mid RT_{ij}(t) < 0\} \quad (6.23)$$

The least generalized cost path tree $Tr(\alpha)$ remains unchanged as long as $\alpha^{lb} \le \alpha \le \alpha^{ub}$. In other words, the closed interval $[\alpha^{lb},\alpha^{ub}]$ defines the (sensitivity) range of VOT for keeping tree $Tr(\alpha)$ optimal. The parametric analysis forms a main building block of PAM.

Starting from the minimal feasible value of VOT ($\alpha^{min}$), the PAM solves for the time-dependent extreme efficient path tree $Tr(\alpha)$ with respect to the current $\alpha$, and determines the upper bound $\alpha^{ub}$ for which the current shortest path tree $Tr(\alpha)$ remains unchanged, by the parametric analysis. This process continues until the maximum feasible value of VOT ($\alpha^{max}$) is reached. Based on the above parametric analysis, the algorithm is able to not only sequentially enumerate all possible time-dependent extreme efficient path trees (and all corresponding sensitivity ranges of VOT) but also directly move from one extreme efficient tree (and its sensitivity range of VOT) to the next one without redundant calculations on the non-extreme efficient solutions.

On the other hand, assume there is a time-dependent extreme efficient path tree not found by the PAM. However, by performing the parametric analysis on that tree, the sensitivity range of VOT $[\alpha^{lb},\alpha^{ub}]$ obtained can be found among the ranges already identified by the PAM, because it enumerate all the possible sensitivity ranges. That tree is actually included in the solution found by the PAM, and this contradicts the assumption. Thus, the PAM can find the complete set of time dependent extreme efficient path trees. This completes the proof.

Note that in order to move to the next VOT segment and obtain a different tree, a small positive value $\Delta(\alpha)$ needs to be added to the $\alpha^{ub}$ found in parametrically analyzing the current tree. This implies that travelers cannot distinguish differences in VOT below $\Delta(\alpha)$ per time unit. The value of $\Delta(\alpha)$ also implicitly sets an upper bound for the finite number of breakpoints and trees generated using the PAM: $(\alpha^{max}-\alpha^{min})/\Delta(\alpha)$.

In each iteration ($k$), the PAM is applied to find the set of VOT breakpoints

$$\boldsymbol{\alpha}=\{\alpha^0,\alpha^1,...,\alpha^B \mid \alpha^{\min}=\alpha^0<\alpha^1<...<\alpha^b<...<\alpha^B=\alpha^{\max}\}$$

that partitions the entire feasible range of VOT into $B$ subintervals: $[\alpha^{b-1},\alpha^b)$, $b = 1,\ldots B$, and hence defines the $B$ master user classes of trips, each master user class $u(b)$ of which covers the trips with VOT $\alpha\in[\alpha^{b-1},\alpha^b)$. Also obtained by the PAM is a set of time-dependent least generalized cost path trees $Tr(b)$, $b = 1,\ldots,B$, each of which optimizes the path generalized cost function: $TC_{odp}^{\tau}+\alpha\times TT_{odp}^{\tau}$ for the corresponding VOT subinterval $[\alpha^{b-1},\alpha^b)$ and consists of time-dependent least generalized cost paths from all origin node to a given destination node for all possible arrival time intervals in a network. The set of VOT breakpoints $\boldsymbol{\alpha}$ and trees $Tr(b)$, $b = 1,\ldots,B$ are essential input for the second stage of the SPAM.

### 6.5.3 Parametric analyses of VOESD (β) and VOLSD (λ) for a VOT subinterval

Given a time-dependent extreme efficient path tree *Tr*(*b*) corresponding to the VOT subinterval $[\alpha^{b-1}, \alpha^{b})$. Without loss of generality, consider there is only one OD pair (*o*, *d*) in the network; the generalization of this approach to networks with multiple OD pairs is fairly straightforward. Let $P(o,d)$ be the set of time-dependent least generalized cost path from *o* to *d* for all arrival time intervals $\tau = 1,\ldots,T1$. Based on the TDLCP-FAT algorithm presented in subsection 6.5.1, for a path *p* from *o* to *d* arriving at time interval $\tau$, the departure time from *o* can be obtained as $\tau - \delta_o(\tau)$, and the generalized cost label at node *o* corresponding to arrival time interval $\tau$, i.e. $\eta_o(\tau)$, is $\gamma_o(\tau) + \alpha^{b-1} \times \delta_o(\tau)$ (or $TC^{\tau}_{odp} + \alpha^{b-1} \times TT^{\tau}_{odp}$). The parametric analyses for VOESD ($\beta$) and VOLSD ($\lambda$) are conducted in the expanded network (Figure 6.3) constructed by adding a dummy destination node *d'* and corresponding node-time *trip cost* labels $\eta_{d'}(\theta)$, for PAT intervals $\theta = 1,\ldots,T2$. For each alternative ($\tau$, *p*), i.e. arrival time and path combination, an artificial (dashed line in the figure) link is added to connect ($d,\tau$) and ($d',\theta$), for every $\theta = 1,\ldots,T2$. Associated with each such artificial link is the early schedule delay (ESD) or late schedule delay (LSD) of an alternative ($\tau$, *p*) with respect to a given PAT interval $\theta$.

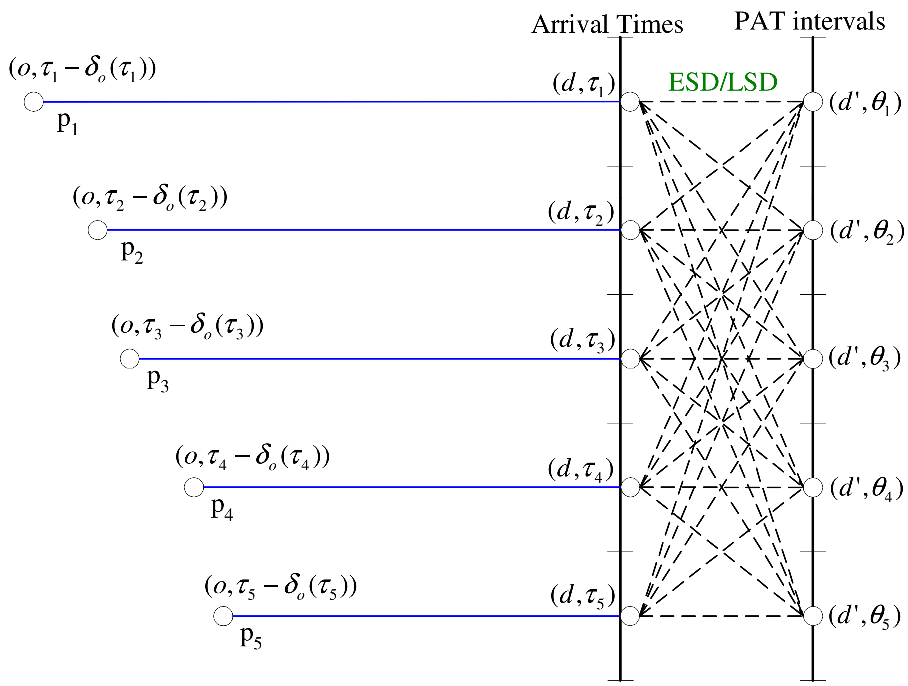



Figure 6.3 Expanded network for the parametric analyses of VOESD and VOLSD

An example demonstrating the parametric analyses of VOESD and VOLSD for a given VOT subinterval $b$ (with the corresponding tree $Tr(b)$) and a given PAT interval $\theta$ is presented in Figure 6.4. As depicted in the figure, the tree $Tr(b)$ consists of five paths ($p_1,\ldots, p_5$) from origin $o$ to destination $d$, each of which corresponds to a different arrival time interval ($\tau_1,\ldots, \tau_5$); that is, there are five alternatives. Consider the PAT interval $\theta_3$. The parametric analysis of VOESD is based on the following logic. When $\beta = \beta^{\max}$, the punctual arrival alternative ($\tau_3$, $p_3$) gives the least trip cost. As the VOESD gets smaller (i.e. move towards $\beta^{\min}$), the alternative(s) other than ($\tau_3$, $p_3$) with positive ESD may give the least trip cost (because the path generalized cost term: $TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau}$ outweighs the early schedule delay term: $ESD_{odp}^{\tau}$). For example, when $\beta > \beta_1$, the alternative ($\tau_3$, $p_3$) gives the least trip cost; whereas when $\beta \le \beta_1$, the alternative ($\tau_1$, $p_1$) is the best alternative. Thus, for the trips with $\alpha \in [\alpha^{b-1}, \alpha^{b})$, if their VOESD are in the subinterval between $\beta^{\max}$ and $\beta_1$, they will choose the alternative ($\tau_3$, $p_3$); if their VOESD are in the subinterval between $\beta_1$ and $\beta^{\min}$, the alternative ($\tau_1$, $p_1$) will be selected. The same logic can be applied to the parametric analysis of VOLSD. When $\lambda = \lambda^{\max}$, the punctual arrival alternative ($\tau_3$, $p_3$) gives the least trip cost. As the VOLSD gets smaller (i.e. move towards $\lambda^{\min}$), the alternative(s) other than ($\tau_3$, $p_3$) with positive LSD may give the least trip cost (because the path generalized cost term: $TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau}$ outweighs the late schedule delay term: $LSD_{odp}^{\tau}$). When $\lambda > \lambda_1$, the alternative ($\tau_3$, $p_3$) gives the least trip cost; whereas when $\lambda \le \lambda_1$, the alternative ($\tau_4$, $p_4$) is the best alternative. The above logic forms the basis of the parametric analyses of VOESD and VOLSD, which are presented in the following.

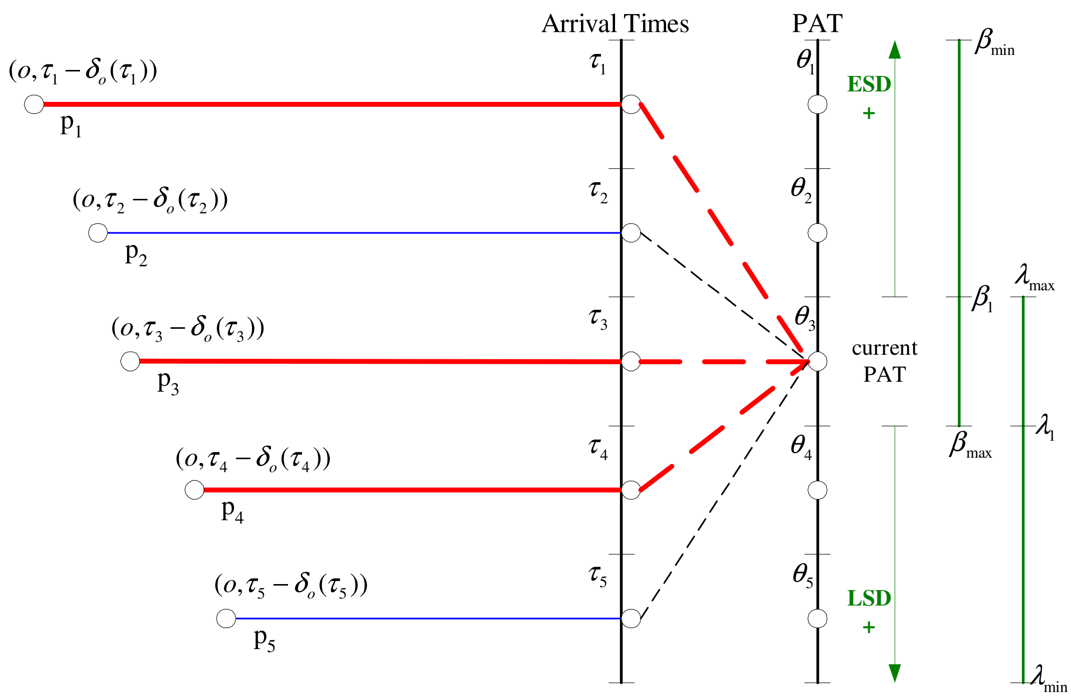



Figure 6.4 An example of parametric analyses of VOESD and VOLSD

Given the tree $Tr(b)$ corresponding to the $b^{th}$ VOT subinterval (i.e. $\alpha \in [\alpha^{b-1}, \alpha^{b})$) and a PAT interval $\theta$, the parametric analysis of VOESD starts with $\beta = \beta^{max}$. Denote ($\tau$*, $p$*) the corresponding best alternative that gives the least trip cost. The trip cost label at $d$' and corresponding to PAT $\theta$ (i.e. $\eta_{d'}(\theta)$) is updated as the following:

$$\begin{aligned} G_{odp*}^{\tau*} &= TC_{odp*}^{\tau*} + \alpha^{b-1} \times TT_{odp*}^{\tau*} + \beta \times ESD_{odp*}^{\tau*} \\ &= \gamma_o(\tau) + \alpha^{b-1} \times \delta_0(\tau) + \beta \times ESD_{odp*}^{\tau*} \end{aligned} \tag{6.16}$$

The artificial link connecting ($d$,$\tau$*) and ($d$',$\theta$) is considered as in-tree, while others are out-of-tree. If an artificial link connecting ($d$,$\tau$) and ($d$',$\theta$) and associated with positive ESD (and zero LSD) is out-of-tree, the following inequality holds:

$$TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau} + \beta \times ESD_{odp}^{\tau} \geq G_{odp*}^{\tau*}. \tag{6.17}$$

Thus, for any out-of-tree artificial link with $ESD_{odp}^{\tau} > 0$ ($LSD_{odp}^{\tau} = 0$), the following inequality determines the sensitivity range in which the artificial link connecting ($d$,$\tau$*) and ($d$',$\theta$) is still in-tree (i.e. ($\tau$*, $p$*) is still the least trip cost alternative).

$$\beta \geq \frac{G_{odp*}^{\tau*} - (TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau})}{ESD_{odp}^{\tau}}. \tag{6.18}$$

By scanning each out-of-tree artificial link with $ESD_{odp}^{\tau} > 0$, the lower bound of $\beta$ can be obtained as the following:

$$\beta^{lb} = \max_{\forall(\tau,p)\neq(\tau*,p*)} \{ \frac{G_{odp*}^{\tau*} - (TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau})}{ESD_{odp}^{\tau}} \mid ESD_{odp}^{\tau} > 0 \}. \tag{6.19}$$

Set $\beta = \beta^{lb} - \Delta(\beta)$, where $\Delta(\beta)$ is a small positive value implying that trips cannot distinguish differences in VOESD below $\Delta(\beta)$ per time unit, and update/re-evaluate the trip costs of the alternatives, that have a positive ESD, with respect to the new $\beta$. Then find the (new) best alternative ($\tau$*, $p$*) corresponding to the updated trip costs, with respect to this new $\beta$. A (new) lower bound can be obtained by using Eq(6.19). This process repeats until reaching $\beta^{min}$. For each VOT subinterval $b$ and PAT interval $\theta$, this parametric analysis of VOESD determines the set of breakpoints $\beta(b,\theta) =$

$$\{\beta^0, \beta^1, ..., \beta^{M(b,\theta)} \mid \beta^{\max} = \beta^0 > \beta^1 > ... > \beta^m > ... > \beta^{M(b,\theta)} = \beta^{\min}\}$$

that partitions entire feasible range of VOESD into $M(b,\theta)$ subintervals: $[\beta^{m-1}, \beta^m)_{b,\theta}$, $m = 1,..., M(b,\theta)$, each of which has its own best alternative $(\tau^*, p^*)_{b,\theta,m}$.

Similarly, the parametric analysis of VOLSD starts with $\lambda^{max}$ and the corresponding best alternative that gives the least trip cost. The approach keeps finding the lower bound of VOLSD ($\lambda^{lb}$) for which the current alternative remains optimal by the parametric analysis, until reaching $\lambda^{min}$. Specifically, for any out-of-tree artificial link with $LSD_{odp}^{\tau} > 0$ ($ESD_{odp}^{\tau} = 0$), the following inequality determines the sensitivity range of $\lambda$ that does not violate the optimality condition.

$$\lambda \geq \frac{G_{odp^*}^{\tau^*} - (TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau})}{LSD_{odp}^{\tau}} \qquad (6.20)$$

By scanning each out-of-tree artificial link with $LSD_{odp}^{\tau} > 0$, the lower bound of $\lambda$ can be obtained as the following:

$$\lambda^{lb} = \max_{\forall(\tau,p)\neq(\tau^*,p^*)} \{\frac{G_{odp^*}^{\tau^*} - (TC_{odp}^{\tau} + \alpha^{b-1} \times TT_{odp}^{\tau})}{LSD_{odp}^{\tau}} \mid LSD_{odp}^{\tau} > 0\}. \qquad (6.21)$$

For each VOT subinterval $b$ and PAT interval $\theta$, this parametric analysis of VOLSD determines the set of breakpoints $\lambda(b,\theta) =$

$$\{\lambda^0, \lambda^1, ..., \lambda^{N(b,\theta)} \mid \lambda^{\max} = \lambda^0 > \lambda^1 > ... > \lambda^n > ... > \lambda^{N(b,\theta)} = \lambda^{\min}\}$$

that partitions the entire feasible range of VOLSD into $N(b,\theta)$ subintervals: $[\lambda^{n-1}, \lambda^n)_{b,\theta}$, $n = 1,..., N(b,\theta)$, each of which has its own best alternative $(\tau^*, p^*)_{b,\theta,n}$. The above parametric analyses for VOESD and VOLSD are sequentially conducted for each VOT subinterval and each PAT interval.

### 6.5.4 Sequential parametric analysis method (SPAM)

The SPAM is summarized as follows.

<u>**STAGE 1**: Parametric Analysis of VOT $\alpha$</u>

Initialize the current value of VOT $\alpha = \alpha^{\min}$.

**WHILE** $\alpha < \alpha^{\max}$ **DO**

Update link generalized costs with current VOT $\alpha$

Apply the TDLCP-FAT algorithm to find the tree $T(\alpha)$

Initialize $\alpha^{ub} = \alpha^{\max}$

**FOR** each out-of-tree arc-time combination $((i, j), t)$ **DO**

Calculate $\alpha((i, j), t) = -RC_{ij}(t) / RT_{ij}(t)$

**IF** $\alpha((i, j), t) < \alpha^{ub}$ and $\alpha((i, j), t) > \alpha$, **THEN** $\alpha^{ub} = \alpha((i, j), t)$

**END FOR**

Set $\alpha = \alpha^{ub} + \Delta(\alpha)$, and output $\alpha$.

**END WHILE.**

<u>**STAGE 2**: Parametric Analyses of VOESD ($\beta$) and VOLSD ($\lambda$) for a VOT subinterval</u>

**FOR** each VOT subinterval $b$ (with its corresponding tree $Tr(b)$) **DO**

**FOR** each OD pair $(o, d)$ **DO**

Construct the expanded network by adding a dummy node and artificial links

**FOR** each PAT interval $\theta$ **DO**

Initialize the current VOESD $\beta = \beta^{max}$ and VOLSD $\lambda = \lambda^{max}$

**WHILE** $\beta > \beta^{min}$ **DO** (parametric analysis for VOESD ($\beta$))

Evaluate the trip cost of the alternatives with ESD>0, with respect to $\beta$

Find the best alternative ($\tau$*, $p$*) and consider it as in-tree

By scanning all the other (out-of-tree) alternatives with ESD>0, determine

$$\beta^{lb} = \max_{(\tau,p)}\{\frac{G^{\tau*}_{odp*} - (TC^{\tau}_{odp} + \alpha^{b-1} \times TT^{\tau}_{odp})}{ESD^{\tau}_{odp}} \mid ESD^{\tau}_{odp} > 0\}$$

Set $\beta = \beta^{lb} - \Delta(\beta)$ and output $\beta$

**END WHILE**

**WHILE** $\lambda > \lambda^{min}$ **DO** (parametric analysis for VOLSD ($\lambda$))

Evaluate the trip cost of the alternatives with LSD>0, with respect to $\lambda$

Find the best alternative ($\tau$*, $p$*) and consider it as in-tree

By scanning all the other (out-of-tree) alternatives with LSD>0, determine

$$\lambda^{lb} = \max_{(\tau,p)}\{\frac{G^{\tau*}_{odp*} - (TC^{\tau}_{odp} + \alpha^{b-1} \times TT^{\tau}_{odp})}{LSD^{\tau}_{odp}} \mid LSD^{\tau}_{odp} > 0\}$$

Set $\lambda = \lambda^{lb} - \Delta(\lambda)$ and output $\lambda$

**END WHILE**

**END FOR**

**END FOR**

**END FOR**

The output from stage 1 includes (i) the set of breakpoints $\alpha$ that defines the master user classes, each master user class $u(b)$ of which covers the trips with VOT $\alpha \in [\alpha^{b-1}, \alpha^{b})$, $b$ = 1,…,$B$. and (ii) the corresponding time-dependent extreme efficient path trees $Tr(b)$, $b$ = 1,…,$B$. The output from stage 2 includes: $\forall b, \theta$ (i) the set of breakpoints of VOESD: $\beta(b, \theta)$ that partitions entire feasible range of VOESD into

$M(b,\theta)$ subintervals: $[\beta^{m-1},\beta^{m})_{b,\theta}$, $m=1,...,M(b,\theta)$, each of which has its own least trip cost alternative $(\tau^*, p^*)_{b,\theta,m}$, and (ii) the set of breakpoints of VOLSD: $\lambda(b,\theta)$ that partitions the entire feasible range of VOLSD into $N(b,\theta)$ subintervals: $[\lambda^{n-1},\lambda^{n})_{b,\theta}$, $n=1,...,N(b,\theta)$, each of which has its own least trip cost alternative $(\tau^*, p^*)_{b,\theta,n}$.

In summary, the SPAM determines, in each outer loop iteration $k$ (as mentioned earlier, superscript $k$ is dropped from the notations in the current subsection), the breakpoints of VOT, VOESD, and VOLSD, which divide the entire population of trips into a (finite) number of user classes, and finds the least trip cost alternative for each of them. The existing alternative set is augmented by adding new alternatives to the corresponding user classes. If there is not a new alternative found for all user classes, or the outer loop iteration counter $k$ equals $K_{max}$ (maximum number of outer iterations) then the algorithm terminates; otherwise it starts the inner loop with the output of the SPAM: the breakpoints of VOT, VOESD, and VOLSD, as well as the current alternative set.

### 6.5.6 Extension to the case of a continuously distributed PAT pattern

The SPAM described in the last subsection bases on the assumption of a given discrete PAT distribution; that is, the entire planning horizon is discretized into several predefined PAT intervals, and the number of trips in each PAT interval is known, a priori. This subsection presents an extension of the SPAM to the case of a continuously distributed PAT pattern (i.e. each trip has its own PAT). In this case, the schedule delay of a trip with PAT $\theta$, VOESD $\beta$ and VOLSD $\lambda$ is defined as the following:

$$\varphi_{odp}^{\tau}(\theta) = \begin{cases} \beta \times (\theta - \tau^{mid}), \text{if } \theta > \tau^{mid}, \\ 0, \text{if } (\tau + TT_{odp}^{\tau}) = \theta, \\ \lambda \times (\tau^{mid} - \theta), \text{if } \theta < \tau^{mid}, \end{cases} \quad (6.22)$$

For a given tree $Tr(b)$ corresponding to the $b^{\text{th}}$ VOT subinterval and PAT $\theta$, the sets of VOESD and VOLSD breakpoints (i.e. $\beta(b,\theta)$ and $\lambda(b,\theta)$) can be obtained by parametrically analyzing the feasible VOESD and VOLSD ranges. The parametric analysis for PAT is conducted by scanning the out-of-tree artificial arcs for each VOESD and VOLSD subintervals. For example, as depicted in Figure 6.5, the out-of-tree arc of the subinterval $[\beta^1, \beta^{\min})_{b,\theta}$ is the arc connecting $(d, \tau_2)$ and $(d', \theta)$, corresponding to the alternative $(\tau_2, p_2)$; while the out-of-tree arc of the subinterval $[\lambda^1, \lambda^{\min})_{b,\theta}$ is the arc connecting $(d, \tau_5)$ and $(d', \theta)$, corresponding to the alternative $(\tau_5, p_5)$.

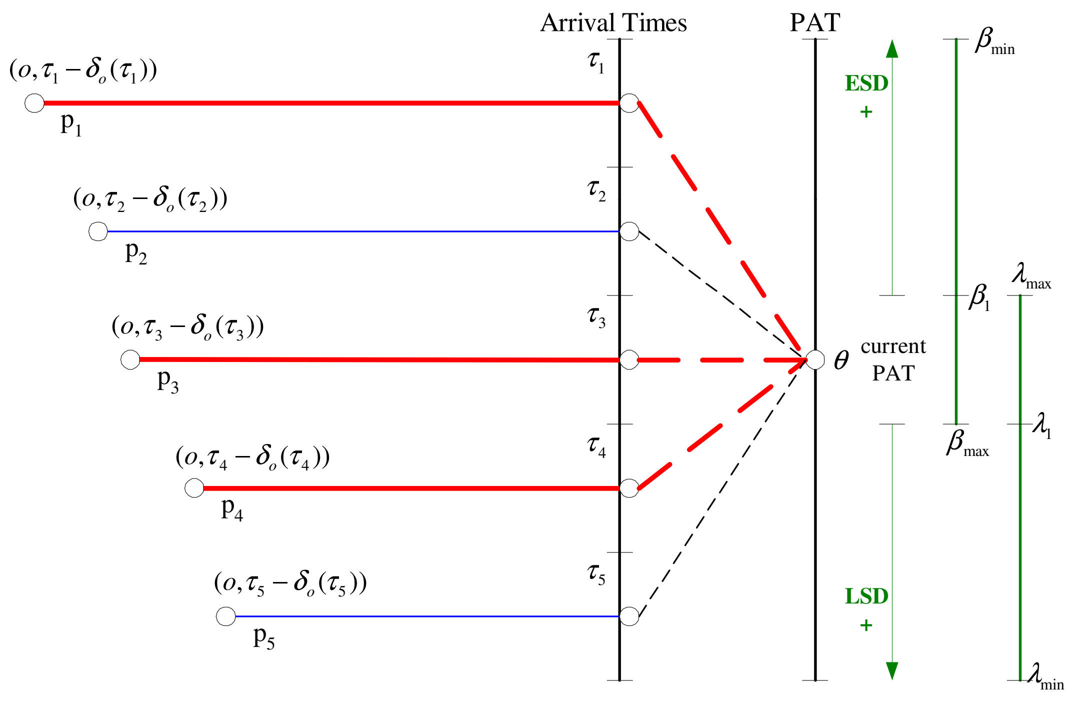



Figure 6.5 Parametric Analyses of VOESD and VOLSD for Continuous PAT Case

Specifically, for a given VOESD subinterval $[\beta^{m-1},\beta^{m})_{b,\theta}$ with its least trip cost alternative ($\tau$*, $p$*), an artificial arc is out-of-tree if the following inequality holds:

$$GC_{odp}^{\tau}(\alpha^{b-1})+\beta^{m-1}\times(\tau-\theta)\geq G_{odp^*}^{\tau^*}. \qquad (6.23)$$

where $GC_{odp}^{\tau}(\alpha^{b-1})=TC_{odp}^{\tau}+\alpha^{b-1}\times TT_{odp}^{\tau}$. It can be obtained that for an out-of-tree arc (or alternative) with $\tau-\theta>0$ (positive ESD),

$$\theta\leq[\frac{(GC_{odp}^{\tau}(\alpha^{b-1})+\beta^{m-1}\times\tau)-G_{odp^*}^{\tau^*}}{\beta^{m-1}}] \qquad (6.24)$$

Similarly, for a given VOLSD subinterval $[\lambda^{n-1},\lambda^{n})_{b,o,d,\theta}$ with its least trip cost alternative ($\tau$*, $p$*), an artificial arc is out-of-tree if the following inequality holds:

$$GC_{odp}^{\tau}(\alpha^{b-1})+\lambda^{n-1}\times(\theta-\tau)\geq G_{odp^*}^{\tau^*} \qquad (6.25)$$

It can be obtained that for an out-of-tree arc with $\theta-\tau>0$ (positive LSD),

$$\theta\geq[\frac{G_{odp^*}^{\tau^*}-(GC_{odp}^{\tau}(\alpha^{b-1})-\lambda^{n-1}\times\tau)}{\lambda^{n-1}}] \qquad (6.26)$$

Therefore, based on Eq.(5.26), the upper bound of PAT can be determined by scanning the out-of-tree alternatives in each VOT subinterval $b$ = 1,…,$B$ and each VOESD subinterval $m=1,...,M(b,\theta)$:

$$\theta^{ub}=\min\{\frac{(GC_{odp}^{\tau}(\alpha^{b-1})+\beta^{m-1}\times\tau)-G_{odp^*}^{\tau^*}}{\beta^{m-1}},\forall(\tau,p)\neq(\tau^*,p^*)_{b,\theta,m},\forall b,\forall m\} \qquad (6.27)$$

Similarly, according to Eq.(6.26), the lower bound of PAT can be found by scanning the out-of-tree alternatives in each VOT subinterval $b$ = 1,…,$B$ and each VOLSD subintervals $n=1,...,N(b,\theta)$:

$$\theta^{lb} = \max\{\frac{G_{odp^*}^{\tau^*} - (GC_{odp}^{\tau}(\alpha^{b-1}) - \lambda^{n-1} \times \tau)}{\lambda^{n-1}}, \forall(\tau, p) \neq (\tau^*, p^*)_{b,\theta,m}, \forall b, \forall m\} \quad (6.28)$$

These upper and lower bounds determine the range of PAT in which the current least trip cost alternative tree remains optimal. Essentially, starting from the earliest (or latest) PAT, the procedure applies the above parametric analysis of PAT to find the upper (or lower) bound of PAT, for which the current least trip cost alternative tree is optimal. This upper (or lower) bound is stored together with its corresponding least trip cost alternative tree. This process is repeated until reaching the latest (or earliest) PAT.

## 6.6 Solving the RMC-SRDUE Problem

### 6.6.1 The RMC-SRDUE problem

With the breakpoints of VOT, VOESD, and VOLSD determined by the SPAM in a outer loop iteration *k* of the column generation-based algorithmic framework, the entire population of heterogeneous trips in a network can be divided into a finite number of user classes, and hence the original (infinite-dimensional) MSRDUE problem of interest can be reduced to the (finite-dimensional) multi-class SRDUE problem, in which the equilibration within each user class is sought. Furthermore, since, in each iteration, the multi-class SRDUE is determined based on the current subset of feasible alternatives, the sub-problem solved in the inner loop is termed the “restricted” multi-class SRDUE (or RMC-SRDUE) problem by following the terminology often adopted in the literature (e.g. Patriksson, 1994). Based on the output of SPAM, for each VOT subinterval $b$ and PAT $\theta$, the corresponding trips are grouped into the user classes:

$$u(b, \theta, m_{\beta(b,\theta)}, n_{\lambda(b,\theta)}),\ m = 1,..., M(b,\theta),\ n = 1,..., N(b,\theta),$$

each of which is defined by a pair VOESD and VOLSD subintervals. Note that this user class notation is simplified as $u(b,\theta,m,n)$ for the ease of presentation. Denote $alt_{od}(b,\theta,m,n)$ be the set of alternatives corresponding to the user class $u(b,\theta,m,n)$ and OD pair *od*. $alt_{od}(b,\theta,m,n) = alt_{od}(b,\theta,m_{b,\theta}) \cup alt_{od}(b,\theta,n_{b,\theta})$, where $alt_{od}(b,\theta,m_{b,\theta})$ and $alt_{od}(b,\theta,n_{b,\theta})$ are the sets of alternatives corresponding to the VOESD subinterval $[\beta^{m-1},\beta^{m})_{b,\theta}$ and VOLSD subinterval $[\lambda^{n-1},\lambda^{n})_{b,\theta}$, respectively. It is important to note that while the user classes change from (outer loop) iteration to (outer loop) iteration, they are considered as fixed in the inner loop (iterations) corresponding to a given outer loop iteration. Moreover, the entire set of (current) alternatives is augmented in every outer loop iteration, and they are then re-grouped according to the user classes defined by the breakpoints obtained by the SPAM in a given iteration *k*.

Specifically, solving the RMC-SRDUE problem is to find a (finite-dimensional) multi-class alternative flow vector that satisfies the RMC-SRDUE definition: *for each user class* $u(b,\theta,m,n)$ *and each OD pair, every trip cannot decrease the experienced trip cost by unilaterally changing departure time and/or path*. The following variables and notations are defined (or redefined) for the RMC-SRDUE problem.

| | |
|---|---|
| $h_{od}(b,\theta,m,n)$ | number of trips of the user class $u(b,\theta,m,n)$ traveling from *o* to *d*. |
| $r^{\tau}_{odp}(b,\theta,m,n)$ | number of trips of user class $u(b,\theta,m,n)$ traveling from *o* to *d* and choosing the alternative $(\tau,p)\in alt_{od}(b,\theta,m,n)$. |
| $r_{od}(b,\theta,m,n)$ | $\equiv\{r^{\tau}_{odp}(b,\theta,m,n),\forall(\tau,p)\in alt_{od}(b,\theta,m,n)\}$; the class-specific alternative flow vector. |
| $r$ | $\equiv\{r(b,\theta,m,n)\}$; the multi-class path flow vector. |

$G^{\tau}_{odp}(r;b,\theta,m,n)$ the trip cost of user class $u(b,\theta,m,n)$ trips traveling from *o* to *d* and choosing the alternative $(\tau, p) \in alt_{od}(b,\theta,m,n)$, evaluated at *r*.

$\pi_{od}(r;b,\theta,m,n)$ the least trip cost of user class $u(b,\theta,m,n)$ trips traveling from *o* to *d* and, evaluated at the assignment *r*.

Let $\Omega_{od}(b,\theta,m,n) \equiv \{r_{od}(b,\theta,m,n)\}$ be the set of feasible class-specific OD alternative flow vectors satisfying the flow conservation and non-negativity constraints:

$$\sum_{(\tau,p)\in alt_{od}(b,\theta,m,n)} r^{\tau}_{odp}(b,\theta,m,n) = h_{od}(b,\theta,m,n), \tag{6.22}$$

$$r^{\tau}_{odp}(b,\theta,m,n) \geq 0, \forall(\tau, p) \in alt_{od}(b,\theta,m,n). \tag{6.23}$$

It can be obtained that, by adapting the result of Proposition 6.1, solving for the RMC-SRDUE flow pattern *r** is equivalent to finding the solution of a system of variational inequalities: $\forall b, o, d, \theta$ (i.e. for each user class $u(b,\theta,m,n)$) and $\forall o, d$,

find $r_{od}*(b,\theta,m,n) \in \Omega_{od}(b,\theta,m,n)$ such that

$$\sum_{(\tau,p)\in alt_{od}(b,\theta,m,n)} G^{\tau}_{odp}(b,\theta,m,n) \times [r^{\tau}_{odp}*(b,\theta,m,n) - r^{\tau}_{odp}(b,\theta,m,n)] \leq 0,$$

$$\forall r_{od}(b,\theta,m,n) \in \Omega_{od}(b,\theta,m,n) \tag{6.24}$$

### 6.6.2 Multi-class alternative flow updating/equilibrating scheme

In the inner loop of the column generation-based algorithmic framework is a multi-class alternative flow updating (or equilibrating) scheme to solve the RMC-SRDUE problem and to update alternative assignments. This multi-class alternative flow updating scheme decomposes the RMC-SRDUE problem into many $(b,\theta,m,n,o,d)$ sub-problems and solves each of them by adjusting OD flows between (all) non-least trip cost alternatives and the least trip cost alternative. Let $(\tau^*, p^*)$ be the *referenced* least trip cost alternative for the user class $u(b,\theta,m,n)$ and each OD pair ($o$, $d$). Specifically, for each $(b,\theta,m,n,o,d)$ sub-problem, the multi-class alternative flow updating scheme in an inner loop iteration $l$ is as follows:

$$r_{odp}^{\tau,l+1}(b,\theta,m,n) =$$
$$\max\{0, r_{odp}^{\tau,l}(b,\theta,m,n) - \rho^l \times \frac{r_{odp}^{\tau,l}(b,\theta,m,n) \times \Delta_{odp}^{\tau,l}(b,\theta,m,n)}{G_{odp}^{\tau}(r^l;b,\theta,m,n)}\}$$

$$\forall(\tau,p) \in alt_{od}(b,\theta,m,n), (\tau,p) \neq (\tau^*, p^*), \tag{6.25}$$

$$r_{odp^*}^{\tau^*,l+1}(b,\theta,m,n) = r_{odp^*}^{\tau^*,l}(b,\theta,m,n) + \psi_{od}^{l}(b,\theta,m,n), \tag{6.26}$$

where $\Delta_{odp}^{\tau,l}(b,\theta,m,n) = G_{odp}^{\tau}(r^l;b,\theta,m,n) - \pi_{od}(r^l;b,\theta,m,n)$ and

$$\psi_{od}^{l}(b,\theta,m,n) = \sum_{(\tau,p)\in alt_{od}(b,\theta,m,n)} \frac{r_{odp}^{\tau,l}(b,\theta,m,n) \times \Delta_{odp}^{\tau,l}(b,\theta,m,n)}{G_{odp}^{\tau}(r^l;b,\theta,m,n)}.$$

This updating scheme implies a natural alternative flow adjustment mechanism: flows on the non-cheapest alternatives are moved to the least trip cost alternative and the volume moved out from a non-cheapest alternative is proportional to the relative (or scaled) difference between its trip cost and the least trip cost, which is intuitively based on the fact that travelers farther from the equilibrium and on alternatives with larger flow rates are more strongly inclined to change departure time and/or path than those on alternatives with smaller flow rates and with trip cost closer to the minimal cost.

### 6.6.3 Multi-class dynamic network loading (MDNL) using the traffic simulator

By the MSRDUE definition, all trips in a network are equilibrated in terms of actual experienced trip costs, consisting of experienced path times and path costs, so it is necessary to determine the experienced trip costs $G(r)$ for a given multi-class alternative flow vector *r*. To this end, the simulation-based dynamic traffic (network loading) model – DYNASMART (Jayakrishnan et al., 1994; Mahmassani, 2001) is employed to evaluate a given assignment *r* and to obtain $G(r)$ and time-dependent link travel times used in the alterative generation step. DYNASMART adopts a hybrid (mesoscopic) approach to capture the dynamics of vehicular traffic flow in the simulation, whereby vehicles are moved individually according to prevailing local speeds, consistent with macroscopic flow relations on links. It should be noted that the algorithm is independent of the specific dynamic traffic model selected; any particle-based (microscopic or mesoscopic) dynamic traffic model capable of capturing complex traffic flow dynamics can be embedded into the proposed algorithm. When a particle-based dynamic traffic model is employed to determine experienced path times, the path time $TT_{odp}^{\tau}(r)$ for a discrete time interval should be considered as the *average* path time of the vehicles with the same $(o,d,\tau,p)$, because, to respect traffic propagation rules and junction exit capacity constraints, different vehicles embarking along path $p\in P(o,d)$ in departure interval $\tau$ will normally reach their destination *d* at different times and hence experience different trip times. This, in turn, means that the definition of RMC-SRDUE (or MSRDUE) in this study must involve the *average* experienced trip cost.

### 6.6.4 Convergence checking using gap values

Several criteria for convergence checking had been considered in the literature of DTA algorithms. For instance, Peeta and Mahmassani (1995) adopted in their simulation-based DTA model a criterion based on the comparison of path assignments (or path flows) over successive iterations. This study extends the gap-based criterion (or measure) proposed in chapter 3 for the DUE problem to the RMC-SRDUE context and defines the multi-class version of the gap function as the following:

$$Gap(r^l) = \sum_{u(b,\theta,m,n)} \sum_{o} \sum_{d} \sum_{(\tau,p)\in alt_{od}(b,\theta,m,n)} r_{odp}^{\tau,l}(b,\theta,m,n) \times \Delta_{odp}^{\tau,l}(b,\theta,m,n) \qquad (6.27)$$

Note that, $Gap(r^l)$ provides a measure of the violation of the RMC-SRDUE conditions in terms of the difference between the total actual experienced path generalized cost and the total least generalized cost evaluated at any given multi-class path flow pattern *r*. The difference vanishes when the path flow vector $r^*$ satisfies the RMC-SRDUE conditions. In the proposed solution algorithm, for practical considerations, if $|Gap(r^l) - Gap(r^{l-1})| \le \varepsilon$ (a predetermined convergent threshold), convergence is assumed and the program goes back to the outer loop (step 2).

### 6.6.5 Vehicle-based implementation technique

The above MSRDUE model and algorithm are featured as the alternative-based approach, necessitating the explicit storage of the alternative set and the assignment results (i.e. alternative flows) for each user class. Although it is straightforward to record all the alternatives and the corresponding choice probabilities for each user class by using multi-dimensional arrays, computer memory requirements grow dramatically when the number of OD pairs is large, or many iterations are required to achieve convergence. Furthermore, the relaxation to the continuously distributed VOT, VOESD, and VOLSD allows a large number of classes of trips to be in a simultaneous equilibrium, each of which requires its own set of alternatives, and the number of user classes is unknown a priori and changes from iteration to iteration, making it more difficult to construct a memory efficient data structure for storing and updating the huge alternative set and assignment results in network applications with practical size. Essentially, as an attempt to accommodate greater behavioral and policy realism in applying DTA models for designing and evaluating dynamic pricing schemes, modeling heterogeneous users with ranges of VOT, VOESD, and VOLSD as opposed to identical users exacerbates the computational complexity and memory requirement.

In a particle-based and simulation-based DTA system, vehicles carry their departure times and paths from iteration to iteration, which implicitly reflect and store the alternative set and the corresponding assignments results. This is particularly advantageous for large-scale DTA applications, as the total number of feasible alternatives generated by the iterative solution algorithm, after a certain number of iterations, could be significantly greater than the total number of vehicles, which is determined a priori by the OD demand table. For example, in the Portland transportation planning network (Nagel et al., 2000), there are about 1,260 traffic analysis zones (TAZ) and 1.5 million OD pairs, and the total trips are 1.5 millions. Obviously, every OD pair requires more than one alternative for reaching the MSRDUE. Thus, storing the vehicle paths and departure times is more memory-efficient than storing the complete alternative set and routing policies for large-scale networks.

With this vehicle-based implementation technique, the multi-class alternative flow updating scheme presented in Eq.(6.25) and Eq.(6.26) can be interpreted as the following. In iteration $l$, for each user class $u(b,\theta,m,n)$, each OD pair ($o$,$d$) and for each alternative $(\tau, p)$, the number of vehicles moved to the (referenced) least trip cost path is $\rho^l \times r_{odp}^{\tau,l}(b,\theta,m,n) \times \dfrac{\Delta_{odp}^{\tau,l}(b,\theta,m,n)}{G_{odp}^{\tau}(r^l;b,\theta,m,n)}$, and the remaining vehicles would keep their current alternatives. Essentially, this implementation technique uses the vehicle path set (ant the departure times) as a proxy for the exact alternative set and assignment results (routing policies), which can be approximately recovered from the realized vehicle paths in the last iteration's simulation results.

## 6.7 NUMERICAL EXPERIMENTS

A set of numerical experiments is conducted to examine the MSRDUE algorithm. The emphases are (i) to examine the algorithmic convergence property and solution quality of the algorithm and, (ii) with the explicit consideration of user heterogeneity, to investigate how the random parameters (i.e. VOT, VOESD, and VOLSD) in the MSRDUE model would affect departure time and path flow patterns (or toll road usage) under different dynamic pricing scenarios; that is, to compare the differences in departure time and path flow patterns between random parameter model and constant parameter model. The proposed MSRDUE algorithm is implemented by using the aforementioned vehicle-based technique, which can be seamlessly integrated with any mesoscopic/microscopic traffic simulator and is considered particularly appealing for large network deployments of DTA models. The algorithm is coded and compiled by using the Compaq Visual FORTRAN 6.6 and evaluated on the Windows XP platform and a machine with an Intel Pentium IV 2.8 GHz CPU and 2GB RAM.

In all the experiments conducted, the following settings of the random parameters are applied. Note that the unit of VOT, VOESD, and VOLSD in this study is United States dollars (USD) per minute. The continuous distribution of the three parameters is assumed to be the normal distribution specified as follows:

VOT distribution: N(0.4, 0.2), [$\alpha^{min}$,$\alpha^{max}$] = [0.01, 3.0];

VOESD distribution: N(0.3, 0.15), [$\beta^{min}$, $\beta^{max}$] = [0.01, 2.0];

VOLSD distribution: N(1.8, 0.6), [$\lambda^{min}$, $\lambda^{max}$] = [0.25, 4.0].

The VOT distribution is adapted from the estimated measurements in a value pricing experiment conducted in Southern California, USA (e.g. Lam and Small, 2001; Brownstone and Small, 2005), while the distributions of VOESD and VOLSD are determined by economic judgments based on the results reported in Small (1982), due to the lack of estimated values from real world data. The resolution (aggregation interval) of the time-dependent shortest path tree calculation is set to 5 minutes, which is the same as the arrival time interval and the PAT interval. A strict convergence criterion is used in the inner loop of the column generation-based algorithm; that is $|Gap(r^l) - Gap(r^{l-1})|/Gap(r^l) \leq 0.001$. The initial solutions of the experiments are obtained by loading given OD demands with an arbitrary guessing departure time distribution to the (static) extreme efficient paths calculated based on prevailing travel times output from the traffic simulator. Another measure of effectiveness (MOE) is collected in the conducted experiments, in addition to the value of *Gap*(*r*). It is the average gap over all the vehicles in the network for a given alternative flow pattern *r*.

$$AGap(r) = \frac{\sum_{u(b,\theta,m,n)} \sum_{o} \sum_{d} \sum_{(\tau,p)\in alt_{od}(b,\theta,m,n)} r_{odp}^{\tau,l}(b,\theta,m,n) \times \Delta_{odp}^{\tau,l}(b,\theta,m,n)}{\sum_{u(b,\theta,m,n)} \sum_{o} \sum_{d} \sum_{(\tau,p)\in alt_{od}(b,\theta,m,n)} r_{odp}^{\tau,l}(b,\theta,m,n)} \tag{6.28}$$

This MOE is independent of problem size and thus useful for examining the convergence pattern and solution quality of the MSRDUE algorithm on different networks. The minimum of the *AGap*(*r*) is zero. Essentially, the smaller the average gap, the closer the solution is to a BDUE.

Note that this study aims at developing a MSRDUE model for evaluating dynamic pricing scenarios but not solving for a toll vector that improves local or network-wide performance. Hence, testing different dynamic toll vectors in the conducted experiments does not intend to compare their effectiveness on reducing congestion, and focuses exclusively on demonstrating what the MSRDUE model can accomplish and why the user heterogeneity should be addressed in evaluating dynamic road pricing scenarios.

### 6.7.1 Experiment conducted on a two-path test network

This experiment is conducted on a small test network (Figure 6.6), consisting of 2 nodes and 2 links (or 2 paths). Associated with each link are the following attributes: length (miles), number of lanes, free flow speed (miles per hour), and capacity (vehicles per hour per lane). There are two paths connecting the only one OD pair (1, 6): 1→2→3→4→6 and 1→2→5→4→6. There are 14,400 vehicles loaded to this network and the (discrete) PAT distribution of those vehicles is shown in Figure 6.7(a). A toll booth is installed on the entry of link (2→3) so the vehicles choosing path (1→2→3→4→6) have to pay (time-varying) tolls. The time-dependent (or step) pricing scenario applied in this small test network is depicted in Figure 6.7(b). The simulation planning horizon is 150 minutes.

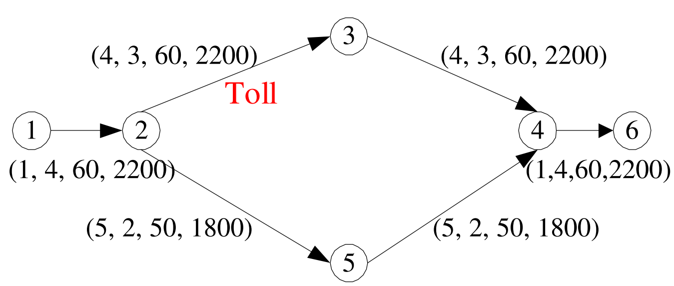
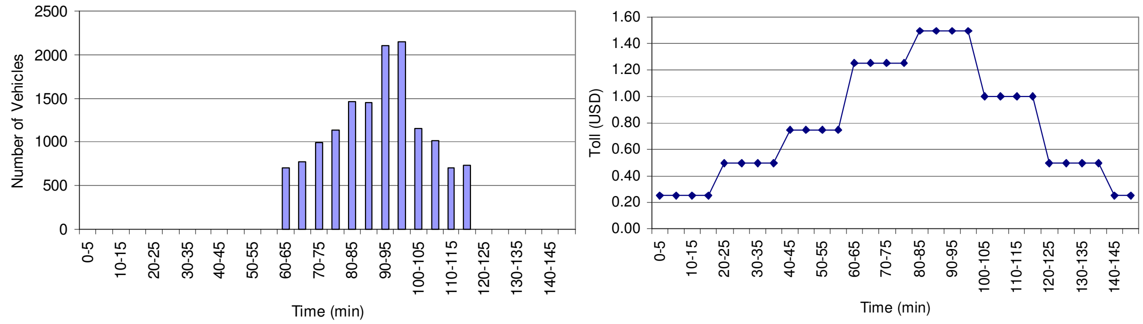



Figure 6.6 The two-path test network




(a) (b)

Figure 6.7 PAT distribution (a) and dynamic pricing scenario (b) in the test network

The convergence pattern of the MSRDUE algorithm in terms of average gap value, defined in Eq.(6.28), is depicted in Figure 6.8. This convergence pattern is compared with that of solving the constant parameter model (by the same algorithm), in which VOT, VOESD, and VOLSD are set to equal to the mean of the corresponding normal distribution assumed in the random parameter model (i.e. VOT = 0.4, VOESD = 0.3, and VOLSD = 1.8). It is shown in the figure that the convergence patterns of the solution algorithm for both models look similar, though the average gap values (and gap values) decrease non-monotonically. Moreover, the solution algorithm is able to find close-to-optimal solutions for both random parameter and constant parameter models, as the final average gap values are very small (around 0.2-0.3 minutes) in both cases. Figure 6.9 shows the numbers of early, late and on-time (punctual arrival) vehicles from iteration to iteration in the random parameter model. As seen in the figure, the number of late vehicles decreases dramatically in the first few iterations, while the numbers of early and on-time vehicles increase steadily iteration by iteration (in reality, trip-makers generally tend to avoid penalties due to late arrival).

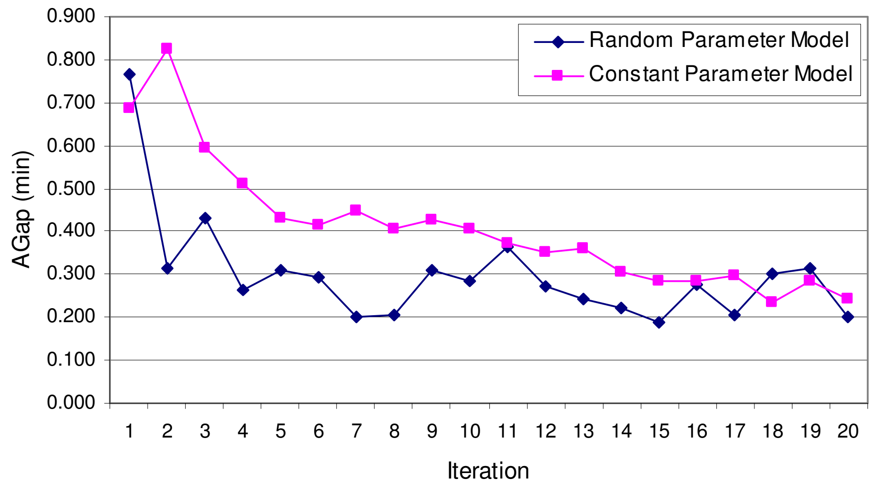



Figure 6.8 Convergence pattern in terms of average gap on the test network

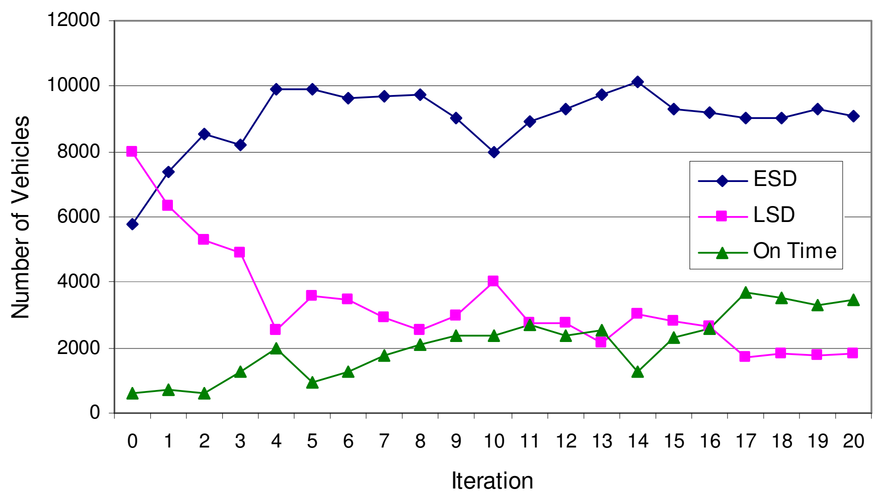



Figure 6.9 Convergence pattern in terms of number of SD vehicles on the test network

The convergence pattern in terms of departure time distribution of the random parameter model is reported in Figure 6.10. Although the departure times in the initial solution, an arbitrary guess, are almost evenly distributed between minutes 30 and 120, the final departure time pattern corresponding to the close-to-MSRDUE solution has an obvious peak between minutes 25 and 40, as trip-makers tend to depart before the toll charge goes high. This indicates that the mechanisms of alternative generation and alternative flow equilibration of the solution algorithm are able to adjust the departure time pattern from disequilibrium to (near-) equilibrium.

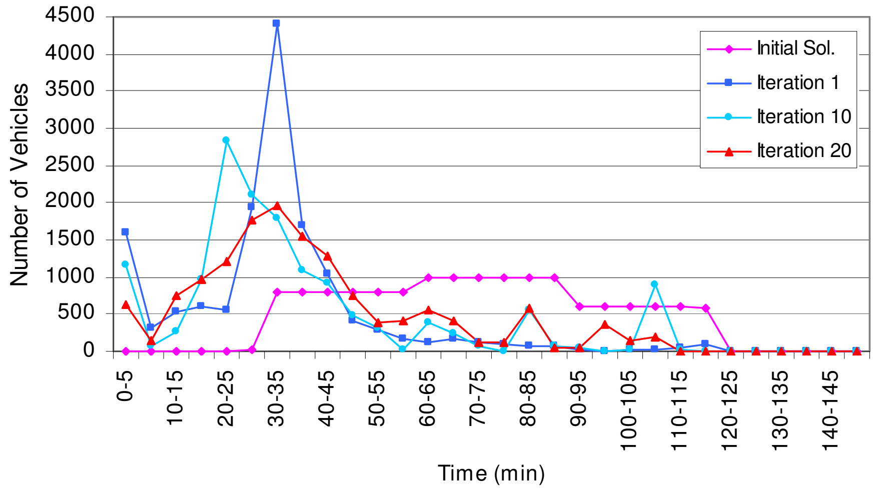



Figure 6.10 Convergence pattern in terms of departure time distribution on the test network

The comparison of departure time patterns in the random parameter model and the constant parameter model is presented in Figure 6.11. While the random parameter model predicts a departure time peak between 25 and 40 minutes, the constant parameter model anticipates a peak between 30 and 45 minutes. Furthermore, the central (peak) tendency of departure times in the constant parameter model (more than 50% of trip-makers would depart in the peak) is higher than that in the random parameter model. In summary, the peak of departure time pattern is higher and happens later in the constant parameter model than that in the random parameter model. Similar observations can be found in the comparison of time-varying toll road usage (defined as the number of vehicles departing at each time interval and using the toll road) in the two models (Figure 6.12). The constant parameter model also predicts a slight higher toll road usage than the random parameter model (10716 versus 10273). This phenomenon is resulted from the constant

VOT, VOESD, and VOLSD assumed in the model and trip-makers behave identically in choosing departure times and paths; while the random parameter model explicitly considers heterogeneous users with different VOT, VOESD, and VOLSD.

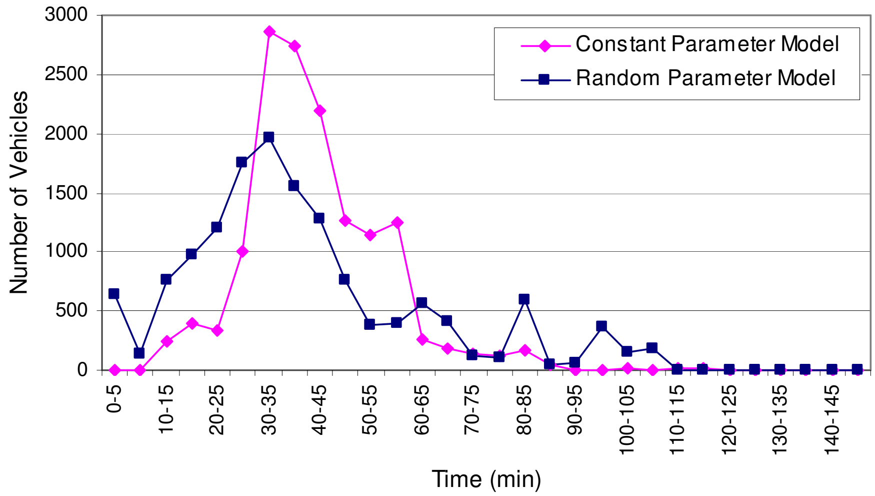



Figure 6.11 Comparison of departure patterns in constant and random parameter models on the test network

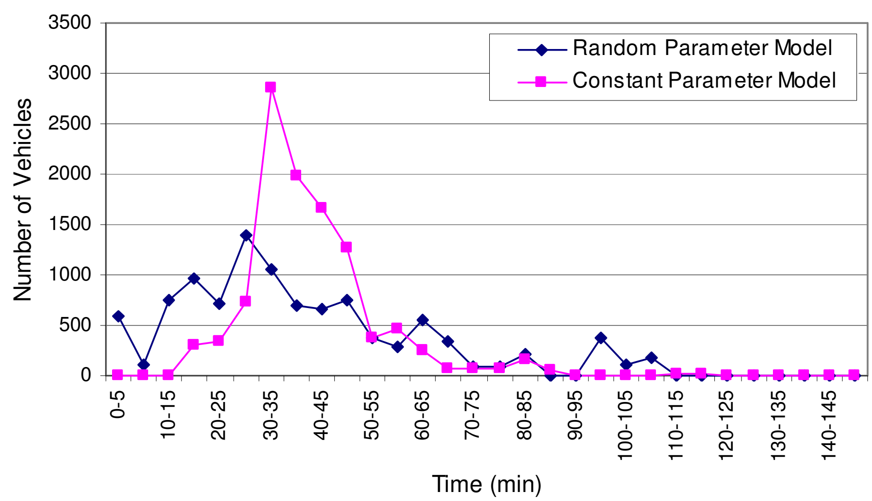
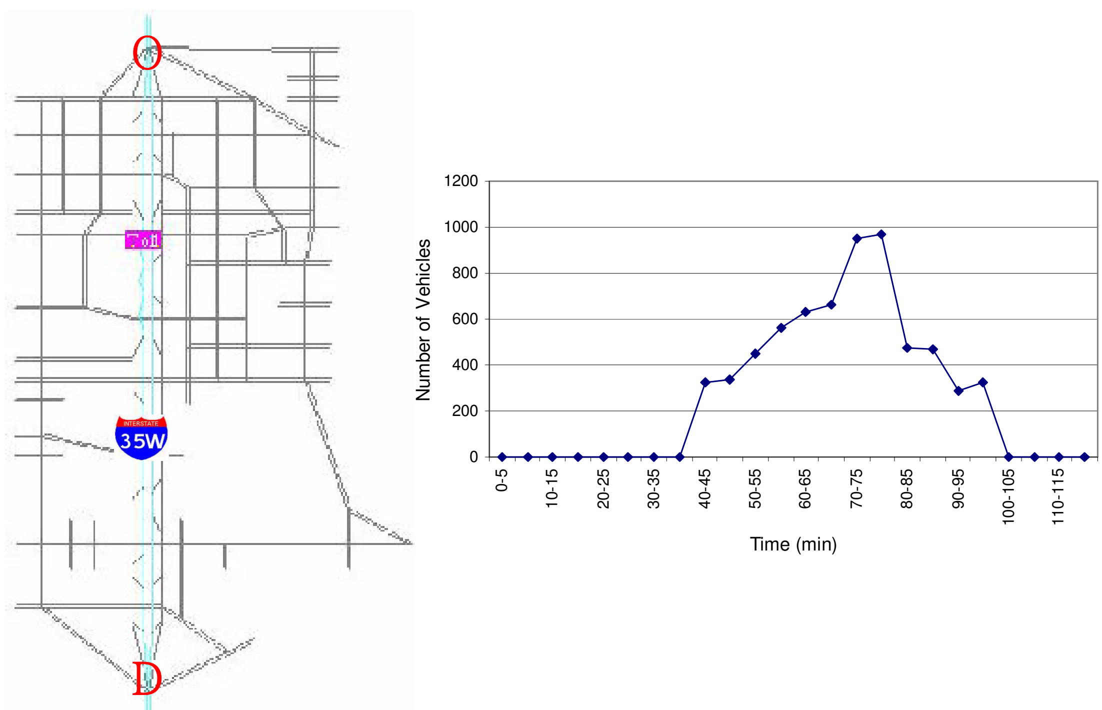



Figure 6.12 Comparison of time-varying toll road usage in constant and random parameter models on the test network

### 6.7.2 Experiment conducted on the Fort Worth network

The Fort Worth (Texas, USA) network (Figure 6.13(a)), consisting of 180 nodes (62 signalized nodes), 445 links and 13 traffic analysis zones (TAZ), is used in this experiment. There are 25,500 vehicles loaded to this network. A critical OD pair (zone 1 to zone 2) is selected to examine the departure time and path flow patterns. This critical OD pair accounts for 25% (6375/25500) of the total demands and the PAT distribution of those vehicles is shown in Figure 6.13(b). To create the hypothetical pricing scenario, a toll road is added to the southbound freeway (I35W) corridor. The toll road is 3 miles long, while the general purpose road (i.e. original non-tolled freeway) is 4.5 miles long. Both roads have three lanes and the same performance function (e.g. capacity and speed limit). Table 6.1 lists the three dynamic pricing scenarios (i.e. low, mid, and high) tested in this experiment. The simulation planning horizon is 150 minutes.




(a) (b)

Figure 6.13 Fort Worth network with hypothetic toll links and PAT pattern

Table 6.1 Dynamic road pricing scenarios tested on the Fort Worth network

| Pricing Scenario | 0-20 minutes | 20-40 minutes | 40-60 minutes | 60-80 minutes | 80-100 minutes | 100-120 minutes | 120-150 minutes |
|---|---|---|---|---|---|---|---|
| #1 (low) | $0.05 | $0.20 | $0.35 | $0.50 | $0.35 | $0.20 | $0.05 |
| #2 (mid) | $0.25 | $0.40 | $0.55 | $0.70 | $0.55 | $0.40 | $0.25 |
| #3 (high) | $0.45 | $0.60 | $0.75 | $0.90 | $0.75 | $0.60 | $0.45 |

We first examine the algorithmic convergence behavior of the MSRDUE algorithm in terms of average gap, number of schedule delay (SD) vehicles (early, late, and on-time vehicles), and departure time distributions, under dynamic pricing scenario #2. The convergence pattern in terms of average gap, defined in Eq.(6.28), is depicted in Figure 6.14. This convergence pattern is compared with that of solving the constant parameter model (by the same algorithm), in which the parameters are set to equal to the mean of the corresponding normal distribution assumed in the random parameter model (i.e. VOT = 0.4, VOESD = 0.3, and VOLSD = 1.8). It is shown in the figure that the convergence patterns of the solution algorithm for both models look similar, though the average gap values decrease non-monotonically. Moreover, the solution algorithm is able to find close-to-optimal solutions for both random parameter and constant parameter models as the final average gap values are very small (0.2-0.3 minutes) in both cases. Figure 6.15 shows convergence pattern in terms of the number of SD vehicles of the critical OD pair in the random parameter model. As seen in the figure, the numbers of early and late vehicles decrease dramatically in the first few iterations, while the number of on-time vehicles increases steadily iteration by iteration (in reality, trip-makers generally tend to avoid penalties due to early or late arrival). The convergence pattern in terms of departure time distribution of the random parameter model is reported in Figure 6.16. Although the departure times in the initial solution, an arbitrary guess, are evenly distributed between minutes 0 and 90, the departure time pattern corresponding to the close-to-MSRDUE solution has an obvious peak between minutes 40 and 65, as trip-makers tend to depart before the toll charge goes high. This indicates that the mechanisms of alternative generation and flow equilibration of the algorithm are able to adjust the departure time pattern from disequilibrium to (near-) equilibrium.

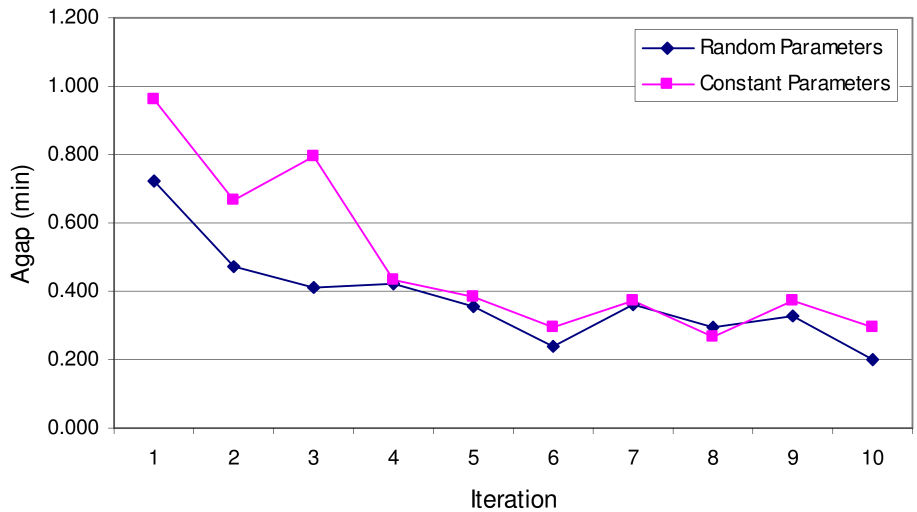



Figure 6.14 Convergence pattern in terms of average gap on the Fort Worth network

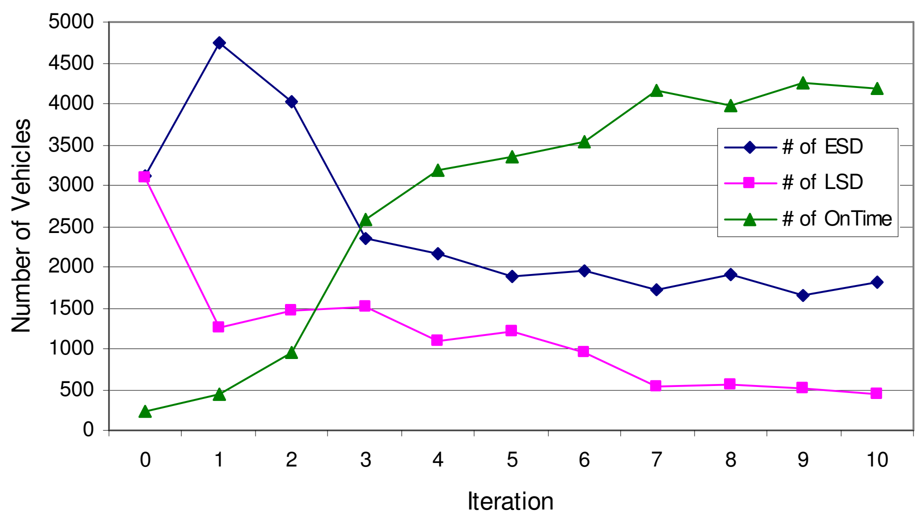



Figure 6.15 Convergence pattern in terms of number of SD vehicles of a critical OD pair on the Fort Worth network

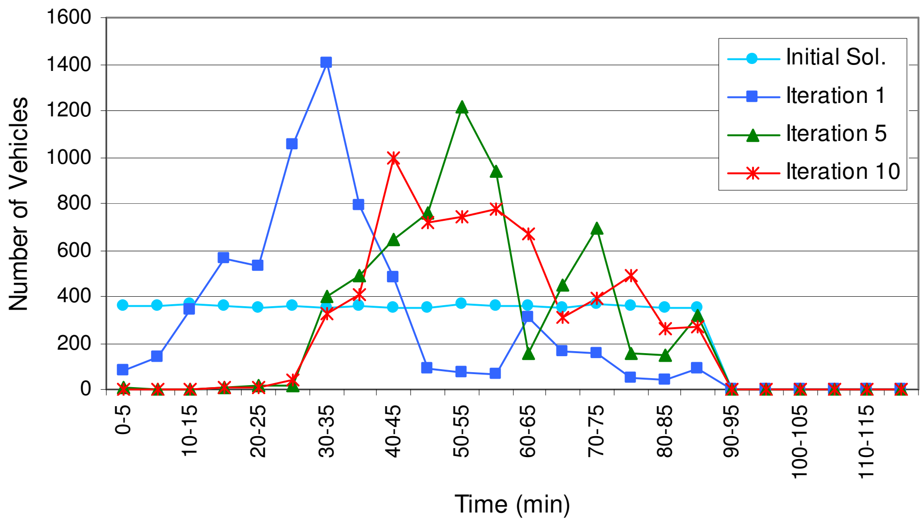



Figure 6.16 Convergence pattern in terms of departure time distribution of a critical OD pair on the Fort Worth network

The comparison of the critical OD pair's departure time patterns in the random parameter model and the constant parameter model is presented in Figure 6.17. While the random parameter model predicts a departure time peak between 40 and 65 minutes, the constant parameter model anticipates a peak between 50 and 65 minutes. Furthermore, the central (peak) tendency of departure times in the constant parameter model is higher than that in the random parameter model. In summary, the peak of departure time pattern is higher and happens later in the constant parameter model than that in the random parameter model. Similar observations can be found in the comparison of time-varying toll road usage (defined as the number of vehicles departing at each time interval and using the toll road) in the two models (Figure 6.18). The constant parameter model also predicts higher toll road usage for this critical OD pair than the random parameter model (2991 versus 2436). These phenomena are resulted from the constant VOT, VOESD, and VOLSD assumed in the model and trip-makers behave identically in choosing departure times and paths; while the random parameter model explicitly considers heterogeneous users with different VOT, VOESD, and VOLSD.

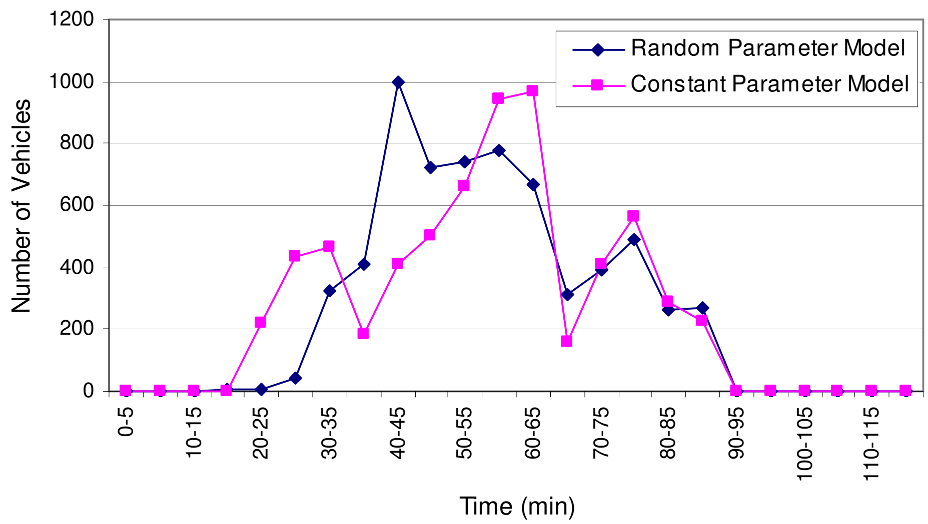



Figure 6.17 Comparison of departure time patterns in constant and random parameter models on the Fort Worth network

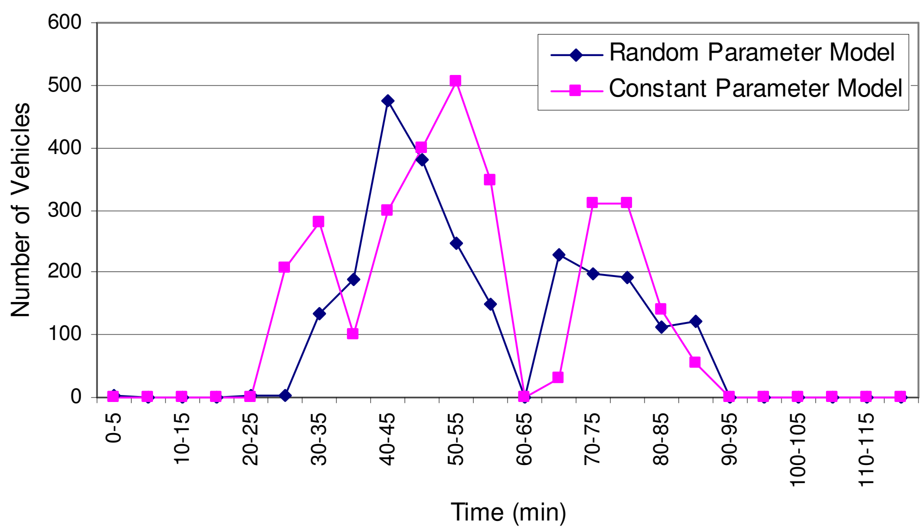



Figure 6.18 Comparison of time-varying toll road usage in constant and random parameter models on the Fort Worth network

The comparison of departure time patterns of that critical OD pair in the random parameter model under the three different dynamic pricing scenarios (or levels) is presented in Figure 6.19. As expected, when the toll charge is high, the departure time pattern shifts leftward, since the majority of trip-makers tends to depart earlier to avoid high tolls. On the other hand, the departure time pattern shifts rightward in the low price case, as most trip-makers are willing to use the cheap toll road to save travel time. These observations are also found in the comparison of time-varying toll road usage of a critical OD pair in the random parameter model under different dynamic pricing scenarios depicted in Figure 6.20. The numbers of vehicles using the toll road are 4323, 2436, and 1921 for the low price, mid price, and high price cases, respectively. Additionally, the peak of toll road usage is shifted to the time period between 20 – 45 minutes in the high price case, far earlier than the mid price and low price cases. These comparisons demonstrate that the proposed MSRDUE model and solution algorithm can effectively describe trip-makers' responses to time-varying toll charges in temporal distribution (departure times) and spatial splits (path flows).

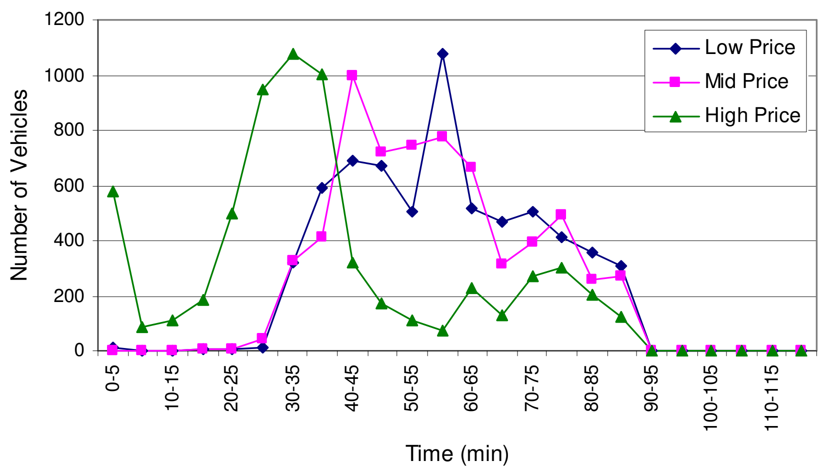



Figure 6.19 Comparison of departure time pattern in the random parameter model under different dynamic pricing scenarios

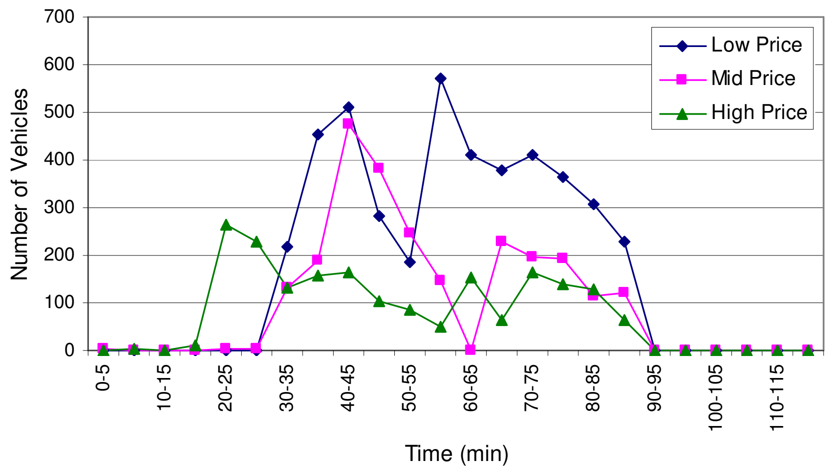



Figure 6.20 Comparison of time-varying toll road usage in the random parameter model under different dynamic pricing scenarios

## 6.8 Summary

This chapter presents the model and solution algorithm for the MSRDUE problem, which explicitly considers heterogeneous trip-makers with different PAT, VOT, VOESD, and VOLSD simultaneously choosing alternatives, combinations of departure times (or arrival times) and paths, that minimize individual's trip cost, a weighted sum of travel time, out-of-pocket cost, and schedule delay cost. The MSDUE problem is formulated as an infinite dimensional VI problem, and solved by the column generation-based algorithmic framework which embeds (i) the alternative generation algorithm – SPAM to obtain the VOT, VOESD, and VOLSD breakpoints that define multiple user classes, and determine the least trip cost alternative for each user class, (ii) the traffic simulator - DYANSMART to capture traffic dynamics and determine experienced path travel times; and (iii) the multi-class path flow equilibrating scheme to solve the RMC-SRDUE problem defined by a subset of feasible alternatives. To circumvent the difficulty of storing the memory-intensive alternative set and assignment results for large-scale network applications, the vehicle-based implementation technique, using the vehicle alternative set as a proxy for keeping track of assignment results, is applied.

The experimental results show that the convergence patterns of the solution algorithm look similar for both the random parameter model and the constant parameter model, and the solution algorithm is able to find close-to-MSRDUE solutions as the final average gap values are very small. Although departure time patterns in the initial solutions are arbitrary guesses, the results also show that the mechanisms of alternative generation and alternative flow equilibration of the solution algorithm are able to adjust the departure time pattern from disequilibrium to (near-) equilibrium. There are significant differences in the estimated/predicted departure time pattern and toll road usage between the two models. The reason is that trip-makers behave identically in choosing departure times and paths in the constant parameter model, while the random parameter model explicitly considers heterogeneous users with different VOT, VOESD, and VOLSD. The comparisons of departure time pattern and toll road usage of a critical OD pair in the random parameter model under the three different dynamic pricing scenarios (or levels) demonstrate that the proposed MSRDUE model and solution algorithm can realistically describe trip-makers' responses to time-varying toll charges in temporal distribution (departure times) and spatial splits (path flows).

# Chapter 7 Conclusions and Future Research

## 7.1 Summary of Contributions and Findings

This section summarizes the contributions and findings of this dissertation.

### 7.1.1 Address heterogeneous users' responses to time-varying toll charges

An essential task of developing DTA (or DUE) models for dynamic road pricing applications is to explicitly recognize and represent heterogeneous users in modeling users' response to time-varying toll charges. Although this critical issue of user heterogeneity has been considered in the literature (see section 2.2), all those network equilibrium assignment models were developed only for flat (static) road pricing schemes, rather than dynamic (or time-dependent) ones. In fact, successful design and evaluation of dynamic pricing schemes relies on a realistic representation of complex traffic dynamics and spatial and temporal vehicular interactions in traffic assignment models, hence necessitating the extension of the heterogeneous traffic assignment model from the static regime to the DTA context.

#### 7.1.1.1 The BDUE traffic assignment model and algorithm

This dissertation presents the bi-criterion dynamic user equilibrium (BDUE) traffic assignment model which explicitly considers heterogeneous trip-makers with different VOT choosing paths that simultaneously optimize the two essential path choice criteria: travel time and out-of-pocket cost. To realistically capture trip-makers' path choice decisions in response to toll charges, in the underlying path choice model, each trip-maker is assumed to select the least generalized cost path, the generalized cost being the sum of travel (out-of-pocket) cost and travel time weighted by that trip-maker's VOT. The VOT is assumed to be continuously distributed among trip-makers. Additionally, the time-dependent OD demands for the entire feasible range of VOT over the planning horizon are also assumed known, a priori. The goal is to obtain a time-varying path flow vector satisfying the BDUE conditions, under a given dynamic road pricing scheme.

The BDUE problem is formulated as an infinite dimensional variational inequality (VI), and solved by the column generation-based algorithmic framework which embeds (i) the extreme non-dominated path finding algorithm – PAM (parametric analysis method) to obtain the breakpoints which partition the entire range of VOT into many subintervals and determine the multiple user classes, and find the least generalized cost path for each user class, (ii) the traffic simulator – DYANSMART (Jayakrishnan, et al. 1994) to capture traffic dynamics and determine experienced path travel times for any given path flow pattern; and (iii) the multi-class path flow updating/equilibrating scheme to solve the restricted multi-class dynamic user equilibrium (RMDUE) problem defined by a subset of feasible paths.

The experimental results show that the convergence pattern of the proposed BDUE algorithm is not affected by the different VOT assumptions, and it is able to find close-to-BDUE solutions. Moreover, when the toll level is increased, the decreasing of the toll road usage for the constant and discrete VOT cases is more dramatic than that for the normal distribution VOT case. Using the random parameter model with a normal VOT distribution as a benchmark, the constant VOT model overestimates the toll road usage when the toll charge is low and underestimates the toll road usage when the toll charge is high. The impact of estimation biases in terms of the toll road usage is also reflected in the overall network performance, in terms of average trip time. The experimental results also provide toll operators useful information: when the toll level changes, users' reactions are not as dramatic as what had been predicted by DTA models with the single constant VOT assumption.

7.1.1.2 The MDUE traffic assignment model and algorithm

The multi-criterion DUE (MDUE) traffic assignment model is a direct extension of the BDUE model by explicitly considering an important path choice criterion – travel time variability in trip-makers' path choice decision framework, in addition to travel time and out-of-pocket cost, and allowing not only the VOT but also the VOR the be continuously distributed across all trip-makers in a network. The travel time variability of a path in a departure time interval is defined as the variance (or standard deviation) of experienced path travel times of vehicles entering that path in that departure time interval. Each trip-maker is assumed to choose a path that minimizes the three essential path choice criteria: out-of-pocket cost (e.g. toll), travel time, and travel time variability. By following the modeling framework typically adopted in discrete time, deterministic DUE models for describing trip-makers' path choice behavior, the (experienced) path generalized cost is defined as the sum of travel cost, travel time weighted by the value of time (VOT) and travel time variability weighted by the value of reliability (VOR). The goal is to obtain a time-varying path flow vector satisfying the MDUE conditions, under a given dynamic road pricing scheme.

Specifically, the multi-criterion dynamic user equilibrium (MDUE) problem is formulated as an infinite dimensional variational inequality (VI), and solved by a column generation-based solution algorithm, which embeds (i) the sequential parametric analysis method (SPAM) to obtain the set of time-dependent extreme efficient (or non-dominated) paths and the corresponding breakpoint vectors of VOT and VOR that naturally define the multiple user classes, each of which corresponds to particular ranges of VOT and VOR, (ii) the traffic simulator – DYANSMART (Jayakrishnan, et al. 1994) to capture traffic dynamics and determine experienced path travel times and their travel time standard deviations for any given path flow pattern, and (iii) the multi-class path flow updating scheme to solve the restricted multi-class dynamic user equilibrium (RMDUE) problem defined by a subset of time-dependent extreme efficient paths.

7.1.1.3 The MSRDUE traffic assignment model and algorithm

In general, a trip-maker facing a toll road with time-varying charges would not only change path (or route) but also adjust departure time so as to minimize his/her total trip cost, so a realistic generalization of the BDUE problem is to allow trip-makers to make departure time choices, in addition to path choices, in response to time-varying toll charges. This dissertation develops the model and algorithm for solving the multi-criterion simultaneous route and departure time user equilibrium (MSRDUE) problem, which explicitly considers heterogeneous trips (or trip-makers) with different VOT, VOESD, and VOLSD simultaneously choosing departure times and paths that minimize the set of trip attributes: travel time, out-of-pocket cost, and schedule delay cost. Classical discrete time, deterministic SRDUE models typically assume each trip-maker chooses the alternative, a combination of departure time (interval) and path, which minimizes his/her trip cost, defined as the sum of travel cost, travel time weighted by VOT, and early or late schedule delay weighted by VOESD or VOLSD. In stead of modeling trip-makers' departure time decisions, this study considers them making arrival time choices and develops an algorithm for computing time-dependent least cost paths for all possible arrival time intervals. This modeling approach would facilitate finding the least trip cost path(s), because, given all possible (early or late) schedule delays, the least trip cost path can be found by computing the least generalized cost paths for all possible arrival time intervals. Note that once the best arrival time interval (and the path associated with it) is selected, the corresponding departure time can be readily determined by subtracting the path travel time from that arrival time (interval). Therefore, modeling trip-makers' selections of arrival time interval is equivalent to modeling their departure time choices.

The MSDUE problem is formulated as an infinite dimensional variational inequality (VI) problem, and solved by the column generation-based algorithmic framework which embeds (i) the (extreme non-dominated) alternative finding algorithm – SPAM (sequential parametric analysis method) to obtain the VOT, VOESD, and VOLSD breakpoints that define multiple user classes, and determine the least trip cost alternative for each user class, (ii) the traffic simulator – DYANSMART (Jayakrishnan, et al. 1994) to capture traffic dynamics and determine experienced travel times; and (iii) the multi-class path flow equilibrating scheme to solve the restricted multi-class SRDUE (RMC-SRDUE) problem defined by a subset of feasible alternatives. Although the mathematical abstraction of the problem is a typical analytical formulation, this study adopts the simulation-based approach to tackle many practical aspects of the DTA applications.

The experimental results show that the convergence patterns of the solution algorithm look similar for both the random parameter model and the constant parameter model, and the solution algorithm is able to find close-to-MSRDUE solutions as the final average gap values are very small. There are significant differences in the estimated/predicted departure time pattern and toll road usage between the two models. The reason is that trip-makers behave identically in choosing departure times and paths in the constant parameter model, while the random parameter model explicitly considers heterogeneous users with different VOT, VOESD, and VOLSD. The comparisons of departure time pattern and toll road usage of a critical OD pair in the random parameter model under the three different dynamic pricing scenarios (or levels) demonstrate that the proposed MSRDUE model and solution algorithm can effectively describe trip-makers' responses to time-varying toll charges in departure times and path flows.

### 7.1.2 Improve the simulation-based DUE approach

The simulation-based DUE approach provides considerable modeling flexibility (e.g. of traffic control measures and information supply strategies) for a wide range of engineering applications because it describes traffic flow propagation, captures spatial and temporal vehicular interactions, and determines link and path travel costs through traffic simulation instead of analytical evaluation. However, using traffic simulation to reflect the properties of the actual underlying real traffic systems, which are generally not well-behaved mathematically, often precludes guaranteed algorithmic convergence and solution optimality (i.e. adherence to the DUE conditions). Based on the strength of the simulation-based approach in adequately capturing traffic flow dynamics, such as queue build-up, spillback and dissipation in congested networks, this study develops a theoretically sound simulation-based DUE model that is capable of realistically capturing traffic dynamics while adhering to the DUE conditions, as well as providing the basis for an algorithm that exhibits better performance (solution quality and computational effort) than commonly used averaging schemes (e.g. the method of successive averages, MSA) on practical networks.

While it has been generally modeled as the VI or nonlinear complementarity problem (NCP) in the literature, the DUE problem is reformulated in this study, via a gap function, as a nonlinear minimization problem (NMP) whose global solution(s) coincides with solutions of the VI problem that satisfies the DUE conditions. This gap function provides a measure of the violation of the DUE conditions in terms of the difference between the total actual experienced path travel cost and the total shortest path cost evaluated at a given feasible time-varying path flow pattern. The difference vanishes when the time-varying path flow vector satisfies the DUE conditions. Thus, solving the DUE problem can be viewed as a process of finding the optimal path flow vector such that the value of the gap function equals zero.

This reformulation is then solved by a column generation-based DUE algorithmic framework, which embeds (i) a simulation-based dynamic traffic (or network loading) model to capture traffic dynamics as well as to determine experienced path costs for any given path flow pattern and (ii) a descent direction method to solve the restricted NMP defined by a subset of feasible paths. The descent direction method has the following important features. First, it applies a scaling approach, in the same manner as the inverse of second order derivatives used in Newton-type methods, to determine appropriate step sizes. The scaling approach, which normalizes path cost differences between non-shortest paths and the shortest paths, also overcomes the deficiency of using absolute path cost differences in updating path assignments. Second, to be applicable in simulation-based DTA models as well as large-scale network problems, the proposed descent direction method does not require computing the gradient of the objective function. As a result, the underlying path (or link) cost functions need not be differentiable. Last, in order to mitigate the impact of possible oscillations and speed up convergence, this method is further integrated with a mixed step size scheme and an active constraint strategy in the column generation solution framework. Note that this descent direction method forms the basis of the multi-class path/alternative flow updating (or equilibrating) scheme in the column generation-based BDUE and MSRDUE solution algorithms.

In summary, the column generation technique is able to avoid explicitly enumerating all feasible paths, and the descent direction method can circumvent the need for computing partial derivatives in estimating the gradient of the objective function. The adoption and integration of the above two methods, coupled with the embedded traffic simulator, could enhance the development and deployment of simulation-based DTA models. Computational results on both small and large real road networks demonstrate that the proposed DUE algorithm is more efficient and effective in obtaining close-to-DUE solutions than the commonly used MSA. ). Moreover, the algorithm is independent of the specific traffic simulator selected; any (macroscopic, microscopic or mesoscopic) dynamic traffic model capable of capturing complex traffic flow dynamics, in particular the effect of physical queuing, as well as preventing violations of the first-in-first-out property, can be embedded into the proposed solution algorithm. The experimental results also suggest the RNMP does not have to be solved optimally in each iteration, in order to improve the overall computational efficiency and achieve satisfactory convergence, as indicated in early studies on the diagonalization algorithm for asymmetric traffic assignment problems (see e.g. Mahmassani and Mouskos, 1988).

### 7.1.3 Enable practical deployments for large scale network applications

When the traffic assignment problem is extended from the static regime to the DTA context, researchers have shown greater interest in the path flow-based formulation that seeks a time-varying path flow vector satisfying the DUE conditions than in the link flow-based formulation, due to the recent advancement and deployment of intelligent transportation systems (ITS), in particular the route guidance information systems. The proposed DUE, BDUE and MSRDUE models and algorithms are featured as the path/alternative-based approach, necessitating the explicit storage of the path/alternative set and the assignment results (i.e. path/alternative flows) for each user class. Although it is straightforward to record all the paths/alternatives and the corresponding choice probabilities for each user class by using multi-dimensional arrays, computer memory requirements grow dramatically when the number of OD pairs is large, or many iterations are required to achieve convergence. Furthermore, the relaxation to the continuously distributed VOT, VOESD, and VOLSD allows a large number of classes of trips to be in a simultaneous equilibrium, each of which requires its own set of paths/alternatives, and the number of user classes is unknown a priori and changes from iteration to iteration, making it more difficult to construct a memory efficient data structure for storing and updating the huge path/alternative set and assignment results in network applications with practical size.

To circumvent the difficulty of storing the memory-intensive path/alternative set and routing policies for large-scale network applications, a vehicle-based implementation technique using the realized vehicle path set as a proxy for keeping track of the path/alternative assignment results is proposed in this dissertation. In a particle-based and simulation-based DTA system, vehicles carry their departure times and paths from iteration to iteration, which implicitly reflect and store the path/alternative set and the corresponding assignments results. This is particularly advantageous for large-scale DTA applications, as the total number of feasible path/alternatives generated by the iterative solution algorithm, after a certain number of iterations, could be significantly greater than the total number of vehicles, which is determined a priori by the OD demand table. Essentially, this implementation technique uses the vehicle path set (ant the departure times) as a proxy for the exact alternative set and assignment results (routing policies), which can be approximately recovered from the realized vehicle paths in the last iteration's simulation results. The experiments conducted on large scale road networks (e.g. the Irvine and CHART networks; see section 4.7.2) show that this vehicle-based implementation technique requires much less computer memory than the typical multi-dimensional grand path/alternative set implementation method.

## 7.2 Future Research and Extensions

This section outlines several directions of future research and extensions of this dissertation.

(1) Extensions to OD-specific and time-varying VOT, VOESD, and VOLSD distributions

The BDUE and MSRDUE models proposed in this dissertation apply the *same* continuous (normal) VOT, VOESD, and VOLSD distributions (with the same parameters – mean and variance) for all trip-makers in a network, regardless of their origins, destinations, and/or departure times. However, these distributions and their parameters may vary with different geographic locations and time of day, and assuming the same distributions across all trip-makers in a network could end up with missing some useful information and lead to a certain extent of under/over-estimations of trip-makers' responses to toll charges. Therefore, the natural extensions of the models would be to consider these distributions be OD-specific and time-varying. Moreover, the (discrete/continuous) PAT distribution could be also extended to be varying with time of day and, more likely, with different destinations.

(2) Development of re-optimization algorithms for the PAM and SPAM

In the current implementation of the extreme non-dominated path-finding algorithm PAM and alternative-finding algorithm SPAM, when a new upper/lower bound of VOT is found by the parametric analysis, the link generalized costs are updated with this new VOT, and the corresponding least generalized cost path tree is computed from scratch. Thus, the computational efforts of applying the PAM and SPAM to find the set of extreme efficient paths/alternatives highly depends on the computational efficiency of computing a least generalized cost path tree. Recognizing that two successive/ neighboring trees only differ in one arc (actually an arc-time combination), due to the nature of the parametric analysis, a future study would be to develop re-optimization algorithms for improving the computation efficiency of the PAM and SPAM.

(3) Applications to dynamic congestion pricing problems

In the literature, the problem of determining tolls to reduce congestion is often referred to as a congestion pricing problem. This problem can be generally classified into two categories: first-best and second-best congestion problem; the former assumes that ever road arc in a network can be tolled, while the later considers that some road arcs are not tollable. Among many mathematical models of the congestion pricing problem, the bi-level programming model (or its special case – mathematical programming with equilibrium constraints) is the most popular method of formulating the problem. Since the lower level problem in the bi-level programming model is typically a network equilibrium assignment problem aiming to determine users' responses to the tolls obtained by solving the upper level (toll design) problem, it is necessary to apply UE traffic assignment algorithms to solve the lower level problem. To the author's current knowledge, very few past studies on congestion pricing problems have considered user heterogeneity (in terms of VOT, VOESD, and VOLSD) in solving lower level UE assignment problem, and this may invalidate the tolls found by solving the bi-level programming model, as users' responses to toll charges are not adequately captured. Thus, one promising future research is to explicitly consider user heterogeneity and apply the BDUE and the MSRDUE models in solving the congestion pricing problem.

# References


Abdel-Aty, M. A., Kitamura, R. and Jovanis, P. P. (1997). Using stated preference data for studying the effect of advanced traffic information on drivers' route choice. Transportation Research Part C, Vol. 5 (1), pp. 39-50.

Abdelghany, K. F. (2001). Stochastic dynamic traffic assignment for intermodal transportation networks with consistent information supply strategies. Ph.D Dissertation, the University of Texas at Austin.

Algers, S., Bergstrom, P., Dahlberg, M., and Dillen, J. L. (1998). Mixed logit estimation of the value of travel time. Working paper, Department of Economics, Uppsala University, Uppsala, Sweden.

Arnott, R., de Palma, A. and Lindsey, R. (1990). Departure time and route choice for the morning commute. Transportation Research Part B, Vol. 24 (3), pp. 209-228, 1990.

Bazara'a, M., Sherali, H., and Shetty, C. (1993). Nonlinear Programming: Theory and Algorithms, second edition. John Wiley & Sons, New York.

Beckmann, M. J., McGuire, C. B., and Winsten, C. B. (1956). Studies in the Economics of Transportation. Yale University Press, New Haven, Connecticut.

Ben-Akiva, M. E., Lerman, S. (1985). Discrete Choice Analysis. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA.

Ben-Akiva, M., De Palma, A., and Kanaroglou, P. (1986). Dynamic model of peak period traffic congestion with elastic arrival rates. Transportation Science, Vol. 20 (2), pp. 164-181.

Ben-Akiva, M. E., Bolduc, M., D., and Bradley, M. (1993). Estimation of travel choice models with randomly distributed values of time. Transportation Research Record, 1413, pp. 88-97.

Ben-Akiva, M. E., Bierlaire, M., Bottom, J., Koutsopoulos, H. N., and Mishalani, R. (1997). Development of a route guidance generation system for real-time applications. Proceedings of the 8th International Federation of Automatic Control Symposium on Transportation Systems, Chania, Greece, pp. 433-438.

Bernstein, D., Friesz, T. L., Tobin, R. L., and Wie, B.-W. (1993). A variational control formulation of the simultaneous route and departure-time choice equilibrium problem. Proceedings of the 12th International Symposium on Transportation and Traffic Theory, Daganzo, C. F. (Ed.) Elsevier, USA, pp. 107-126.

Bertsekas, D. P. and Gafni, E. M. (1982). Projection methods for variational inequalities with application to the traffic assignment problem. Mathematical Programming Study, Vol. 17, pp. 139-159.

Bertsekas, D. P. and Gafni, E. M. (1983). Projected Newton methods and optimization of multi-commodity flows. IEEE Transactions on Automatic Control, Vol.28 (12), pp. 1000-1006.

Bertsekas, D. P. (1995). Nonlinear Programming. Athena Scientific, Belmont, MA, USA.

Boyce, D. E., Ran, B. and Li, Y. (1998). Considering travelers' risk-taking behavior in dynamic traffic assignment. In: Bell, M. G. H. (Ed.), Transportation Networks: Recent Methodological Advances. Elsevier, Oxford, pp. 67-81.

Boyce, D., Mahmassani, H. S., and Nagurney, A. (2005). A retrospective on Beckmann, McGuire and Winsten's studies in the economics of transportation. Papers in Regional Science, Vol. 84 (1), pp. 85-103.

Brownstone, D., and Train, K. (1999). Forecasting new product penetration with flexible substitution patterns. Journal of Econometrics, Vol. 89 (1-2), pp. 109-129.

Brownstone, D., and Small, K. A. (2005). Valuing time and reliability: assessing the evidence from road pricing demonstrations. Transportation Research Part A, Vol. 39 (4), pp. 279-293.

Brumbaugh-Smith, J, and Shier, D. (1989). An empirical investigation of some bicriterion-shortest path algorithms. European Journal of Operations Research, Vol. 43 (2), pp. 216-224.

Cantarella, G. E. and Binetti, M. (1998). Stochastic equilibrium traffic assignment with value-of-time distributed among users. International Transactions in Operational Research, Vol. 5 (6), pp. 541-553.

Carey, M. (1992). Nonconvexity of the dynamic traffic assignment problem. Transportation Research Part B, Vol. 26 (2), pp. 127-133.

Chen, H. -K. and Hsueh, C. -F. (1998). A model and an algorithm for the dynamic user-optimal route choice problem. Transportation Research Part B, Vol. 32 (3), pp. 219-234.

Cirillo, C., and Axhausen, K. W. (2006). Evidence on the distribution of values of travel time savings from a six-week diary. Transportation Research Part A, Vol. 40 (5), pp. 444-457.

Climaco, J.C.N., and Martin, E. Q. V. (1982). A bicriterion shortest path algorithm. European Journal of Operation Research, Vol. 11 (4), pp. 399-404.

Cybis, H. B. B. (1995). A dynamic user equilibrium assignment model – a rigorous formulation. In: Gartner, N. H., Improta, G. (Eds.) Urban Traffic Networks: Dynamic Flow Modeling and Control, Springer, Berlin, pp. 233-250.

Dafermos, S. C. (1968). Traffic assignment and resource allocation problem in transportation networks. Ph.D thesis, Johns Hopkins University, Baltimore, Maryland.

Dafermos, S. C. and Sparrow, F. T. (1969). The traffic assignment problem for a general network. Journal of Research of the National Bureau of Standards, 73B, pp. 91-118.

Dafermos, S. C. (1980). Traffic equilibrium and variational inequalities. Transportation Science, Vol. 14 (1), pp. 42-54.

Dafermos, S. C. (1981). A multicriteria route-mode choice traffic equilibrium model. Unpublished manuscript, Lefschetz Center for Dynamical Systems, Brown University, Rhode Island.

Daganzo, C. F. (1979). Multinomial probit: the theory and its application to demand forecasting, Academic Press, New York, USA.

Daganzo, C. F. (1994). The cell transmission model: a simple dynamic representation of highway traffic. Transportation Research Part B, Vol. 28 (4), pp. 269-287.

Daganzo, C. F. (1995a). The cell transmission model part II: network traffic. Transportation Research Part B, Vol. 29 (2), pp. 79-93.

Daganzo, C. F. (1995b). Properties of link travel time functions under dynamic loads. Transportation Research Part B, Vol. 29 (2), pp. 95-98.

De Palma, A., Ben-Akiva, M., Lefevre, C., and Litinas, N. (1983). Stochastic equilibrium model of peak period traffic congestion. Transportation Science, Vol. 17 (4), pp. 430-453.

De Palma, A., Hansen, P, and Labbe, M. (1990). Commuters' Paths with penalties for early or late arrival time. Transportation Science, Vol. 24 (4), pp. 276-286.

Dial, R. B. (1979). A model and algorithms for multicriteria route-mode choice. Transportation Research Part B, Vol. 13 (4), pp. 311-316.

Dial, R. B. (1996). Bicriterion traffic assignment: basic theory and elementary algorithms. Transportation Science, Vol. 30 (2), pp. 93-110.

Dial, R. B. (1997). Bicriterion traffic assignment: efficient algorithms plus examples. Transportation Research Part B, Vol. 31 (5), pp. 357-379.

Dong, J., Erdoğan, S., Lu, C.-C., and Mahmassani, H. S. (2006). State-dependent pricing for real-time freeway management: static, reactive and anticipatory. Paper presented at the 86th (2007) Annual Meeting of the Transportation Research Board.

Elimam, A., and Kohler, D. (1997). Two engineering applications of a constrained shortest-path model. European Journal of Operations Research, Vol.103 (3), pp. 426-438.

Florian, M., and Nguyen, S. (1974). A method for computing network equilibrium with elastic demands. Transportation Science, Vol. 8 (4), pp. 321-332.

Florian, M. (2006). Network equilibrium models for analyzing toll highways. In: Mathematical and Computational Models for Congestion Charging. Lawphongpanich, S. et al. eds. Springer, New York, USA, pp. 105-116.

Friesz, T. L., Bernstein, D., Mehta, N. J., Tobin, R. L., and Ganjalizadeh, S. (1989). Dynamic network traffic assignmnet considered as a continuous time optimal control problem. Operations Research, Vol. 37 (6), pp. 893-901.

Friesz, T. L., Bernstein, D., Smith, T. E., Tobin, R. L. And Wie, B. W. (1993). A variational inequality formulation of the dynamic network user equilibrium problem. Operations Research, Vol. 41 (1), pp. 179-191.

Garey, M., and Johnson, D. (1979). Computers and Intractability: A Guide to the Theory of NP Completeness. W. H. Freeman, San Francisco.

Han, D. and Lo, H. K. (2004). Solving nonadditive traffic assignment: a descent method for co-coercive variational inequalities. European Journal of Operational Research, Vol. 159, pp. 529-544.

Hansen, P. (1980). Bicriterion path problems. In G. Fandel and T. Gal (eds.), Multiple Criteria Decision Making: Theory and Applications, Springer, Heidelberg, pp. 109-127.

Hassin, R. (1992). Approximation schemes for the restricted shortest path problem. Mathematics of Operation Research, Vol.17, pp. 36-42.

Hearn, D. (1982). The gap function of a convex program. Operations Research Letter, Vol. 1 (2), pp. 67-71.

Hendrickson, C. and Kocur, G. (1981). Schedule delay and departure time decisions in a deterministic model. Transportation Science, Vol. 15 (1), pp. 62-77.

Hendrickson, C. and Plank, E. (1984). The flexibility of departure times for work trips. Transportation Research 18A, pp. 25-36.

Henig, M. I. (1985). The shortest path problem with two objectives. European Journal of Operation Research, Vol. 25 (2), pp. 281-291.

Hensher, D. A. (2001a). The valuation of commuter travel time savings for car drivers: evaluating alternative model specifications. Transportation, Vol. 28 (2), pp. 101-118.

Hensher, D. A. (2001b). Measurement of the valuation of travel time savings. Journal of Transport Economics and Policy, Vol. 35 (1), pp. 71-98.

Huang, H. -J. and Lam, W. H. K. (2002). Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues. Transportation Research Part B, Vol. 36 (3), pp. 253-273.

Huang, H. –J. and Lam, W. H. K. (2003). A multi-class dynamic user equilibrium model for queueing networks with advanced traveler information systems. Journal of Mathematical Modeling and Algorithm, Vol. 2, pp. 349-377.

Iida, Y. (1999). Basic concepts and future directions of road network reliability analysis. Journal of Advanced Transportation, Vol. 33 (2), pp. 125-134.

Jackson, B. and Jucker, J. V. (1981). An empirical study of travel time variability and travel choice behavior. Transportation Science, Vol. 16 (4), pp. 460-475.

Jang, W., Ran, B., and Choi, K. (2005). A discrete time dynamic flow model and a formulation and solution method for dynamic route choice. Transportation Research Part B, Vol. 39 (7), pp. 593-620.

Janson, B. N. (1991a). Convergent algorithm for dynamic traffic assignment. Transportation Research Record, 1328, pp. 69-80.

Janson, B. N. (1991b). Dynamic traffic assignment for urban road networks. Transportation Research Part B, Vol. 25 (2/3), pp. 143-161.

Janson, B. N. and Robles, J. (1993). Dynamic traffic assignment with arrival time costs. Proceedings of the 12th International Symposium on Transportation and Traffic Theory, Daganzo, C. F. (Ed.) Elsevier, USA, pp. 127-146.

Jayakrishnan, R., Mahmassani, H. S., and Hu, T.-Y. (1994a). An evaluation tool for advanced traffic information and management systems in urban networks. Transportation Research Part C, Vol. 2 (3), pp. 129-147.

Jayakrishnan, R., Tsai, W. K., Prashker, J. N., and Rajadhyaksha, S. (1994b). A faster path-based algorithm for traffic assignment. Transportation Research Record, 1443, pp. 75-83.

Joksimovic, D., Bliemer, M. C. J., Bovy, P. H. L., and Verwater-Lukszo, Z. (2005). Dynamic road pricing for optimizing network performance with heterogeneous users. In Networking, Sensing, and Control, Proceedings, IEEE.

Korpelevich, G. M. (1977). The extragradient method for finding saddle points and other problems. Matekon (Translations of Russian and East European mathematical economics), Vol. 13, pp. 35-49.

Lam, W. H. K. and Xu, G. A. (1999). A traffic simulator for network reliability assessment. Journal of Advanced Transportation, Vol. 33 (2), pp. 159-182.

Lam, T. C., and Small, K. A. (2001). The value of time and reliability: measurement from a value pricing experiment. Transportation Research Part E, Vol. 37 (2/3), pp. 231-251.

Larsson, T. and Patriksson, M. (1992). Simplicial decomposition with disaggregated representation for the traffic assignment problem. Transportation Science, Vol. 26 (1), pp. 4-17.

Lawphongpanich, S. and Hearn, D. W. (1984). Simplicial decomposition of asymmetric traffic assignment problem. Transportation Research Part B, Vol. 18 (2), pp. 123-133.

Leurent, F. (1993). Cost versus time equilibrium over a network. European Journal of Operational Research, Vol. 71 (2), pp. 205-221.

Leurent F. (1996). The theory and practice of a dual criteria assignment model with continuously distributed values-of-times. In J. B. Lesort, editor, Transportation and Traffic Theory, Pergamon, Exeter, England, pp. 455-477.

Liu, H. X., Recker, W., and Chen, A. (2004). Uncovering the contribution of travel time reliability to dynamic route choice using real-time loop data. Transportation Research Part A, Vol. 38 (6), pp. 435-453.

Liu, H. X, He, X, and Recker, W. (2007). Estimation of time-dependency of values of travel time and its reliability from loop detector data. Transportation Research Part B, Vol. 41 (4), pp. 448-461.

Lo, H. K. and Chen, A. (2000). Reformulating the general traffic equilibrium problem via a smooth gap function. Mathematical and Computer Modeling, Vol. 31 (2/3), 179-195.

Lo, H. K. and Szeto, W. Y. (2002). A cell-based variational inequality formulation of the dynamic user optimal assignment problem. Transportation Research Part B, Vol. 36 (5), pp. 421-443.

Lu, C.-C., Zhou, X. and Mahmassani, H. S. (2006). Variable toll pricing and heterogeneous users: model and solution algorithm for the bi-criterion dynamic traffic assignment problem. Transportation Research Record: Journal of the Transportation Research Board, No. 1964, pp.19-26.

Magnanti, T. L. and Perakis, G. (1993a). A unifying geometric solution framework and complexity analysis for variational inequalities. Working paper OR 276-93, Operations Research Center, Massachusetts Institute of Technology.

Magnanti, T. L. and Perakis, G. (1993b). On the convergence of classical variational inequalities. Working paper OR 282-93, Operations Research Center, Massachusetts Institute of Technology.

Magnanti, T. L. and Perakis, G. (1997a). Averaging schemes for variational inequalities and systems of equations. Mathematics of Operations Research, Vol. 22 (3), pp. 568-587.

Magnanti, T. L. and Perakis, G. (1997b). Solving variational inequality and fixed point problems by averaging and optimizing potentials. Report # OR 324-97, Operations Research Center, Massachusetts Institute of Technology.

Maher, M. J. and Hughes, P. C. (1997). A probit-based stochastic user equilibrium assignment model. Transportation Research Part B, Vol. 31 (4), pp. 341-355.

Maher, M. J. (1998). Algorithms for logit-based stochastic user equilibrium assignment. (1998). Transportation Research Part B, Vol. 32 (8), pp. 539-549.

Mahmassani, H. S. and Herman, R. (1984). Dynamic user equilibrium departure time and route choice on idealized traffic arterials. Transportation Science, Vol. 18 (4), pp. 362-384.

Mahmassani, H. S. and Chang, G.-L. (1985). Dynamic aspects of departure time choice behavior in a commuting system: theoretical framework and experimental analysis. Transportation Research Record, 1037, pp. 88-101.

Mahmassani, H. S. and Mouskos, K. (1988). Some numerical results on the diagonalization algorithm for network assignment with asymmetric interactions between cars and trucks. Transportation Research Part B**,** Vol. 22 (4), pp. 275-290.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. Networks and Spatial Economics, Vol. 1 (3/4), pp. 267-292.

Mahmassani, H. S., Qin, X., Zhou, X., Mahfoud, R. H., Lu, C.-C. and Erdogan, S. (2003). TMC evaluation report for a real-time DTA system prototype: phase 1.5B. Technical Report, Maryland Transportation Initiative, University of Maryland, College Park, USA.

Mahmassani, H. S., Fei, X., Eisenman, S. and Zhou, X. (2005a). Adaptation and testing of DYNASMART-X as a decision support tool for the Maryland CHART traffic management center. Technical Report, Maryland Transportation Initiative, University of Maryland, College Park, USA.

Mahmassani, H. S., Zhou, X., and Lu, C.-C. (2005b). Toll pricing and heterogeneous users: approximation algorithms for finding bi-criterion time-dependent efficient paths in large-scale traffic networks. Transportation Research Record: Journal of the Transportation Research Board, No. 1923, pp.28-36.

Marcotte, P. and Zhu, D. L. (1997). Equilibria with infinitely many differential classes of customers. In Complementarity and Variational Problems. State of the Art, Proceedings of the 13th International Conference on Complementarity Problems, Jong-Shi Pang and Michael Ferris, eds., SIAM, Philadelphia, pp. 234-258.

Marcotte, P. (1999). Reformulations of a bi-criterion equilibrium model. In M. Fukushima, L. Qi, eds. Reformulation – Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods. Kluwer Academic, Dordrecht, The Netherlands, pp. 269-292.

Martins, E. Q. V. (1984). On a multicriteria shortest path problem. European Journal of Operational Research, Vol. 16 (2), pp. 236-245.

McFadden, D. and Train, K. (2000). Mixed MNL models for discrete response. Journal of Applied Ecnonmetrics, Vol. 15 (5), pp. 447-470.

Mekky, A. (1995). Toll revenue and traffic study of highway 407 in Toronto. Transportation Research Record, 1498, pp. 5-15.

Mekky, A. (1997). Comparing tolling strategies for highway 407 in the Greater Toronto area. Paper presented at the 76th Annual Meeting of Transportation Research Board, Washington D. C., USA.

Merchant, D. K. and Nemhauser, G. L. (1978a). A model and algorithm for the dynamic traffic assignment problem. Transportation Science, Vol. 12 (3), pp. 183-199.

Merchant, D. K. and Nemhauser, G. L. (1978b). Optimality conditions for a dynamic traffic assignment model. Transportation Science, Vol. 12 (3), pp. 200-207.

Miller-Hooks, E., and Mahmassani, H. S. (1998). Optimal Routing of Hazardous Materials in Stochastic, Time-Varying Transportation Networks. Transportation Research Record 1645, pp. 143-151.

Mirchandani, P. and Soroush, H. (1987). Generalized traffic equilibrium with probabilistic travel times and perceptions. Transportation Science, Vol. 21 (3), pp. 133-152.

Mote, J., Murthy, I. and Olson, D. (1991). A parametric approach to solving bicriterion shortest path problems. European Journal of Operation Research, Vol. 53 (1), pp. 81-92.

Nagel, K., Esser, J., and Rickert, M. (2000). Large-scale traffic simulations for transport planning. In: Stauffer, D. (Ed.), In Annual Review of Computer Physics VII. World Scientific, Singapore, pp. 151–202.

Nagurney, A., and Zhang, D. (1996). Projected Dynamical Systems and Variational Inequalities with Applications. Kluwer Academic Publishers, Boston, MA, USA.

Nagurney, A., and Zhang, D. (1997). Projected dynamic systems in the formulation, stability analysis, and computation of fixed-demand traffic network equilibria. Transportation Science Vol. 31 (2), pp. 147-158.

Nagurney, A. (1998). Network Economics: A Variational Inequality Approach. Kluwer Academic Publishers, Boston, MA, USA.

Nagurney, A., and Dong, J. (2002). A multiclass, multicriteria traffic network equilibrium model with elastic demand. Transportation Research Part B, Vol. 36 (5), pp. 445-469.

Newell, G. F. (1987). The morning commute for nonidentical travelers. Transportation Science, Vol. 21(2), pp. 74-88.

Nguyen, S. (1974). A unified approach to equilibrium methods for traffic assignment. In M. Florian (ed.), Traffic Equilibrium Methods, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York, pp. 148-182.

Nielsen, L. R. (2003). Route choice in stochastic time-dependent networks: theory, algorithms, and computational experience. Ph. D. Thesis, Department of Operations Research, University of Aarhus, Denmark.

Nielsen, O. A., Daly, A., and Frederiksen, R. D. (2002). A stochastic route choice model for car travelers in the Copenhagen region. Networks and Spatial Economics, Vol. 2 (4), pp. 327-346.

Patriksson, M. (1994). The Traffic Assignment Problem: Models and Methods. VSP, Utrecht, The Netherlands.

Patriksson, M. (1999). Nonlinear Programming and Variational Inequality Problems – A Unified Approach. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Peeta, S. (1994). System optimal dynamic traffic assignment in congested networks with advanced information systems. Ph. D Dissertation, the University of Texas at Austin.

Peeta, S. and Mahmassani, H. S. (1995). System optimal and user equilibrium time-dependent traffic assignment in congested networks. Annals of Operation Research, 60, pp. 81-113.

Peeta, S. and Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: the past, the present and the future. Networks and Spatial Economics, Vol. 1 (3/4), pp. 233-265.

Ran, B., Boyce, D. E., and LeBlanc, L. J. (1993). A new class of instantaneous dynamic user-optimal traffic assignment models. Operations Research, Vol. 41 (1), pp. 192-202.

Ross, S. (2002). A First Course in Probability. Prentice-Hall, Inc. New York.

Roth, G. (1967). Paying for Roads: Economics of Traffic Congestion, Penguin Books, Middlesex, England.

Sbayti, H., Lu, C.-C., and Mahmassani, H. S. (2006). Efficient implementations of the method of successive averages in simulation-based DTA models for large-scale network applications. Forthcoming in Transportation Research Record.

Sheffi, Y. (1985). Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, New Jersey, USA.

Skriver, A. J. V., and Andersen, K. A. (2000). A label correcting approach for solving bicriterion shortest-path problems. Computers & Operations Research, Vol. 27 (6), pp. 507-524.

Small, K. A. (1982). The scheduling of consumer activities: work trips. American Economic Review, Vol. 72 (3), pp. 467-479.

Small, K. A., and Yan, J. (2001). The value of "value pricing" of roads: second-best pricing and product differentiation. Journal of Urban Economics, Vol. 49 (2), pp. 310-336.

Small, K. A., Winston, C., and Yan, J. (2005). Uncovering the distribution of motorists' preferences for travel time and reliability. Econometrica, Vol. 73 (4), pp. 1367-1382.

Smith, M. J. (1979). The existence, uniqueness and stability of traffic equilibria. Transportation Research B, Vol. 13 (4), pp. 295-304.

Smith, M. J. (1993). A new dynamic traffic model and the existence and calculation of dynamic user equilibrium on congested capacity-constrained road networks. Transportation Research Part B, Vol. 27 (1), pp. 49-63.

Smith, M. J. and Wisten, M. B. (1995). A continuous day-to-day traffic assignment model and the existence of a continuous dynamic user equilibrium. Annals of Operations Research, 60, pp. 59-79.

Szeto, W. Y. and Lo, H. K. (2004). A cell-based simultaneous route and departure time choice with elastic demand. Transportation Research Part B, Vol. 38 (7), pp. 593-612.

Szeto, W. Y. and Lo, H. K. (2005). Non-equilibrium dynamic traffic assignment. Proceedings of the 16th International Symposium on Transportation and Traffic Theory, Mahmassani, H. S. (Ed.) Elsevier, USA, pp. 427-445.

Taylor, M. A. P. (1999). Dense network traffic models, travel time reliability and traffic management II: applications to reliability. Journal of Advanced Transportation, Vol. 33 (2), pp. 235-244.

Tong, C. O. and Wong, S. C. (2000). A predictive dynamic traffic assignment model in congested capacity-constrained road networks. Transportation Research Part B, Vol. 34 (8), pp. 625-644.

Train, K. (2003). Discrete choice analysis with simulation. Cambridge University Press, New York, USA.

Verhoef, E. T. (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 (8), pp. 707-729.

Vickrey, W. S. (1963). Pricing in urban and suburban transport. American Economic Review, Vol. 53 (2), pp. 452-465.

Vickrey, W. S. (1969). Congestion theory and transport investment. American Economics Review, 59 (2), pp. 251-260.

Walters, A. A. (1961). The theory and measurement of private and social cost of highway congestion. Econometrica, Vol. 29 (4), pp. 676-699.

Warburton, A. (1987). Approximation of pareto optima in multiple-objective shortest path problems. Operations Research, Vol. 35 (1), pp. 70-79.

Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. Proceedings of Institute of Civil Engineers, II (1), pp. 325-378.

Wie, B.-W., Tobin, R. L., Friesz, T. L., Bernstein, D. (1995). A discrete time, nested cost operator approach to the dynamic network user equilibrium problem. Transportation Science, Vol. 29 (1), pp. 79-92.

Wie, B.-W. and Tobin, R. L. (1998). Dynamic congestion pricing models for general traffic networks. Transportation Research Part B, Vol. 32 (5), pp. 313-327.

Yang, H., and Bell, M. G. H. (1997). Traffic restraint, road pricing and network equilibrium. Transportation Research Part B, Vol. 31 (4), pp. 303-314.

Yang, H., and Meng, H. (1998). Departure time, route choice and congestion toll in a queuing network with elastic demand. Transportation Research Part B, Vol. 32 (4), pp. 247-260.

Yang, H., and Meng, H. (2000). Highway pricing and capacity choice in a road network under a build-operate-transfer scheme. Transportation Research Part A, Vol. 34 (3), pp. 207-222.

Yang, H., Tang, W. H., Cheung, W. M., and Meng, Q. (2002). Profitability and welfare gain of private toll roads in a network with heterogeneous users. Transportation Research Part A, Vol. 36 (6), pp. 537-554.

Yang, H., Huang, H.-J. (2004). The mulit-class, multi-criteria traffic network equilibrium and systems optimum problem. Transportation Research Part B, Vol. 38 (1), pp. 1-15.

Yildirim, M. B. and Hearn, D. W. (2005). A first best toll pricing framework for variable demand traffic assignment problems. Transportation Research Part B, Vol. 39 (8), pp. 659-678.

Yin, Y. and Ieda, H. (2001). Assessing performance reliability of road networks under non-recurrent congestion. Transportation Research Record, 1771, pp. 148-155.

Yin, Y. Lam, W. H. K., and Ieda, H. (2004). New technology and the modeling of risk-taking behavior in congested road networks. Transportation Research Part C, Vol. 12 (3/4), pp.171-192.

Ziliaskopoulos, A. K. and Mahmassani, H. S. (1993). Time Dependent Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications. Transportation Research Record, 1408, pp. 94-100.

Ziliaskopoulos, A. K. and Rao, L. (1999). A simultaneous route and departure time choice equilibrium model for dynamic networks. International Transactions in Operational Research, Vol. 6 (1), pp. 21-37.

Ziliaskopoulos, A. K. and Waller, S. T. (2000). An internet-based graphical information system that integrates data, models, and users for transportation applications. Transportation Research Part C, Vol. 8 (1), pp. 427-444.

Ziliaskopoulos, A. K., Waller, S. T., Li, Y., and Byram, M. (2004). Large-Scale Dynamic Traffic Assignment: Implementation Issues and Computational Analysis. Journal of Transportation Engineering, Vol. 130 (5), pp. 585-593.