PROBIT-BASED STOCHASTIC USER EQUILIBRIUM PROBLEMS AND THEIR APPLICATIONS IN CONGESTION PRICING

LIU ZHIYUAN

NATIONAL UNIVERISTY OF SINGAPORE

2011

PROBIT-BASED STOCHASTIC USER EQUILIBRIUM PROBLEMS AND THEIR APPLICATIONS IN CONGESTION PRICING

LIU ZHIYUAN

(B.ENG., Southeast University, Nanjing, China)

A DISSERTATION SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHYLOSOPY

IN THE DEPARTMENT OF

CIVIL AND ENVIRONMENTAL ENGINEERING

NATIONAL UNIVERISTY OF SINGAPORE

2011

ACKNOWLEDGEMENT

My sincerest appreciation goes to my supervisor, Associated Professor Meng Qiang for his guidance, constructive suggestions and continuous encouragement throughout my graduate education. In each stage of my Ph.D study, from course study to the qualifying exams and to the research work, he was always supportive and giving me valuable advices. Without him, the work in this dissertation would not be possible. He will always be taken as a mentor and friend throughout my career and life.

I am very grateful to Prof. Chin Hoong Chor, Dr. Ong Ghim Ping Raymond and Dr. Szeto Wai Yuen for their encouragement and advices on this research work.

I also acknowledge Mr. Foo Chee Kiong, Madam Yap-Chong Wei Leng, and Madam Theresa Yu-Ng Chin Hoe for their hospitality and kind assistance.

Thanks are also extended to my research colleagues: Qu Xiaobo, Khoo Hooi Ling, Wang Tingsong, Wang Xinchang, Wang Shuaian, Weng Jinxian, H.R. Pasindu, William Yap, Zhang Jian, Zhao Ben, Xu Haihua, and Yan Yadan for their support and cooperation throughout my Ph.D study. A special debt of gratitude is also owed to the other research mates for their help and encouragement.

Finally, my deepest appreciation goes to my parents, my parents-in-law and my beloved wife Wang Zhijing for their endless love as well as enthusiastic and consistent support for my Ph.D study.

TABLE OF CONTENTS

ABSTRACT

When compared to other user equilibrium principles for traffic assignment, the probit-based stochastic user equilibrium (SUE) is known to have properties well suited for practical conditions. However, theoretical studies and practical implementations of probit-based SUE are largely limited due to the difficulties of solving such a problem. Thus, a primary objective of this dissertation is to inherently reduce the computational time of solving the probit-based SUE problem. To further improve its suitability to practical conditions, the following extensions are further taken into consideration for the traffic assignment problem (a) elastic demand, (b) asymmetric link travel time functions, termed as probit-based asymmetric SUE problem with elastic demand (PA-SUEED).

Although it converges sub-linearly, the cost averaging (CA) method is the only known convergent algorithm for PA-SUEED in the literature. This dissertation accelerates the computation of PA-SUEED from two aspects: firstly, it proposes two projection-type prediction-correction (PC) algorithms with linear convergent speed. As validated by numerical experiments, the two PC algorithms can accelerate the computational speed for five to ten times, when compared with CA method; secondly, note that the solution algorithms for SUE problems need to calculate the stochastic network loading (SNL) problem in each iteration, and solution algorithm for the SNL in the context of PA-SUEED is still an open question. A link-based two-stage Monte Carlo simulation method is proposed for the SNL problem, wherein each trial of this Monte Carlo simulation method is independent with identical tasks, thus it has a superior parallelism. Therefore, this dissertation further improves the computational speed of solving PA-SUEED by

proposing three distributed (parallel) computing approaches for its SNL problem. Based on a comprehensive numerical experiment, it shows that the distributed computing approaches can further improve the computational speed for over fifty times.

Link capacity constraints are recognized to be a logical extension of standard traffic assignment problems. However, studies for SUE problem with link capacity constraints are fairly scarce, due to the difficulties in formulating and solving this problem. In the context of PA-SUEED, this problem becomes even more complicated and challenging. This dissertation thus investigates about formulating and solving the PA-SUEED with link capacity constraints, which is a highly mathematical topic with considerable theoretical contributions. A VI model is proposed and the monotonicity and Lipschitz-continuity of this VI model are rigorously proven. Based on these properties of the VI model, convergence of a PC algorithm thus can be guaranteed to solve the VI model. The proposed methodology is finally validated by a numerical example.

The un-cooperative travel behavior of drivers would usually lead to traffic congestions, especially in the dense urban areas. Thereby, the network authorities intend to encourage them to use uncongested road segments. Congesting pricing is one of the few instruments for this purpose, thus it is a good complement for the studies of traffic assignment. Note that the drivers' value-of-time (VOT) is necessitated for the analysis of congestion pricing. In this study, VOT is assumed to be continuously distributed, to cover the vast diversity of drivers' income levels. On the other hand, the drivers' diversity of perception errors on travel times should also be considered, which gives rise to SUE principles. Thus, another objective of this dissertation is to investigate about the congestion pricing

problem with PA-SUEED constraints. Originated from the current toll adjustment procedure used by the Electronic Road Pricing (ERP) system in Singapore, a practical-oriented research topic, termed as speed-based toll design, for cordon-based congestion pricing scheme is discussed. Subsequently, in view that the ERP system intends to update its current entry-based charge to a distance-based charge, the distance-based toll design for cordon-based congestion pricing scheme is then formulated and solved. These two toll design topics are of considerable importance to the practical implementations of congestion pricing schemes. It should be noted that formulations and solution algorithms for congestion pricing problems with probit-based SUE constraints are also quite limited. Thus, the achievements in this dissertation not only contribute to the theoretical studies of congestion pricing problems, but also significantly facilitate to the practical operations and supervisions of congestion pricing schemes.

GLOSSARY OF NOTATION

A	Set of links
$\lvert A\rvert$	Cardinality of set A
c_{wk}	Travel time on path $k \in R_w$, $c_{wk}(\mathbf{v}) = \sum_{a \in A} t_a(\mathbf{v}) \delta_{ak}^w$
$\mathbf{c}_w$	Column vector of travel time on all the paths associated with OD pair $w \in W$, $\mathbf{c}_w = (c_{wk}, k \in R_w)^{\mathrm{T}}$
$D(\cdot)$	Demand function
$\mathbf{f}_w$	Column vector of traffic flows on all the paths between OD pair $w \in W$, namely, $\mathbf{f}_w = (f_{wk}, k \in R_w)^{\mathrm{T}}$
$\mathbf{f}$	Column vector of traffic flow on all the paths in the network, i.e., $\mathbf{f} = (\mathbf{f}_w, w \in W)^{\mathrm{T}}$
h_a	Capacity of link flow v_a
H_a	A threshold for link flow v_a, i.e., $H_a \le h_a$
N	Set of nodes
p_{wk}	Choice probability of path $k \in R_w$
q_w	Travel demand between OD pair $w \in W$
$\mathbf{q}$	Column vector of all the OD travel demands, $\mathbf{q} = (q_w, w \in W)^{\mathrm{T}}$
$\bar{q}_w$	Upper bound of q_w
R_w	Set of all the paths between OD pair $w \in W$
$\lvert R_w\rvert$	Cardinality of set R_w

S_w	Satisfaction function, the expected minimal cost on paths between $w \in W$
$t_a(\mathbf{v})$	Travel time on link $a \in A$, and it is a function of link traffic flow vector $\mathbf{v}$
$\mathbf{t}(\mathbf{v})$	Column vector of all the link travel time functions, $\mathbf{t}(\mathbf{v}) = \left(t_a(\mathbf{v}), a \in A\right)^{\mathrm{T}}$
t_a^0	Free flow travel time of link $a \in A$
v_a	Flow on link $a \in A$
$\mathbf{v}$	Column vector of all the link traffic flows, $\mathbf{v} = \left(v_a, a \in A\right)^{\mathrm{T}}$
$\mathbf{u}$	Column vector of all the Lagragian multipliers of the capacity constraints
W	Set of OD pairs
α	Drivers' value-of-time
γ_i	average speed of all the vehicles in cordon i
$\underline{\gamma}_i$ ($\overline{\gamma}_i$)	predetermined lower (upper) bound of γ_i
$\boldsymbol{\tau}$	Column vector of toll charges, $\boldsymbol{\tau} = \left(\tau_a, a \in \overline{A}\right)^{\mathrm{T}}$
Δ_w	Link/path incidence matrix associated with OD pair $w \in W$, namely, $\Delta_w = \left(\delta_{ak}^w, a \in A, k \in R_w\right)$
Δ	Link/path incidence matrix for the entire network, $\Delta = \left(\Delta_w, w \in W\right)$
ζ_{wk}	Drivers' perception error on c_{wk}
$\boldsymbol{\zeta}_w$	Column vector of perception errors on all the paths between OD pair $w \in W$, i.e., $\boldsymbol{\zeta}_w = \left(\zeta_{wk}, w \in W\right)^{\mathrm{T}}$
Λ	OD pair/path incidence matrix, $\Lambda = \left(\delta_{wk}, w \in W, k \in R_w\right)$, where δ_{wk}

	equals to 1 if path $k \in R_w$ and 0, otherwise
Ω	Symbol for feasible sets, for instance, Ω_v is the feasible set for link flows

ACRONYMS

CA	Cost-averaging method
CBD	Central Business District
CDF	Cumulative Distribution Function
DL	Distributed Loading
DRGA	Distributed Revised Genetic Algorithm
DSP	Distributed Shortest-Path
DUE	Deterministic User Equilibrium
ERP	Electronic Road Pricing system in Singapore
GA	Genetic Algorithm
IIA	Independent and irrelevant alternatives
IL	Integrated Loading
LTA	Land Transport Authority
MPEC	Mathematical programming with equilibrium constraints
MSA	Method of Successive Average
OD	Origin-Destination
PA-SUEED	Probit-based Asymmetric SUE problem with Elastic Demand
PC	Prediction-correction algorithm
PDF	Probability density function
SNL	Stochastic network loading
SO	System Optimum
SUE	Stochastic User Equilibrium
TAP	Traffic assignment problem

TSB	Total social benefit
UE	User Equilibrium
VI	Variational inequality
VOT	Value-of-time

CHAPTER 1 INTRODUCTION

1.1 Background and Motivations

Transportation planning for urban road networks aims to effectively and efficiently satisfy the citizens' requirement for movement, which may influence economic vitality and affect the quality of life (Shiftan et al., 2007). Together with an enormous increase in demand, some other challenges have emerged for the transportation planning, which include environmental degradation and global warming, safety issue and increasing complexity of commuters' travel behavior. To cope with these challenges, the studies for transportation planning have attracted much attention.

The most well-known approach for urban transportation planning so far is the four-step method, including trip generation, trip distribution, mode split, and traffic assignment (Bell and Iida, 1977; Patriksson, 1994a; Ortuzar and Willumsen, 1995). Among these four steps, traffic assignment was the first and most-prevalent topic investigated by the professionals. Traffic assignment deals with allocating traffic demands to existing or hypothetical transportation networks. It can hence be utilized to assess the deficiencies in the existing network or the effects of some improvements (expansions of the road section or some newly built links), and to evaluate alternative transportation system plans. Furthermore, traffic assignment is also a preliminary for some research topics based on the transportation network, e.g. congestion pricing problems (Ferrari, 1995; Yang and Huang, 2005; Verhoef et al., 2008), signal control problems (Smith, 1987; Yang and Yagar, 1995; Wong and Yang, 1997), network design problems (Abdulaal and LeBlanc, 1979; Ben-Ayed et al., 1988; Yang and Bell, 1998) and Origin-Destination (OD) matrix

estimation problems (Maher, 1983; Bell, 1991; Yang et al., 1992; Cascetta and Postorino, 2001).

The first approach used for traffic assignment is the all-or-nothing assignment; i.e. all the travel demand is allocated to the shortest path. No iterative updates are required for this approach and it is thus quite computationally economical. But this technique is unrealistic and it gives rise to improper results as per some early empirical studies (e.g. Campbell, 1950; Carroll, 1959).

Most well designed transportation systems rely on good understanding of human behaviors. A milestone of the studies in traffic assignment also results from an in-depth analysis of the commuters' travel behaviors. Wardrop (1952) proposed two famous principles for the network flows: (a) the User Equilibrium (UE), when assuming that all the network users make their route choice by selecting the path with minimal travel time. This assumption would give rise to equilibrium of network flows, where no one can reduce his/her trip time by changing the trip route; (b) the System Optimum (SO), by assuming that the users would mutually cooperate to minimize the total travel time in the transportation system. In reality, the UE principle is more realistic, since the users are non-cooperative when making trip decisions.

These two principles are vital foundations for the mathematical models and algorithms developed for the traffic assignment problems. A major breakthrough was put forward by Beckmann et al. (1956) via a convex nonlinear optimization model, whose solution is the

UE link flow. Note that Frank and Wolfe (1956) provided a convex combination algorithm for solving the nonlinear convex optimization problems, which is a method of feasible directions. When applying this method to solve Beckmann et al.'s model, it incorporates a series of sub-problems which are merely the all-or-nothing assignments, and this property is commonly known as the Cartesian product structure (Larsson and Patriksson, 1992). This structure inherently reduces computational demands for solving UE problems. However, Beckmann's model relies on some oversimplified and unrealistic assumptions, including fixed demand, separable link travel time functions and no link capacity constraints. These assumptions were later relaxed by some studies (see the discussions in section 2.2 to 2.4).

A pioneering work was made by Daganzo and Sheffi (1977): since the UE principle unrealistically assumes that the users have an accurate estimation of the on-trip travel time before their journey, Daganzo and Sheffi extended this assumption by defining the users' perceived travel time as random variables. This new principle is commonly known as stochastic user equilibrium (SUE). It was formulated by Daganzo (1982) and Sheffi and Powell (1982) as an un-constrained optimization models, and these models can be solved by the famous method of successive average (MSA) introduced by Powell and Sheffi (1982). Many refer to UE as Deterministic User Equilibrium or DUE to distinguish it from SUE.

In previous studies for SUE, the users' perceived travel time is always assumed to follow Gumbel distribution (logit-based SUE) or normal distribution (probit-based SUE).

Formulations and algorithms for logit-based SUE have been fully studied (e.g. Dial, 1971; Fisk, 1980; Bell, 1995a&b). However, the logit model has an inherent drawback that it cannot differentiate the overlapping parts of path alternatives, which is known as Independent and Irrelevant Alternatives (IIA) (Chapter 10 of Sheffi, 1985). The probit-based SUE, in nature, can avoid IIA problem and thus better represents realistic conditions.

Regarding the studies of probit-based SUE problems, the major difficulty results from the fact that no closed-form expression can be provided for the choice probability on each path (see section 2.1 for further discussions). Thus, despite its better representativeness to practical conditions, the probit-based SUE has not been sufficiently investigated. Although the concept of probit-based SUE has been proposed as early as in 1977 by Daganzo and Sheffi, many significant extensions to this problem are still open questions, including probit-based SUE with elastic demand, asymmetric link flow interactions and link capacity constraints. Compared with the standard probit-based SUE, these extensions make the resulting models more realistic. This study thus intends to take an in-depth investigation about the mathematical models and computational methods for these problems.

Congestion pricing is one of the most effective measures utilized in urban area to alleviating traffic congestions. It levies toll charges on vehicles driving at particular links or areas to encourage the drivers using uncongested road segments, in order to achieve a better network condition. Ever since Pigou (1920), the literature on congestion pricing

problems is extensive, see the monographs by Small, 1992; Yang and Huang, 2005; Lawphongpanich, et al., 2006; Verhoef, et al., 2008, among many others. There is a strong connection between traffic assignment and congestion pricing. Because, on one hand, congestion pricing is an effective economic lever to adjust the outcomes of traffic assignment, and on the other hand traffic assignment is a crucial foundation and preliminary for the analysis of congestion pricing. Therefore, congestion pricing is a perfect complement of the studies of traffic assignment, also taken as a target of this dissertation.

1.2 Research Scope

The aforementioned general model put forward by Daganzo and Sheffi (1977) is assumed to be in the framework of fixed demand, separable link travel time functions (no link flow interactions), and no capacity constraint. In contrast with the practical conditions, these assumptions are unrealistic: (a) for the travel demand, the whole transportation system is a service system for the total travel demand, and each travel mode involved acts as a competitor evaluated by the users. As the congestion in road network increases, the potential users would change to other travel modes (e.g. public transport systems) or even cancel their trip plans. Hence, travel demand on static road networks should be a function of the travel cost; (b) for the link flow interactions, in some road sections, the interactions between flows on different links could be quite significant and asymmetric, thus they should not be neglected, for instance, heavy traffic on two-way streets without separated road median, and un-signalized intersections; (c) for the capacity constraints, link flow in reality could not exceed its physical capacity, but traffic assignment problem with no

capacity constraint would generate unrealistically saturated link flows, which undermines the reasonability of traffic assignment results.

Therefore, the unrealistic assumptions from these three aspects are relaxed, and this research targets at the probit-based asymmetric SUE problem with elastic demand, which is abbreviated as PA-SUEED. The mathematical models and efficient solution algorithms for PA-SUEED are first investigated, and then PA-SUEED with link capacity constraints is addressed. Further efforts are devoted to the congestion pricing problems in the context of PA-SUEED.

1.3 Objectives

The objectives of this research are as follow:

1. To develop mathematical models and *efficient computational algorithms* for the probit-based asymmetric SUE problem with elastic demand (PA-SUEED);
2. To develop mathematical models and computational algorithms for the *capacity constrained* PA-SUEED;
3. To further accelerate the computational speed of proposed algorithms using *Distributed Computing approaches*, based on large-scale numerical experiments;
4. Theoretical analysis of the practically implemented *congestion pricing schemes* as well as the second-generation *distance-based pricing*.

1.4 Organization of the Dissertation

Chapter 1 provides a general introduction to the probit-based SUE problem and congestion pricing, where the significance and rationality for current research are discussed. Furthermore, the objectives and research scope of this study are highlighted.

Chapter 2 presents a detailed literature review about the research topics involved in this dissertation, namely models and algorithms for (a) probit-based SUE problems, (b) traffic assignment with elastic demand, asymmetric link travel time functions and/or link capacity constraints, (c) congestion pricing problems.

Chapters 3 to 5 focus on the theoretical analysis of the PA-SUEED problem itself, which are in the domain of traffic assignment. Then, Chapters 6 and 7 address two crucial congestion pricing schemes, taken as significant complement to facilitate urban transportation demand management.

Chapter 3 aims to propose mathematical models and efficient computational approaches for the PA-SUEED problem. Two variational inequality (VI) models are first provided for this problem, and it is rigorously proved that these two models both possess unique solutions, which are equivalent to the PA-SUEED link flow. Subsequently, a link-based two-stage computational procedure is presented for the probit-based stochastic network loading procedure using Monte Carlo simulation. Then, it is proven that a projection-based prediction-correction (PC) algorithm incorporating this Monte Carlo simulation

method is more efficient than the existing algorithms to solve the PA-SUEED problem. Superiority of the proposed algorithm is further validated by two numerical examples.

Chapter 4 intends to add capacity constraints to the PA-SUEED problem. Likewise to the capacity constrained DUE problem, we first provide a set of equivalent conditions for this capacity constrained SUE problem, named as generalized SUE conditions. These conditions are thence formulated by a proposed monotone and Lipschitz-continuous VI model, and the existence and uniqueness property of its optimal solution are rigorously proven. Regarding the solution method, this problem is converted into solving a serial of un-capacity constrained traffic assignment problems which can be handled by the algorithms introduced in Chapter 3. The PC algorithm with adaptive step sizes is also utilized to solve the VI model proposed for the capacity constrained PA-SUEED problem, which can converge to the optimal solution.

Chapter 5 investigates about the Distributed Computing approaches to further accelerate the algorithms discussed in Chapter 3 for PA-SUEED. The Monte Carlo simulation-based method for the stochastic network loading has satisfactory accuracy level, while it has largely increased the computational burdens, thus inhibits the research for probit-based SUE problems. However, the Monte Carlo simulation-based method has perfect parallelism, making it ideal for parallel computing. Thereby, in this chapter three approaches are proposed on the workload partition of the Monte Carlo simulation method for distributed (parallel) computing. Performances of the three approaches are

comprehensively tested by numerical experiments wherein a randomly generated network as well as a large-scale network is used.

Inspired by the toll adjustment roles used by Electronic Road Pricing (ERP) system in Singapore, Chapter 6 addresses the speed-based toll design for cordon-based congestion pricing scheme, where the commuters' route choice behavior is assumed to follow the PA-SUEED with continuously distributed value-of-time (VOT). In practice, to improve traffic conditions within the cordon area is a major concern of the cordon-based congestion pricing. However, this concern has seldom been considered in the theoretical studies. In this chapter, average travel speed of vehicles in the cordon area is first taken as an index of its traffic conditions, and then the toll charges on each entry of the cordon are designed such that the average travel speed can be maintained in a targeted range. Termed as speed-based toll design, this problem is formulated as a mathematical programming with equilibrium constraints (MPEC) model.

Chapter 7 discusses about the distance-based toll design for cordon-based congestion pricing scheme in the context of PA-SUEED. Targeted as the next generation of ERP system (Ohno, 2007), distance-based toll charge is more equal/fair than the current pay-per-entry or daily license basis toll charges. It is assumed that the toll charge is decided by a nonlinear function of the travel distance in the cordon. Termed as toll-charge function, such a function should be allowed to be generic to any positive and non-decreasing functional form. Drivers' travel behavior is still assumed to be PA-SUEED with continuously distributed value-of-time (VOT). A methodology is then introduced in

Chapter 7 to efficiently solve the distance-based toll design with the objective of maximizing total social benefit (TSB).

Chapter 8 concludes this dissertation and discusses about future research advices.

CHAPTER 2 LITERATURE REVIEW

2.1 Users' Travel Behavior and Probit-based SUE

2.1.1 User's travel behavior and SUE

An accurate understanding of human behavior would help to improve the service level of a system. Urban transportation is a crucial *service* system for the citizens' travel requirements. A satisfying planning scheme of urban transportation system is thus highly relied on a rational recognition of the commuters' travel behavior. Dating back to 1841, Kohl assumed that the travelers would individually choose the route perceived as the shortest/cheapest. This assumption of the travel behavior results in an equilibrium of the flows in the road network, which is summed up by Wardrop (1952):

> *The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.*

This theory is usually referred to as Wardrop's first principle or deterministic user equilibrium (DUE). The rationale underlying this principle is quite straightforward, since if travel times on used routes are not equal, network users would have an incentive to change to the shorter one. The users are regarded as "selfish", since they only concerns about their own travel cost, and it may lead to traffic congestions. Therefore, DUE is not the most realistic equilibrium network condition. Wardrop also proposed another equilibrium representing the perfect network condition, which is called system optimum (SO) or Wardrop's second principle:

> *The average journey time is a minimum.*

The SO can be achieved if all the users make their travel plans in terms of their marginal travel costs; namely, SO implies that the users' marginal travel costs on all the used routes are equal. In reality, the users' travel behavior can be adjusted in order to achieve SO in two cases: a centralize control over trip making decisions (in an industrial logistics system or computer controlled networks in rail system) or using first-best congestion pricing strategy, see section 2.3.1.

Nonetheless, it should be pointed out that Kohl and Wardrop's assumptions of users' travel behavior presumed that all the travelers has complete and accurate information about the entire network before their trips. This assumption is unrealistic even if the users have a long-term experience about the network conditions, due to the daily variations of travel times and the diversity from users' sense of time. A well known breakthrough on this issue was made by Daganzo and Sheffi (1977), where they extended this assumption by assuming users' pre-trip perceived travel times on all the routes are random variables. Namely, users' perceived travel times $\mathbf{C}_w = \left[C_{wk}, k \in R_w\right]^{\mathrm{T}}$ on paths connecting Origin-Destination (OD) pair $w \in W$ equal to the actual travel times $\mathbf{c}_w$ plus a multivariate random variable ζ_w. Therefore, the users would choose the route with minimal perceived travel time. In accordance with this new assumption of users' behavior, a new network-equilibrium can be achieved based on the discrete choice model (Daganzo and Sheffi, 1977). The new equilibrium principle, named stochastic user equilibrium (SUE), can be stated as:

> *In a SUE network no user **believes** he can improve his travel time by unilaterally changing routes.*

Despite that all the users intend to minimize their perceived travel times, the perceived travel times on all the used paths are not equal. Instead, each route is only personally perceived by the users on it to be the shortest one among all the alternatives. For a network with fixed travel demand, any flow pattern that can fulfill the following condition is regarded as a SUE link flow pattern:

$$f_{wk} = q_w p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\right)\right), w \in W, k \in R_w. \tag{2.1}$$

where q_w is the OD demand and p_{wk} is the path choice probability, defined by

$$\Pr\left(C_{wk} \le C_{wl}, \forall l \in R_w\right) = \int_{C_{wk} \le C_{w1}} \int_{C_{wk} \le C_{w2}} \cdots \int_{C_{wk}=-\infty}^{C_{wk}=+\infty} \phi_{\zeta_w}\left(x_1, \cdots, x_{|R_w|}\right) dx_1 \cdots dx_{|R_w|}. \tag{2.2}$$

where C_{wk} is the users' perceived path travel time, which is a random variable. $\phi_{\zeta_w}\left(x_1, \cdots, x_{|R_w|}\right)$ is the probability density function of the multivariate random variable $\mathbf{C}_w = \left(C_{wk}, k \in R_w\right)^{\mathrm{T}}$.

Similarly to the DUE case, the researchers kept searching for a proper definition for the stochastic based social optimum. Only recently, Maher et al. (2005) provided an in-depth investigation about the stochastic social optimum (SSO), which is defined as:

At the SSO solution, the total of the users' perceived costs is minimized.

Maher et al. also proved that "*the marginal costs play the same role in the SSO as the standard costs play in SUE*". Namely the SSO solution can be achieved by using the algorithms for solving SUE, where the travel time functions should be replaced by the marginal costs. Similarly to SO, this theory provides a first-best benchmark for the network operations.

In the case the perception error ζ_w is allowed to follow any kind of random distributions, it is called as General SUE problem. In particular, a serial of studies have also been conducted in the literature by assuming ζ_w following any specific distribution, including uniform distribution (Williams, 1977), Gumbel distribution (logit-based SUE problem) and normal distribution (probit-based SUE problem), see, Sheffi (1985) and Patriksson (1994a). Among these SUE problems, the logit-based SUE and probit-based SUE have been mostly focused and comprehensively studied.

The logit-based SUE has been well investigated since it can provide an explicit and concise expression for the path choice probability $p_{wk}(\mathbf{c}_w)$:

$$p_{wk}(\mathbf{c}_w) = \frac{\exp(-\theta c_{wk})}{\sum_{k \in R_w} \exp(-\theta c_{wk})} \tag{2.3}$$

where θ is a positive parameter. The choice probability of logit-based SUE *merely* depends on the difference between each two path costs. In spite of its computational advantages, the logit-based SUE has an inherent drawback, which is known as Independent and Irrelevant Alternatives (IIA). That is, logit-based model is lack of sensitivity to network topology and only depends on the difference in travel time (Sheffi, 1985).

Probit-based SUE takes into account the correlation of the travel costs on different paths, thus overcomes the IIA problem. Therefore, probit-based model has better representativeness to the practical conditions and it is a superior representativeness of the

SUE problems. However, despite these robust characteristics, no closed form can be provided for the choice probability of probit-based problem, thus prohibits the investigation for this problem. Compared with logit-based SUE, the research for probit case is quite limited. Thereby, probit-based SUE problem is a timely research topic with significant theoretical contributions. This dissertation aims to contribute to this topic from two aspects: accurately estimate the choice probability and then provide an efficient solution technique for the probit-based SUE problem.

2.1.2 Models and Algorithms for the SUE Problem

The breakthrough work made by Daganzo and Sheffi (1977) provides a conceptual framework of General SUE problem as well as stochastic network loading (SNL) procedure. Herein, the SNL aims to load flows to the network in terms of fixed link travel costs (see comprehensive discussions in Section 2.1.3), and to solve the SNL is equivalent to solving the choice probability. Regarding the equivalent mathematical model for the General SUE problem, Daganzo (1982) provided an un-constrained convex optimization model as follows:

$$\min z(\mathbf{t}) = -\sum_{w \in W} q_w E\left[\min_{k \in R_w} \{C_{wk} \mid \mathbf{c}_w\}\right] + \sum_{a \in A} \int_{t_a^0}^{t_a} t_a^{-1}(x) dx , \tag{2.4}$$

where t_a^0 represents the free flow travel time and $t_a^{-1}(v)$ is the inverse of link travel time function. This un-constrained model is proven to be convex, yet it requires calculating the inverse travel time functions, which are computationally demanding. Sheffi and Powell (1982) therefore transformed this model into the following one:

$$\min z(\mathbf{v}) = -\sum_{w \in W} q_w E\left[\min_{k \in R_w} \{C_{wk} \mid \mathbf{c}_w\}\right] + \sum_{a \in A} v_a t_a(v_a) + \sum_{a \in A} \int_0^{v_a} t_a(x) dx . \tag{2.5}$$

This model is equivalent to the previous one and much easier in terms of computation. It can be seen that these two models both contain an implicit proportion $E\left[\min_{k\in R_w}\{C_{wk} \mid \mathbf{c}_w\}\right]$ in the objective function, thus it is difficult to conduct a line-search procedure in the solution algorithm for the optimal step sizes. Therefore, the conventional Frank-Wolfe Method used for solving DUE problem is no longer available for these two models. An convergent solution algorithm was put forward by Powell and Sheffi (1982) using the method of successive average (MSA). This algorithm has a brief recursive function:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \alpha_n\left(\mathbf{y}^n - \mathbf{v}^n\right), \tag{2.6}$$

where the auxiliary link flow $\mathbf{y}^n$ can be obtained by performing a SNL procedure in terms of fixed link travel time pattern $\mathbf{t}(\mathbf{v})$, and α_n is a predetermined step size, commonly taken as $\frac{1}{n}$. This predetermined step size circumvents the difficulty in line-search but also brings an inferior sub-linear convergent speed. Convergence of the MSA type algorithms are usually proven by virtue of the Blum's theory (Blum, 1954; Daganzo, 1983; Cantarella, 1997). Noted that some further efforts have been made recently to improve the efficiency of MSA (see, e.g., Liu et al., 2009).

For solving model (2.5), another efficient solution algorithm called Stochastic Assignment Method (SAM) was developed by Maher and Hughes (1997b). SAM adopts the Clark's approximation (see, Rosa and Maher, 2002) to calculate the objective function, and therefore a line-search can be conducted for the step size.

Another milestone of modeling for General SUE problem was made by Daganzo (1983) in studying SUE problem with asymmetric link travel time functions, where a fixed-point model is proposed:

$$f_{wk} = q_w p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\right)\right), w \in W, k \in R_w . \tag{2.7}$$

It has been proven that when the link travel time functions are continuous and strictly monotone, the optimal solution of this model is unique. In addition, Daganzo proved that the MSA algorithm is still available to solve this problem. It should be pointed out that this model proposed is also effective for the case of joint DUE and SUE problems.

The aforementioned models and algorithms for *General* SUE problem are all effective for *logit-based* and *probit-based* SUE problem. Apart from these models and algorithms, studies for probit-based SUE are quite limited. Yet, an extensive literature can be observed for investigations of logit-based SUE, because the logit model can be derived from entropy maximization functions, which overcomes the uncertainty issue of its objective function. A model with explicit objective function in terms of path flow was proposed by Fisk (1980), and ever since then various extensions have been developed for logit-based SUE problems (e.g. Dial, 1971; Chen and Alfa, 1991; Bell, 1995a&b; Prashker and Bekhor, 2004; Bekhor and Toledo, 2005, etc.).

2.1.3 Stochastic Network Loading Procedure

As claimed above, the solution algorithms for SUE problem incorporates a SNL procedure, which provides a set of SUE link flows in terms of fixed link travel times based on the theory of discrete choice model. This SNL procedure can be regarded as a mapping from the feasible set of link travel time to that of link flows, and it plays a

similar role as all-or-nothing assignment in solving DUE problem. The SNL problem in the context of SUE is usually analyzed by the well known discrete choice model (see, e.g., Ben-Akiva and Lerman, 1985).

For logit-based SUE, a pioneer work was conducted by Dial (1971) for solving the SNL, where a heuristic algorithm named STOCH was provided. This algorithm only covers those "reasonable" routes, which would only take the drivers farther from the origin and closer to the destination. This algorithm is quite efficient in that it can avoid path enumeration and obviates the paths with cycles. The STOCH algorithm has been further extended by Gunarsson (1972), Tobin (1977); Bell (1995 a&b) and Leurent (1995), etc.

For the probit case, its SNL has no close form, thus it is approximately solved by two types of methods: analytical approximation methods and Monte Carlo simulation-based methods. Regarding the analytical approximation methods, five different methods have been developed so far (Rosa and Maher, 2002), including the Improved Clark method, Simple Clark method, Mendell and Elston method (Mendell and Elston, 1974), Separated Split method (Langdon, 1984 a&b) as well as Tang and Melchers method (Tang and Melchers, 1987). Among all these methods, the Improved Clark method was first developed and most commonly used in the literature. It was originally put forward by Daganzo and Sheffi (1977) for solving the probit-based SUE problem, and later extended by Maher and Hughes (1997b) in the aforementioned SAM method. These approximation methods all possess certain drawbacks, e.g. since the approximation processes are conducted separately, the choice probability does not sum up to 1, and moreover they are

getting quite computational demanding and inaccurate when the number of alternative variables are getting larger.

Thus, this study aims to use Monte Carlo simulation for solving the SNL of probit-based SUE problem. The Monte Carlo simulation method uses the choice frequency to estimate the choice probability, the rationality of which is ensured by the weak law of large numbers. It was first utilized to estimate the probit-based choice probabilities by Lerman and Manski (1978). The Monte Carlo simulation method can be adopted not only for probit-based SUE but also any other kind of distributions, by merely generating different random number series following different distributions.

2.1.4 Parallel Computing for Monte Carlo simulation

Monte Carlo method has been widely adopted in the areas of simulating physical and mathematical systems (see the monographs by, e.g., Rubinstein, 1981; Binder and Heermann, 1992; Gentile, 1998), and it refers to the reputed simulations using random numbers to approximate the answer to a stochastic problem. To achieve a higher accuracy level, the Monte Carlo method usually requires a larger sample size, and hence it is computationally prohibitive. However, when dealing with independent tasks, the repeatability of the sampling procedure of Monte Carlo method makes it ideal for parallel computation by different processors. Note that for solving the SNL problem, each trial of the proposed Monte Carlo simulation-based method is independent, thus it is ideal for parallel computing.

A brief survey is conducted here for the parallel Monte Carlo method in solving independent tasks. Moatti et al. (1987) has proposed a parallel approach for the high energy physics Monte Carlo simulations, and it is suitable for any Monte Carlo method with subtask treelike structure, where each subtask can be executed independently. Later, in the field of molecular systems, a parallel Monte Carlo method is introduced by Traynor and Anderson (1988) to determine energy differences among molecules, and the calculations are carried out independently with different scales of length and energy parameters for cancellations of random sampling error. Yet, as pointed out by Esselink et al. (1995), the parallelism in Traynor and Anderson (1988) is not used to speed up one single problem, since the simulations with different scale of length and energy parameters are complete and irrelevant. A more straightforward parallel approach was used by Zhao and Wood (1989) for the radiation transport analysis, in which each trial of the Monte Carlo method is independent and it thus has a natural parallelism. This parallel Monte Carlo method proposed by Zhao and Wood (1989) was subsequently extended by Wood et al. (1991) as well as Singleton et al. (1991) by analyzing its implementations on different types of parallel computer architectures. Note that in Singleton et al. (1991) the parallel Monte Carlo method for independent tasks is tested on a distributed-memory multiprocessor system, which coincides with the objective of this chapter. As a side note, various studies are also found in the literature for the parallel Monte Carlo method in solving dependent (sequential) tasks, e.g., the parallel hybrid Monte Carlo method (Kennedy, 1999).

The Monte Carlo simulation method used for solving SNL is similar to that addressed in Zhao and Wood (1989), and it therefore also possesses a superior parallelism. Thus, it is convenient to accelerate the computational speed of Monte Carlo simulation for solving SNL by using distributed (parallel) computing, which is presented at Chapter 5.

2.2 Extensions of Conventional User Equilibrium Problem

As claimed in Chapter 1, a major scope of this study focuses on the probit-based SUE problem with three extensions: elastic demand, asymmetric link travel time functions, and link capacity constraints, and such a problem is abbreviated as PA-SUEED with/without link capacity constraints. These three extensions have modified some unrealistic assumptions of the conventional SUE problem discussed above. Thus, an investigation about the literature of these three extensions for user equilibrium (both DUE and SUE) is first provided, which are preliminaries for research topics in the remaining chapters.

2.2.1 Elastic Demand

2.2.1.1 Models and Algorithms for DUE with Elastic Demand

The aforementioned models at Section 2.1 all assume that the trip rate between each OD pair is fixed and known. However, from practical point of view, the total travel demand for a static road network would be inherently influenced by the level of service on road network. Rationality of the fluctuation in travel demand results from two aspects: (a) the potential users may switch to other travel modes if the congestion on road network gets worse (b) if their trips have lower emergency, some users may change their travel plan to other time spans or even drop their travel plan. This phenomenon should be taken into

account by assuming that the travel demand $q_w, w \in W$ is a function of the travel disutility between OD pair $w \in W$. For the DUE case, the travel disutility is defined as the minimum travel time between each OD pair, μ_w, so demand function follows the following form:

$$q_w = D_w(\mu_w), \tag{2.8}$$

where $D_w(\bullet)$ is the demand function, and for the fixed demand case, this function equals to a constant. Beckmann et al. (1956) proposed the following equivalent model for the DUE problem with elastic demand:

$$\min \sum_{a \in A} \int_0^{v_a} t_a(w)\,dw - \sum_{w \in W} \int_0^{q_w} D_w^{-1}(w)\,dw \tag{2.9}$$

Subject to

$$\begin{aligned} &\sum_{k \in R_w} f_{wk} = q_w \\ &f_{wk} \geq 0, k \in R_w, w \in W \end{aligned} \tag{2.10}$$

Regarding the solution algorithm, the Frank-Wolfe method can also be adopted here, since the computational demand for solving its sub-problem is simply equals to that of an all-or-nothing assignment. A more efficient method for this problem is using network representation (Sheffi, 1985, Section 6.3) to transform it into a DUE problem with fixed demand, and thence solve a standard DUE traffic assignment problem (TAP).

2.2.1.2 Models and Algorithms for SUE with Elastic Demand

Likewise to the DUE case, a demand function is defined for the total trip rate of each OD pair. The travel disutility for SUE problem is defined as the mean value of users' minimal perceived travel time between OD pair, $S_w(\mathbf{c}_w)$, which equals to:

$$S_w(\mathbf{c}_w) = E\left[\min_{k \in R_w}\{C_{wk} \mid \mathbf{c}_w\}\right], w \in W. \tag{2.11}$$

$S_w(\mathbf{c}_w)$ is commonly recognized as satisfaction function in the literature. Accordingly, travel demand for SUE with elastic demand should follow this function (Cantarella, 1997; Maher and Zhang, 2000):

$$q_w = D_w\left(S_w(\mathbf{c}_w)\right). \tag{2.12}$$

Note that for eqns. (2.8) and (2.12) the elastic demand q_w should be bounded from above: $q_w \in [0, \overline{q}_w]$, where the upper bound $\overline{q}_w$ would be influenced by the total population and car ownership of the origin zone.

An equivalent model for General SUE problem with elastic demand was put forward by Cantarella (1997) by extending Daganzo (1983)'s fixed-point model into the elastic demand case:

$$f_{wk} = q_w p_{wk}\left(\mathbf{c}_w(\mathbf{f})\right), k \in R_w, w \in W \tag{2.13}$$

$$q_w = D_w\left(S_w\left(\mathbf{c}_w(\mathbf{f})\right)\right), w \in W \tag{2.14}$$

Cantarella proved the existence and uniqueness of the solution. This model can generalize the fixed demand case and elastic demand case, since travel demand $D_w(\bullet)$ is only required to be non-increasing and it covers the constant case. Thus, this generic assumption of OD demand function is also made in this dissertation. As to the solution

method for this model, Cantarella (1997) has proposed a convergent Cost-Averaging method, which is a variation of the MSA.

General SUE problem with elastic demand was conducted by the group of Maher et al., see Maher, 2001; Maher and Hughes, 1997a; Maher and Hughes, 1998; Maher et al. 1999; Maher and Zhang, 2000. In these studies, they extend Sheffi and Powell (1982)'s optimization model to the elastic demand case by providing the following model:

$$\begin{aligned}\min z(\mathbf{v},\mathbf{q}) = &\sum_{a\in A} v_a t_a(v_a) - \sum_{a\in A}\int_0^{v_a} t_a(x)\,dx + \sum_{w\in W} D_w^{-1}(q_w) D_w\left(S_w(\mathbf{c}_w)\right) \\ &- \sum_{w\in W} S_w(\mathbf{c}_w) D_w\left(S_w(\mathbf{c}_w)\right) + \sum_{w\in W}\int_0^{q_w} D_w^{-1}(\omega)\,d\omega - \sum_{w\in W} q_w D_w^{-1}(q_w)\end{aligned} \tag{2.15}$$

Optimum of this equivalent model can fulfill the SUE link flow condition as well as elastic demand functions. A Balanced Demand Algorithm (BDA) is developed (Maher and Zhang, 2000) for solving this model, where quadratic equation is utilized to approximate the value of objective function, and thus a line-search can be conducted.

Only recently, Connors et al. (2007) defined a new concept for elastic demand problem, based on a disaggregate point of view. Namely, each individual makes his/her travel decision separately: each user gains a fixed utility from his/her trip, so if the travel disutility is larger than this fixed utility value, he/she would drop this trip plan. Therefore, a demand function can be defined based on the average value of travel utility among all the users in corresponding origin zone. If the travel disutility is larger than this aggregate utility value, the travel demand would be zero, otherwise it would takes the upper bound value. The network representation methods mentioned above can be adopted to solve this

new elastic demand problem. However, it should be noted that this concept contains two major drawbacks: the demand function is not single-valued and continuously differentiable; the aggregate utility value over each origin zone is difficult to estimate.

2.2.2 Asymmetric Link Travel Time Functions

2.2.2.1 Models and Algorithms for DUE with Asymmetric Link Travel Time Functions

Conventional studies of traffic assignment problems all assume that the link travel time functions are independent, namely, link travel time is merely a function of its own flow. This kind of link travel time function is termed separable, and the Jacobian matrix of link travel time vector wrt. link flow vector is symmetric and diagonal. Nonetheless, this assumption is not always valid in reality, because the link flow interactions may be quite severe in some road sections: two-way streets with heavy traffic; priority junctions; intersections with responsive traffic signals (Dafermos, 1980 & 1983; Watling, 1998, etc.). Therefore, it is rational to assume that link travel time is also influenced by flows on other links, and such an influence may be asymmetric. Thus, each link travel time is defined as a function of flows on *all the links*, denoted by $t_a(\mathbf{v}), a \in A$. For conciseness, DUE or SUE problem with symmetric/asymmetric link travel time functions is usually termed as symmetric/asymmetric DUE or SUE.

Dafermos (1971) first investigated about the symmetric DUE problem. This work was later extended to the asymmetric case by Smith (1979) who developed a variational inequality (VI) formulation:

$$\mathbf{t}(\mathbf{v}^*)(\mathbf{v}-\mathbf{v}^*) \geq 0, \forall \mathbf{v} \in \Omega_v, \tag{2.16}$$

where the feasible set for link flows denoted by Ω_v is compact. It should be pointed out that a similar equivalent VI model in terms of path flow and path travel time can also be utilized:

$$\mathbf{c}\left(\mathbf{f}^*\right)\left(\mathbf{f}-\mathbf{f}^*\right)\geq 0, \forall \mathbf{f}\in\Omega_f . \tag{2.17}$$

This path-based formulation is effective for the TAP with non-additive path travel time functions. Smith (1979) claimed that the existence and uniqueness of these two VI models can be guaranteed if the travel time functions are continuous and strictly monotone. However, Dafermos (1980) concurrently showed that Smith's uniqueness condition was equivalent to the positive definite property of the Jacobian. For an asymmetric matrix, its positive definiteness can be ensured if it can fulfill the following two conditions: (a) diagonal values are all larger than zero; (b) the matrix and its transpose matrix are diagonally dominant. These two conditions imply that the influence on travel time from its own link flow is much larger than that of other link flows.

The VI model (2.16) has then been further extended by Florian (1979), Dafermos (1980 & 1982), Fisk and Nguyen (1982), Florian and Spiess (1982), Hearn and Lawphongpanich (1984), Lawphongpanich and Hearn (1984), and Panicucci et al. (2007), to name a few. An alternative formulation was put forward by Aashtiani (1979) and Aashtiani and Magnanti (1981), which showed that the asymmetric DUE problem could be formulated as a non-linear complementary problem.

Regarding the solution algorithms, Asmuth (1978), Aashtiani (1979) and Aashtiani and Magnanti (1981) proposed applications of fixed-point, non-linear complementarity and

linear complementarity algorithms, respectively. Meanwhile, Dafermos (1980 & 1982) presented some projection methods and Nguyen and Dupuis (1982) utilized the cutting plane algorithm to solve the asymmetric DUE problem.

2.2.2.2 Models and Algorithms for SUE with Asymmetric Link Travel Time Functions

Few studies can be observed for the asymmetric SUE problem. An equivalent fixed-point model was developed by Daganzo (1983) as introduced above, at eqn. (2.7). Daganzo showed that the uniqueness of optimal solution can be assured if the link travel time functions are strictly monotone, while the positive definiteness of Jacobian matrix is not required.

This fixed-point model was extended to the elastic demand case by Cantarella (1997), see, eqn. (2.13) and (2.14). For such an asymmetric SUE problem with elastic demand, only the CA method proposed by Cantarella (1997) is a *convergent* solution algorithm in the existing literature.

2.2.3 Link Capacity Constraints

In reality, when link flow approaches its physical capacity, this link would be over-saturated and travel time is geometrically increased. Thus, link flow could never exceed its physical capacity. This phenomenon, however, is not taken into consideration in the standard TAPs. Thus, the resultant link flows would be many times larger than its physical capacity, which highly undermines the reliability and rationality of traffic assignment results (Larsson and Patriksson, 1995 & 1999). Studies for the TAP with link capacity constraints are thus of considerable significance.

In the existing literature, two approaches can be adopted to cope with the TAP with link capacity constraints. The first approach is to use the asymptotic link travel time function, where the link travel time would be infinity when flow approaches its capacity. The second approach is to put an additional term to the link travel time functions, and these new functions are called as generalized link travel time functions, denoted by $T_a(\mathbf{v}) = t_a(\mathbf{v}) + u_a, a \in A$. Regarding the additional term, u_a, in generalized link travel time functions, it has three interpretations: (a) link tolls on the saturated links that drives away some potential users (e.g., Beckmann and Golob, 1974; Ferrari, 1995 & 1997; Yang and Lam, 1996); (b) the additional waiting time caused by traffic signal (Smith, 1987; Yang and Yagar, 1995; Wong and Yang, 1997); (c) queuing delay on the entry of over-saturated links (Miller et al., 1975; Thompson, 1976; Bell, 1995a). Boyce et al. (1981) indicated that the first approach would cause unrealistic high travel times, thus the second approach is commonly used in the literature.

If the optimal toll/queuing-delay value, u_a, on each link is given, the capacity constrained problem is equivalent to a standard TAP. Therefore, to solve the TAP problem with link capacity constraints is, in general, equals to search for the optimal u_a. Any pattern of $u_a, a \in A$ would be regarded as optimal if and only if it can fulfill the following conditions:

$$\begin{gathered} v_a \le H_a, a \in A \\ u_a \left(v_a - H_a \right) = 0, a \in A \\ u_a \ge 0, a \in A \\ v_a = \sum_{w \in W} \sum_{k \in R_w} f_{wk} \delta_{ak}^w \\ f_{wk} \ge 0 \end{gathered} \tag{2.18}$$

where H_a denotes the capacity constraints on each link. These conditions imply that u_a may be larger than zero only when the link flow equals to its capacity.

For the DUE problem with link capacity constraints, a suitable optimization model can be obtained by directly adding the link flow capacity constraints to Beckmann's model (Larsson and Patriksson, 1995):

$$\min z(\mathbf{v}) = \sum_{a \in A} \int_0^{v_a} t_a(\omega) d\omega \tag{2.19}$$

Subject to

$$\begin{gathered} v_a \le H_a, a \in A \\ f_{wk} \ge 0, k \in R_w, w \in W \\ v_a = \sum_{w \in W} \sum_{k \in R_w} f_{wk} \delta_{ak}^w \end{gathered} \tag{2.20}$$

However, when Frank Wolfe method is adopted for solving this model, its sub-problem contains the capacity constraints hence making it more computationally demanding than solving a shortest path problem. Thus, some methods are used to relax the capacity constraints and transform it into solving a series of un-capacity constrained DUE problem. These methods include penalty function methods (Hearn and Ribera, 1980; Inouye, 1987), the dual methods (Hearn and Lawphongpanich, 1990; Larsson and Patriksson, 1999) and the augmented Lagrangian multiplier methods (Larsson and Patriksson, 1995). It was

further verified by Nie et al. (2004) that the DUE problem with link capacity constraints is computationally tractable for large-scale networks.

Studies for SUE problem with link capacity constraints are quite scarce. This is because, unlike the DUE case, directly adding capacity constraints to optimization model for un-constrained SUE problem is not effective. For logit-based SUE, based on the explicit model proposed by Fisk (1980), Bell (1995a) developed an optimization model and a solution algorithm for capacity constrained problem. Only recently, Meng et al. (2008) gave a linearly constrained convex minimization model and a convergent Lagrangian dual method for the General SUE problem with link capacity constraints. However, both of these two studies assumed separate link travel times and fixed demand. To our best knowledge, no effective study so far has been conducted for the PA-SUEED problem with link flow capacity constraints. This problem is addressed in Chapter 4, which is a complicated topic with high theoretical contribution.

2.3 Congestion Pricing with User Equilibrium Constraints

Studies on traffic assignment are mainly based on the user equilibrium, which captures users' travel behaviors at selecting their route plans. However, as claimed by Wardrop (1952), all the users are "selfish" who only take into account their own on-trip travel costs. Accordingly, some road sections would be extremely saturated, and the whole road network is operating with inferior efficiency. This anarchy traffic assignment results thus cannot satisfy the transport authorities' requirements. The authorities attempt to get a tool that can guide the network users towards making more rational decisions and using the un-congested road segments. A straightforward strategy to adjust their behavior is to levy

additional travel costs on the congested road sections. The concept of congestion pricing thus comes out (see the monograph by Small, 1992; Lewis, 1993; Yang and Huang 2005; Lawphongpanich et al. 2006; Verhoef et al., 2008). Apart from mitigating congestions, the congestion pricing also internalizes the externalities (e.g., travel delays) into toll revenue, which can invest back to the transportation system.

As claimed at Chapter 1, congestion pricing is an effective tool to manage the traffic assignment process, and on the other hand, the theories for traffic assignment problems are foundations for the studies in congestion pricing. It implies that the connection between traffic assignment and congestion pricing is quite significant and intrinsic. Congestion pricing is thus taken as an important scope of this study. A brief introduction and review are provided as follows for three pricing schemes, first-best, second-best and cordon-based.

2.3.1 First-Best and Second-Best Congestion Pricing

A remarkable benchmark for better traffic conditions should be the system optimum, where the total travel time (or total social benefit) on the whole network is optimized. First-best pricing is defined as a toll charge pattern that can achieve such a system-wide index. It has been well recognized that the marginal cost pricing is a solution to the first-best pricing scheme (Walters, 1961; Yang and Huang, 1998; Verhoef et al., 2008). Validity of the marginal cost pricing on general transportation networks was proven by many studies with different assumptions; for instance, with elastic demand (Huang and Yang, 1996), with multi-user classes (Dafermos, 1973), with non-separable link costs

(Smith, 1979), with logit-based SUE constraints (Yang and Huang, 1998), with General SUE constraints (Maher et al., 2005), with stochastic demand (Sumalee and Xu, 2011).

The optimal marginal cost pricing can be easily obtained by solving a traffic assignment problem. An practical-oriented efficient trial-and-error method was proposed by Yang et al. (2004) and Zhao and Kockelman (2006) with DUE constraints and logit-based SUE constraints respectively, where the values of travel demands are not required for the calculation. The work of Yang et al. (2004) was recently extended by Han and Yang (2009) using more efficient step sizes. An in-depth review was provided by Xu (2006) regarding the trial-and-error method on general transportation networks.

The marginal cost pricing requires each link to be charged, thus it is not practically applicable. When assuming only a proportion of the network is charged, the second-best pricing was introduced by Marchand (1968). In the case of only partial links are charge, the marginal-cost, however, is unlikely to give an optimal total social benefit.

A extensive literature can be observed for the second-best pricing problems, which are usually formulated as a Bi-level programming model (BLPM), where its upper-level is to optimize a system-wide index (total social benefit or total travel time) and its lower-level is a traffic assignment problem (Shimizu et al., 1997; Bard, 1998). This BLPM is a typical Stachelberg game in game theory. The lower-level problem of BLPM can be treated as a constraint of the upper-level, thus it gives the form of a mathematical programming with equilibrium constraints (MPEC) model; see, Luo et al., 1996; Verhoef

et al., 1996; Liu and McDonald, 1998 & 1999; Outrata et al., 1998; Bellei, et al., 2002; Verhoef, 2002; Chen and Bernstein, 2004, etc.

A wide range of solution algorithms have been investigated to solve the BLPM or MPEC model, including iterative optimization-assignment algorithm (Allsop, 1974), linearization methods (Ben-Ayed et al., 1988); equilibrium decomposed optimization (Suwansirikeu et al., 1987; Verhoef and Rowendal, 2004), sensitivity-analysis based algorithm (Friesz et al., 1990; Yang, 1997; Patriksson, 2004); augmented Lagrangian algorithm (Meng et al., 2001); Karush-Kuhn-Tucker (KKT) based methods (Verhoef et al., 1996; Verhoef, 2002); stochastic search methods (Shepherd and Sumalee, 2004); gradient-based descent methods (Chiou, 2005); constraint cutting method (Lawphongpanich and Hearn, 2004; Koh et al, 2009).

2.3.2 Cordon-based Congestion Pricing Schemes

As a special second-best pricing scheme, the cordon-based congestion pricing scheme has been not only examined theoretically (e.g. Zhang and Yang 2004; Sumalee, et al. 2005; Sumalee, 2007), but also been implemented practically in some major cities in UK, Singapore and Scandinavia, etc. The cordon-based congestion pricing scheme is defined as: certain district in urban area is encircled by a pricing cordon and any vehicle passing through the cordon is charged. By affecting drivers' route choice plans and subsequently restricting the total number of vehicles entering the encircled district, the cordon-based congestion pricing scheme is taken as an effective tool to mitigate traffic congestion, and it is also convenient for practical operations (May et al., 2002; Akiyama and Okushima, 2006). Cordon-based pricing is thus taken as a target of this dissertation. Assuming the

framework of PA-SUEED for drivers' route choice behavior, this study aims to contribute to the studies of cordon-based pricing by investigating some practical-oriented topics. In terms of practical implementations, these timely topics, including speed-based and distance-based toll design, are more attractive than the first-best and second-best pricing. Formulations and solution algorithms for speed-based and distance-based toll design problems are addressed in Chapters 6 and 7, respectively.

The previous studies for cordon-based congestion pricing can be mainly classified into two categories: optimal cordon location and optimal toll design. The first category intends to find the optimal cordon locations and the second one aims to identify a proper toll fare solution that optimizes a system-wide objective function including the total social benefit based on given cordon locations.

The optimal cordon location problem can be actually formulated as a nonlinear integer programming model with DUE or SUE constraints. The proposed models were approximately solved by the meta-heuristic methods such as genetic algorithm (GA) due to their NP-hardness (May et al., 2002; Shepherd and Sumalee, 2004; Sumalee, 2004; Zhang and Yang, 2004; Sumalee, 2007). The optimal toll design problem for the cordon-based congestion pricing schemes can also be formulated as a BLPM or MPEC, and then solved by the algorithms mentioned at Section 2.3.1.

Practical implementations of cordon-based congestion pricing are then discussed. Cordon-based congestion pricing is first adopted in Singapore in 1975, named as the Area

License Scheme (ALS) (Phang and Toh, 1997), where a daily license is required for any vehicle accessing a restricted zone in the downtown area of Singapore. The ALS scheme was then extended to the Electronic Road Pricing (ERP) system in 1998 (Foo, 2000). By virtue of the electronic devices, ERP uses a non-stop charging system, and its charging rates are differentiated in terms of time of day, location and vehicle type, thus making it more flexible to the congestion level. A "shoulder price" pricing scheme is adopted to ameliorate traffic congestions, i.e. set highest price for morning and evening peak hours, and then cut off this price in steps each half hour *before and after* these two peak hours (website of Singapore Land Transport Authority).

The London Congestion Charging Scheme (LCCS) has attracted much attention. First established in February 2003 by the Transport for London, LCCS is an area-based (license-based) pricing scheme, where each vehicle driving in the charging zone is imposed by a charge of 8 pounds. Santos (2008) argued that LCCS had achieved the aim of reducing traffic congestion in and around the charging zone.

In a more recent trial in Stockholm, between January and July of 2006, a time-differentiated cordon-based pricing scheme is established. The scheme is related to an inner-city zone, where a toll charge is levied on 18 control points located at its entrances and exits. Eliasson et al. (2009) claimed the superiority of congestion pricing scheme in reducing congestions, compared with other traffic management measures (road investments and policies of free public transport, etc.). This trial was later complemented

with a public transport improvement scheme, since a well-functioning public transport system is a prerequisite for demand management.

2.3.3 Continuously Distributed Value-of-Time

Analysis of the congesting pricing problems relies on the theories of traffic assignment, which is the result of commuters' route choice decisions. The commuters select their route plan based on the travel costs on each path. In the case of congestion pricing, travel costs on each path consist of two components: travel time and toll charge. These two components are not in the same units, thus a value-of-time (VOT) is required for the conversion here. It is well-known that the VOT varies among different drivers due to their different levels of income and trip emergency. To reflect this variation, it is thus more rational to take VOT as a continuously distributed random variable across the population (Verheof and Small, 2004; Small et al., 2005; Van den Berg and Verhoef, 2011), instead of assuming homogeneous network users with constant VOT or multiple-user classes with discrete VOTs (Han and Yang, 2008).

Traffic assignment problem with continuously distributed VOT has been comprehensively examined on a hypothetical two-route example, e.g., Mayet and Hansen (2000); Verhoef and Small (2004); Xiao and Yang (2008); Nie and Liu (2010); to name a few. Some findings in these studies, however, may not be available when extended to a realistic network with more than two links.

The traffic assignment problem on real-size networks with continuously distributed VOT was investigated by Leurent (1993), where the commuters' route choice behavior is

assumed to follow DUE principle and travel demand elasticity is also considered. Dial (1996) further studied about the DUE problem with continuously distributed VOT based on a bi-criterion network, assuming that travel times as well as out-of-pocket costs are functions of the link flows. While, regarding the SUE problem with continuously distributed VOT, Cantarella and Binetti (1998) proposed a solution framework for the probit-based SUE problem with continuously distributed VOT. This framework assumed fixed OD demand and relies on a path-based simulation technique to cope with the probit-based SNL. Thus, in the case of PA-SUEED with continuously distributed VOT, it is still an open question. In the Chapter 6 and 7 of this dissertation, a more efficient link-based solution technique is then proposed to solve this problem, which avoids path generation/enumeration.

CHAPTER 3 TWO EFFICIENT PREDICTION-CORRECTION ALGORITHMS FOR PA-SUEED

This chapter presents two variational inequality (VI) models for the probit-based asymmetric SUE problem with elastic demand (PA-SUEED). These two models then contribute to two projection-based prediction-correction (PC) algorithms, which are more efficient than the existing Cost-Averaging (CA) method. Superiority of the proposed algorithm is then numerically validated by two network examples.

3.1 Background

As claimed in Chapter 1, PA-SUEED has better representativeness to the practical conditions, compared with the DUE or standard SUE principles. Yet, the extensions from three aspects (probit-based model, elastic demand and asymmetric link travel time functions) have largely increased the challenges in formulating and solving PA-SUEED. In the existing literature, a fixed-point model proposed by Cantarella (1997) is a suitable formulation for this problem, and the only convergent algorithm was also found in Cantarella (1997), termed as cost-averaging (CA) method, which is a variation of the well-known method of successive average (MSA) (Powell and Sheffi, 1982; Daganzo, 1983), where predetermined step sizes are adopted. Instead, CA method updates the value of link travel times, rather than link flows.

Due to the existence of predetermined and suboptimal step sizes, the CA method only has a sub-linear convergent speed (Nagurney and Zhang, 1996). Thus, it is crucial to find a more efficient solution algorithm for PA-SUEED, in view its theoretical significance and

that it is a foundation for the analysis in the following Chapters. This chapter first proposes two equivalent VI models for the PA-SUEED. The existence and uniqueness conditions of these two models are proven. Furthermore, the Lipschitz-continuity of these two VI models is also demonstrated. Based on these properties, two projection-based prediction-correction (PC) algorithms with self-adaptive step sizes are adopted from the studies for VI (He and Liao, 2002) to solve PA-SUEED. In nature, these two algorithms possess a linear convergent speed, which is superior to the existing CA method.

It should be pointed out that the stochastic network loading (SNL) plays an essential role in the algorithms for SUE problems, as discussed at Section 2.1.3. Monte Carlo simulation-based methods are regarded as an effective one to handle the probit-based SNL. However, in the context of PA-SUEED, it is still an open question for the Monte Carlo simulation-based SNL method. Thereby, before addressing the projection-based solution algorithm, a link-based two-stage Monte Carlo simulation-based SNL method is first put forward in this chapter, which is later taken as one subroutine of the CA method and two PC algorithms.

This chapter is organized as follows. Section 3.2 reviews the two fixed-point models provided by Cantarella (1997) for PA-SUEED. Section 3.3 then proposes one link flow-based and one link cost-based VI model, and some essential properties of these two models are rigorously proven. Section 3.4 presents the said link-based two-stage Monte Carlo simulation-based SNL method. Lower bound for the sample size of each stage of this simulation is theoretically estimated. Section 3.5 introduces the three solution

algorithms, the existing CA method and two PC algorithms. These three algorithms are validated in section 3.6 by the network examples, which numerically verify that two PC algorithms outperform the CA method. Section 3.7 then concludes this chapter.

3.2 SUE Conditions and Two Fixed-point Models

3.2.1 Notation and Definitions

Consider a strongly connected network denoted by $G=(N,A)$ where N and A are the sets of nodes and directed links, respectively. Let W be the set of the OD pairs and R_w be the set of paths between OD pair $w\in W$. Travel demand between OD pair $w\in W$ is denoted by q_w and $\mathbf{q}=(q_w, w\in W)^{\mathrm{T}}$ is a column vector for all these travel demands. Let f_{wk} be traffic flow on path $k\in R_w$ between OD pair $w\in W$, $\mathbf{f}_w=(f_{wk}, k\in R_w)^{T}$ be a column vector of all these path flows between OD pair w and $\mathbf{f}=(\mathbf{f}_w^{\mathrm{T}}, w\in W)^{\mathrm{T}}$ be a column vector of all the path flows over the entire network. Let v_a denote traffic flow on link $a\in A$ and $\mathbf{v}=(v_a, a\in A)^{\mathrm{T}}$ is a column vector of all these link flows. The following flow conservation equations should be fulfilled:

$$\mathbf{q}=\Lambda\mathbf{f}, \tag{3.1}$$

$$\mathbf{v}=\Delta\mathbf{f}, \tag{3.2}$$

$$\mathbf{f}\geq 0. \tag{3.3}$$

where $\Delta=\left[\delta_{ak}^{w}\right]_{|A|\times K}$ and $\Lambda=\left[\delta_{k}^{w}\right]_{|W|\times K}$ are the incidence link/path and OD pair/path matrices, where $|A|$, $|W|$ and K are the number of links, OD pairs and paths,

respectively; $\delta_{ak}^{w}=1$ if link a is on path $k\in R_w$, and $\delta_{ak}^{w}=0$, otherwise. $\delta_{k}^{w}=1$ if path k connects OD pair $w\in W$, and $\delta_{k}^{w}=0$, otherwise.

The travel time on link $a\in A$ is assumed to be a function of link flow vector $\mathbf{v}$, denoted by $t_a(\mathbf{v})$, which is termed as asymmetric link travel time function. These link travel time functions are grouped into a column vector $\mathbf{t}(\mathbf{v})=\left(t_a(\mathbf{v}),a\in A\right)^{\mathrm{T}}$. It is assumed that vector function $\mathbf{t}(\mathbf{v})$ is non-negative, monotonically increasing and continuously differentiable. Travel time on path $k\in R_w$ is the sum of travel times of all the links on this path, denoted by $c_{wk}(\mathbf{v})$, namely,

$$c_{wk}(\mathbf{v})=\sum_{a\in A}t_a(\mathbf{v})\delta_{ak}^{w}. \tag{3.4}$$

These path travel times are grouped into vector $\mathbf{c}_w(\mathbf{v})=\left(c_{wk}(\mathbf{v}),k\in R_w\right)^{\mathrm{T}}$. It is assumed that the travel time on path $k\in R_w$ perceived by drivers, denoted by $C_{wk}(\mathbf{v})$, is a random variable with the expression:

$$C_{wk}(\mathbf{v})=c_{wk}(\mathbf{v})+\zeta_{wk}, \tag{3.5}$$

where ζ_{wk} is a random perception error following normal distribution with zero mean and constant variance. Let $\zeta_w=\left(\zeta_{wk},k\in R_w\right)^{\mathrm{T}}$ denote the column vector of the perception errors on all the paths associated with OD pair $w\in W$, and in the context of probit-based SUE, ζ_w should follow a multivariate normal distribution with zero mean. Let ϕ_{ζ_w} denote

the probability density function of ζ_w, which is strictly positive and continuously differentiable (Daganzo, 1979).

Let $S_w(\mathbf{c}_w)$ denote the expected value of the minimum perceived travel time among all the paths between OD pair $w \in W$, i.e. $S_w(\mathbf{c}_w) = E\left[\min_{k \in R_w}\{C_{wk}\}\right]$. $S_w(\mathbf{c}_w)$ is also known as satisfaction or satisfaction function in the literature (Sheffi 1985).

Travel demand between OD pair $w \in W$ is assumed to be a function of the satisfaction:

$$q_w = D_w\left(S_w(\mathbf{c}_w)\right) \le \overline{q}_w, w \in W, \tag{3.6}$$

where parameter $\overline{q}_w$ is a given upper bound of travel demand between OD pair $w \in W$. The demand function $D_w(\bullet)$ is required to be continuously differentiable and non-increasing. Note that for the fixed demand case, $D_w(\bullet)$ is constant. This assumption is previously made by Cantarella (1997) and Maher and Zhang (2000) for the conventional SUE problem with elastic demand.

Remark 3.1: Link travel time function, $\mathbf{t}(\mathbf{v})$, in this study is assumed to be non-separable with *asymmetric* Jacobian matrix, $\nabla_{\mathbf{v}}\mathbf{t}(\mathbf{v})$. It should be noted that the models and algorithms in this dissertation are also effective for the case of *separable* or *symmetric* link travel time functions.

3.2.2 Probit-based Asymmetric SUE Conditions with Elastic Demand

Any link-flow vector that can satisfy the following eqn. (3.7) is a solution for the PA-SUEED problem. Eqn. (3.7) is thus termed as probit-based asymmetric SUE *conditions* with elastic demand.

$$v_a = \sum_{w \in W} D_w \left(S_w \left(\left(\Delta_w \right)^{\mathrm{T}} \mathbf{t}(\mathbf{v}) \right) \right) P_{wa} \left(\mathbf{t}(\mathbf{v}) \right), a \in A, \tag{3.7}$$

where $P_{wa}\left(\mathbf{t}(\mathbf{v})\right)$ refers to the link usage probability and it equals to

$$P_{wa}\left(\mathbf{t}(\mathbf{v})\right) = \sum_{k \in R_w} p_{wk}(\mathbf{v}) \delta_{ak}^{w}, w \in W, \tag{3.8}$$

where $p_{wk}(\mathbf{v})$ is probability that path $k \in R_w$ is perceived as the shortest one, which is termed as path choice probability, i.e.,

$$\begin{aligned} p_{wk}(\mathbf{v}) &= \Pr\left(C_{wk} \le C_{wl}, \forall l \in R_w\right) \\ &= \int_{c_{wk} \le c_{w1}} \int_{c_{wk} \le c_{w2}} \cdots \int_{c_{wk}=-\infty}^{c_{wk}=+\infty} \phi_{\zeta_w}\left(x_1, \cdots, x_{|R_w|}\right) dx_1 \cdots dx_{|R_w|} \end{aligned} \tag{3.9}$$

It should be pointed out that the path choice probability for path $k \in R_w$ equals to the first-order derivative of the satisfaction function $S_w(\mathbf{v})$ with respect to the path travel time $c_{wk}(\mathbf{v})$ (Sheffi, 1985), namely,

$$\frac{\partial S_w(\mathbf{c}_w)}{\partial c_{wk}(\mathbf{v})} = p_{wk}(\mathbf{v}), k \in R_w, w \in W. \tag{3.10}$$

3.2.4 A Stochastic Network Loading Map and Two Fixed-Point Formulations

Let us define a set of feasible path flows as follows:

$$\Omega_f^0 = \left\{ \mathbf{f} \geq 0 \left| \begin{array}{l} \text{there is a travel demand } q_w \in [0, \bar{q}_w], \\ \text{such that } \sum_{k \in R_w} f_{wk} \delta_{ak}^w = q_w, w \in W \end{array} \right. \right\}. \tag{3.11}$$

With the set of feasible path flows, the set of feasible link flows can be hence defined by

$$\Omega_v^0 = \left\{ \mathbf{v} \mid \mathbf{v} = \Delta \mathbf{f}, \mathbf{f} \in \Omega_f^0 \right\}. \tag{3.12}$$

Given any feasible link flow solution $\mathbf{v} \in \Omega_v^0$, the link travel time vector $\mathbf{t}(\mathbf{v}) = \left(t_a(\mathbf{v}), a \in A\right)^{\mathrm{T}}$ can be easily calculated using the link travel time functions. Let Θ_t^0 be the image set of set Ω_v^0 though link cost function vector $\mathbf{t}(\mathbf{v})$, namely,

$$\Theta_t^0 = \left\{ \mathbf{t} = \mathbf{t}(\mathbf{v}) \middle| \mathbf{v} \in \Omega_v^0 \right\}. \tag{3.13}$$

Although the set Θ_t^0 is bounded and non-empty, however, it may be non-closed and non-convex. Thus, let $\Omega_t^0 \subseteq \Re_+^{|A|}$ be the closed convex hull of set Θ_t^0, and Ω_t^0 is therefore a non-empty, convex and compact set. Cantarella (1997) defined a stochastic network loading (SNL) map from set Ω_t^0 to Ω_v^0:

$$\boldsymbol{\psi}(\mathbf{t}) = \left(\psi_a(\mathbf{t}), a \in A\right)^{\mathrm{T}} : \Omega_t^0 \to \Omega_v^0, \tag{3.14}$$

where

$$\psi_a(\mathbf{t}) = \sum_{w \in W} D_w \left(S_w \left(\left(\Delta_w \right)^T \mathbf{t} \right) \right) P_{wa}(\mathbf{t}), a \in A. \tag{3.15}$$

It can be seen that the SNL map is a probit-based SUE problem with a given/fixed link travel time pattern $\mathbf{t}$. Based on SNL map and link travel time function, the probit-based asymmetric SUE problem with elastic demand can be formulated by two fixed-point models (Cantarella, 1997):

Find a link flow vector $\mathbf{v}^* \in \Omega_v^0$ such that

$$\mathbf{v}^* = \boldsymbol{\psi}\left(\mathbf{t}\left(\mathbf{v}^*\right)\right). \tag{3.16}$$

Find a feasible link travel time vector $\mathbf{t}^* \in \Omega_t^0$ such that

$$\mathbf{t}^* = \mathbf{t}\left(\boldsymbol{\psi}\left(\mathbf{t}^*\right)\right). \tag{3.17}$$

The SNL map $\boldsymbol{\psi}(\mathbf{t})$ is proven to be continuously differentiable and monotone decreasing with respect to vector $\mathbf{t}$ (Lemma 3, Cantarella, 1997). In addition, sets Ω_v^0 and Ω_t^0 both are non-empty, convex and compact. Therefore, these two fixed-point models have unique solution (Theorems 1 and 2, Cantarella, 1997).

3.3 Two Variational Inequality Models

To build a variational inequality (VI) model for the probit-based SUE problem with elastic demand, a set of link flow is defined as follows:

$$\Omega_v = \left\{\mathbf{v} \middle| v_a \in \left[0, v_a^{\max}\right], a \in A\right\}, \tag{3.18}$$

where $v_a^{\max}$ is a predetermined upper bound of traffic flow on link $a \in A$. The upper bound does exist due to the assumption that the travel demand functions are uniformly bounded above; for example,

$$v_a^{\max} = \sum_{w \in W} \overline{q}_w, a \in A. \tag{3.19}$$

In the link travel time space, a set of link travel times can also be defined:

$$\Omega_t = \left\{\mathbf{t} \middle| t_a \in \left[t_a(\mathbf{0}), t_a\left(\mathbf{v}^{\max}\right)\right], a \in A\right\}, \tag{3.20}$$

where $\mathbf{0}$ is the link flow vector with zero traffic flow and vector $\mathbf{v}^{\max} = \left(v_a^{\max}, a \in A\right)^{\mathrm{T}}$. The feasible set of link flows Ω_v and the feasible set of link travel times Ω_t are non-empty, company and convex, and in addition:

$$\Omega_v^0 \subseteq \Omega_v \text{ and } \Omega_t^0 \subseteq \Omega_t . \tag{3.21}$$

By virtue of the SNL map and link travel time function, two vector functions are defined on the sets Ω_v and Ω_t, respectively:

$$\mathbf{F}_v(\mathbf{v}) = \left(F_{a,v}(\mathbf{v}), a \in A\right)^{\mathrm{T}} = \mathbf{v} - \boldsymbol{\psi}\left(\mathbf{t}(\mathbf{v})\right) : \Omega_v \to \Re^{|A|} \tag{3.22}$$

$$\mathbf{F}_t(\mathbf{t}) = \left(F_{a,t}(\mathbf{t}), a \in A\right)^{\mathrm{T}} = \mathbf{t} - \mathbf{t}\left(\boldsymbol{\psi}(\mathbf{t})\right) : \Omega_t \to \Re^{|A|} \tag{3.23}$$

***Proposition 3.1*:** Two vector functions $\mathbf{F}_t(\mathbf{t})$ and $\mathbf{F}_v(\mathbf{v})$ are Lipschitz-continuous over sets Ω_t and Ω_v, respectively.

Proof.

The link travel time function, $\mathbf{t}(\mathbf{v})$, and network loading map $\boldsymbol{\psi}(\mathbf{t})$ are continuous and continuously differentiable over the feasible set Ω_t and Ω_v respectively. Consequently, based on the property of compound functions, these two compound functions $\mathbf{F}_t(\mathbf{t})$ and $\mathbf{F}_v(\mathbf{v})$ are also continuous and continuously differentiable.

Thus, for function $\mathbf{F}_t(\mathbf{t})$, its Jacobian matrix $\frac{\partial \mathbf{F}_t(\mathbf{t})}{\partial \mathbf{t}}$ is continuous over set Ω_t. Since Ω_t is compact and convex, the 2-norm of the Jacobian matrix, $\frac{\partial \mathbf{F}_t(\mathbf{t})}{\partial \mathbf{t}}$, is bound from above, i.e. there exists a constant L, such that $\left\| \frac{\partial \mathbf{F}_t(\mathbf{t})}{\partial \mathbf{t}} \right\|_2 \le L$. Furthermore, according to the mean value theorem, there exists a vector $\upsilon = \mathbf{t}_1 + c\mathbf{t}_2, c \sim [0,1]$ such that

$$\left\| \mathbf{F}_t(\mathbf{t}_1) - \mathbf{F}_t(\mathbf{t}_2) \right\| \le \left\| \frac{\partial \mathbf{F}_t(\upsilon)}{\partial \mathbf{t}} \right\|_2 \left\| \mathbf{t}_1 - \mathbf{t}_2 \right\| \le L \left\| \mathbf{t}_1 - \mathbf{t}_2 \right\|, \tag{3.24}$$

where $\|\bullet\|$ denotes the Euclidean Norm for a vector. This shows that the vector function $\mathbf{F}_t(\mathbf{t})$ is Lipschitz continuous, and similarly the Lipschitz-continuity of $\mathbf{F}_v(\mathbf{v})$ can also be guaranteed. □

Based on the link flow-based vector function $\mathbf{F}_v(\mathbf{v})$, a VI model in the link flow space is built as: find a link flow vector $\mathbf{v}^* \in \Omega_v$ such that

$$\mathbf{F}_v(\mathbf{v})^{\mathrm{T}} (\mathbf{v} - \mathbf{v}^*) \ge 0, \forall \mathbf{v} \in \Omega_v . \tag{3.25}$$

In the link travel time space, the following VI model is proposed: find a link travel time vector $\mathbf{t}^* \in \Omega_t$ such that

$$\mathbf{F}_t(\mathbf{t})^T (\mathbf{t} - \mathbf{t}^*) \ge 0, \forall \mathbf{t} \in \Omega_t . \tag{3.26}$$

The following two propositions show that these two VI models are suitable formulations for the PA-SUEED problem. For the sake of presentation, the two VI models are denoted by $VI\left(\mathbf{F}_v,\Omega_v\right)$ and $VI\left(\mathbf{F}_t,\Omega_t\right)$, respectively

Proposition 3.2**:** $\mathbf{v}^*$ is a solution of $VI\left(\mathbf{F}_v,\Omega_v\right)$ if and only if $\mathbf{F}_v\left(\mathbf{v}^*\right)=\mathbf{0}$. In other words, the feasible link flow solution $\mathbf{v}^*$ fulfils the probit-based asymmetric SUE conditions with elastic demand.

Proposition 3.3**:** $\mathbf{t}^*$ is a solution of $VI\left(\mathbf{F}_t,\Omega_t\right)$ if and only if $\mathbf{F}_t\left(\mathbf{t}^*\right)=\mathbf{0}$. In other words, the feasible link flow solution $\boldsymbol{\psi}\left(\mathbf{t}^*\right)$ satisfies the probit-based SUE conditions with elastic demand.

Proof.

First, the necessary and sufficient conditions for Proposition 3.2 are presented as follows:

Necessary condition:

Let $\mathbf{v}_{SUE}$ be a SUE link flow solution of the probit-based asymmetric SUE problems with elastic demand. Thus,

$$\mathbf{v}_{SUE}=\boldsymbol{\psi}\left(\mathbf{t}\left(\mathbf{v}_{SUE}\right)\right). \tag{3.27}$$

Eqn. (3.27) implies that $\left(\mathbf{v}_{SUE}-\boldsymbol{\psi}\left(\mathbf{t}\left(\mathbf{v}_{SUE}\right)\right)\right)$ is a null vector. It therefore follows that

$$\left(\mathbf{v}_{SUE}-\boldsymbol{\psi}\left(\mathbf{t}\left(\mathbf{v}_{SUE}\right)\right)\right)^{\mathrm{T}}\left(\mathbf{v}-\mathbf{v}_{SUE}\right)=0,\forall\mathbf{v}\in\Omega_v. \tag{3.28}$$

In other words, $\mathbf{v}_{SUE}$ is a solution to VI model (3.25).

Sufficient condition:

Suppose that vector $\mathbf{v}^* = \left(v_1^*, v_2^*, \cdots, v_{|A|}^*\right)^{\mathrm{T}}$ is a solution to $VI\left(\mathbf{F}_v, \Omega_v\right)$. It is first proven that each element of the vector is positive, namely, $v_a^* > 0, \forall a \in A$. Without loss of generality, it is assumed that $v_1^* = 0$ and construct a particular feasible link flow vector:

$$\hat{\mathbf{v}} = \left(1, v_2^*, \cdots, v_{|A|}^*\right)^{\mathrm{T}} \in \Omega_v . \tag{3.29}$$

Substituting $\mathbf{v}$ in $VI\left(\mathbf{F}_v, \Omega_v\right)$ with $\hat{\mathbf{v}}$ yields that

$$\psi_1\left(\mathbf{t}\left(\mathbf{v}^*\right)\right) \le 0 . \tag{3.30}$$

The value of SNL map $\psi_1\left(\mathbf{t}\left(\mathbf{v}^*\right)\right)$ is always positive for probit-based SUE problem, which contradicts eqn. (3.30). In other words, any solution to $VI\left(\mathbf{F}_v, \Omega_v\right)$ is a positive vector.

Another two feasible link flow vectors $\mathbf{v}^1 \neq \mathbf{v}^*$, and $\mathbf{v}^2 \neq \mathbf{v}^*$ are employed, which follow this rule: (i) choose any particular link $b \in A$ and then set $v_b^1 = v_b^* + \bar{\lambda}$, $v_b^2 = v_b^* - \bar{\lambda}$ where $\bar{\lambda}$ is a given very small positive number in the interval $\left(0, v_b^*\right)$; (ii) traffic flow on the other links of these two new vectors are identical to that of $\mathbf{v}^*$. Substituting $\mathbf{v}$ in $VI\left(\mathbf{F}_v, \Omega_v\right)$ with $\mathbf{v}^1$ and $\mathbf{v}^2$ yield that

$$\left(v_b^* - \psi_b\left(\mathbf{t}\left(\mathbf{v}^*\right)\right)\right) \times \left(v_b^* + \bar{\lambda} - v_b^*\right) \ge 0 , \tag{3.31}$$

$$\left(v_b^* - \psi_b\left(\mathbf{t}\left(\mathbf{v}^*\right)\right)\right) \times \left(v_b^* - \overline{\lambda} - v_b^*\right) \geq 0 . \tag{3.32}$$

Eqns. (3.31) and (3.32) imply that

$$v_b^* = \psi_b\left(\mathbf{t}\left(\mathbf{v}^*\right)\right), \forall b \in A . \tag{3.33}$$

Eqn. (3.33) implies that $v_b^* - \psi_b\left(\mathbf{t}\left(\mathbf{v}^*\right)\right) = 0, \forall b \in A$, i.e. $\mathbf{F}_v\left(\mathbf{v}^*\right) = \mathbf{0}$ and it can be concluded that any solution of $VI\left(\mathbf{F}_v, \Omega_v\right)$ fulfils the SUE conditions with elastic demand.

Proposition 3 can be proven by using a similar procedure. □

The Lipschitz-continuity of two vector function $\mathbf{F}_t\left(\mathbf{t}\right)$ and $\mathbf{F}_v\left(\mathbf{v}\right)$ in conjunction with the nonempty, bounded, compact and convex properties of two sets Ω_t and Ω_v implies that $VI\left(\mathbf{F}_v, \Omega_v\right)$ and $VI\left(\mathbf{F}_t, \Omega_t\right)$ both have at least one solution (Corollary 2.2.5, Facchinei and Pang, 2003). Propositions 3.2 and 3.3 ensure that any solution of VI model (3.25) or (3.26) is a SUE link flow solution. According to Theorem 2 of Cantrarella (1997), this SUE link flow solution is unique, which is highlighted in Proposition 3.4.

***Proposition 3.4*:** VI models (3.25) and (3.26) both have unique solution.

3.4 Link-based Two-stage Monte Carlo Simulation for SNL

It can be seen that the SNL map plays a vital role in the fixed-point formulations and VI models developed for the PA-SUEED problem. Cantarella (1997) examined some fundamental mathematical properties of the SNL map. However, he did not provide a

computational method for the SNL map. Such a computational method is quite necessary when solving the PA-SUEED problem.

This section aims to provide a link-based Monte Carlo simulation method to solve the SNL problem. However, directly using the Monte Carlo simulation method would require path enumeration/generation, since the users' perception error on travel time ζ_{wk} is defined on paths. Hence, an alternative representation of the perception error is first used, which enables a link-based procedure.

3.4.1 An Alternative Representation of Perception Error

Sheffi (1985) showed that perceived *path* travel time can be derived from the normally distributed *link* perceived travel times. He assumed that perceived travel time on link $a \in A$, denoted by $T_a(\mathbf{v})$, has the following form:

$$\begin{aligned} T_a(\mathbf{v}) &= t_a(\mathbf{v}) + \xi_a, a \in A, \\ \xi_a &\sim N(0, \beta t_a^0), \end{aligned} \tag{3.34}$$

where error term ξ_a is a normally distributed random variable with zero mean and flow-independent variance equal to βt_a^0. Here, t_a^0 is the free flow travel time and constant β is termed as variance parameter. Clearly, the perception errors on different links are independent.

Thus, users' perceived travel time on path $k \in R_w$ can be given by:

$$\begin{aligned} C_{wk}(\mathbf{v}) &= \sum_{a\in A} T_a(\mathbf{v})\delta_{ak}^w \\ &= \sum_{a\in A} t_a(\mathbf{v})\delta_{ak}^w + \sum_{a\in A}\xi_a\delta_{ak}^w = c_{wk}(\mathbf{v}) + \sum_{a\in A}\xi_a\delta_{ak}^w, k\in R_w, w\in W \end{aligned} \tag{3.35}$$

Let $\zeta_{wk} = \sum_{a\in A}\xi_a\delta_{ak}^w, k\in R_w, w\in W$, and we can see that ζ_{wk} obtained here is also normally distributed with zero mean and constant variance, and moreover the covariance between the perception errors on two different paths equals to the total value of variances on their overlapping proportions (see Section 11.2 of Sheffi, 1985).

This link-based representation for the perceived path travel times enables link-based Monte Carlo simulation methods for solving the SNL map. In addition, sampling for each random variable ξ_a is independent, thus simplified the generation of random numbers. Accordingly, a link-based Monte Carlo simulation method is proposed in the following section.

3.4.1 Two-stage Monte Caro Simulation-based SNL Method

For the probit-based SUE problem with fixed demand, a Monte Carlo simulation-based SNL method can be found in Page 301 of Sheffi (1985). However, in this method proposed by Sheffi (1985), the elastic demand has not been taken into consideration. Thus, in this section, a two-stage procedure for the Monte Carlo simulation is proposed, where the first stage aims to estimate OD demands and the second stage calculates the estimated link flows, and it is summarized as follows:

Two-stage Monte Caro simulation based computational method

Input: Link travel time vector $\mathbf{t} = (t_a, a\in A)\in\Omega_t$

Output: Stochastic network loading map value $\mathbf{v}=\boldsymbol{\psi}(\mathbf{t})$

Step 1.0: (Initialization). Let the number of simulations $n=1$ and the initial estimated satisfaction $\bar{S}_w^{(0)}=0, w\in W$.

Step 1.1: (Sampling). For each link $a\in A$, sample a link travel time denoted by $\tilde{T}_a^{(n)}$ from the normal distributed population $N\left(t_a,\beta t_a^0\right)$.

Step 1.2: (Shortest path time calculation). With the link travel time pattern $\left\{\tilde{T}_a^{(n)}, a\in A\right\}$, calculate travel time of the shortest path between each OD pair $w\in W$, denoted by $\tilde{C}_w^{(n)}$, namely,

$$\tilde{C}_w^{(n)}=\min_{k\in R_w}\left(\tilde{c}_{wk}^{(n)}=\sum_{a\in A}\tilde{T}_a^{(n)}\delta_{ak}^w\right), w\in W. \tag{3.36}$$

Step 1.3: (Satisfaction estimation). Estimate the satisfaction for each OD pair $w\in W$ by the average scheme:

$$\bar{S}_w^{(n)}=\frac{(n-1)\bar{S}_w^{(n-1)}+\tilde{C}_w^{(n)}}{n}, w\in W. \tag{3.37}$$

Step 1.4: (Accuracy checking). If the number of iterations $n\geq n_0$, where n_0 is a predetermined sample size, go to Step 1.5; otherwise, set $n=n+1$ and go to Step 1.1.

Step 1.5: (OD demand calculation). Calculate OD travel demand pattern by the formulae:

$$q_w=D_w\left(\bar{S}_w^{(n)}\right), w\in W. \tag{3.38}$$

Go to Step 2.0

Step 2.0: (Initialization). Let the number of simulations $m=1$ and the initial estimated stochastic network loading map value $\bar{\psi}_a^{(0)}=0, a\in A$

Step 2.1: (Sampling). For each link $a\in A$, sample a link travel time denoted by $\tilde{T}_a^{(m)}$ from the normal distributed population $N\left(t_a,\beta t_a^0\right)$.

Step 2.2: (All-or-nothing traffic assignment).

(i) Define an initial OD pair based link flow solution:

$$y_{aw}^{(m)}=0, a\in A, w\in W . \tag{3.39}$$

(ii) With the link travel time pattern $\left\{\tilde{T}_a^{(m)}, a\in A\right\}$, find the shortest path for each OD pair w, then assign OD travel demand q_w calculated in Step 1.5 to each link of the shortest path, namely,

$$y_{aw}^{(m)}=q_w, \text{ for any link } a \text{ on the shortest path between OD pair } w\in W \tag{3.40}$$

(iii) Summing up traffic flow of each link yields the auxiliary link flow pattern $\left\{y_a^{(m)}=\sum_{w\in W} y_{aw}^{(m)}, a\in A\right\}$.

Step 2.3: (Stochastic network loading map estimation). Calculate the stochastic network loading map value by the average scheme:

$$\bar{\psi}_a^{(m)}=\frac{(m-1)\bar{\psi}_a^{(m-1)}+y_a^{(m)}}{m}, a\in A. \tag{3.41}$$

Step 2.4: (Stop criterion check). If the number of iterations $m\geq m_0$, where m_0 is the predetermined sample size, then stop and output the stochastic network

loading map value $\left\{\psi_a(\mathbf{t}) = \bar{\psi}_a^{(m)}, a \in A\right\}$. Otherwise, let $m = m + 1$ and go to step 2.1.

Both stages of the Monte Carlo simulation employ pseudo random numbers to sample the normally distributed link travel times. Note that the satisfaction estimator shown in eqn. (3.37) and the SNL map estimator expressed by (3.41) can be respectively rewritten as follows:

$$\bar{S}_w^{(n)} = \frac{\sum_{i=1}^{n} \tilde{C}_w^{(i)}}{n}, w \in W, \tag{3.42}$$

$$\bar{\psi}_a^{(m)} = \frac{\sum_{i=1}^{m} y_a^{(i)}}{m}, a \in A. \tag{3.43}$$

The two sample sizes n_0 and m_0 respectively used in Step 1.4 and Step 2.4 determines the accuracy of the computational algorithm, and their lower bounds with a given accuracy are elaborated in the following sub-section.

3.4.2 Sample Size Estimation

Firstly, the sample size n_0 for the first stage is addressed. Given a link travel time vector $\mathbf{t} = (t_a, a \in A) \in \Omega_t$, the relevant minimum perceived path travel time for OD pair $w \in W$ is a random variable, namely, $\min_{k \in R_w}\{C_{wk}\}$ is a random variable where $C_{wk} = \sum_{a \in A}(t_a + \xi_a)\delta_{ak}^w, k \in R_w$. It can be seen that the satisfaction $S_w\left((\Delta_w)^{\mathrm{T}}\mathbf{t}\right)$ is the expected value of the random variable $\min_{k \in R_w}\{C_{wk}\}$. Let σ_w^2 denote the variance of

$\min_{k \in R_w} \{C_{wk}\}$. When the sample size (i.e., the number of iterations n) is large enough, say $n \geq 30$ (Johnson and Wichern 2002), according to the central limit theorem, the average satisfaction estimator, $\overline{S}_w^{(n)}$ follows a normal distribution:

$$\overline{S}_w^{(n)} \sim N\left(S_w\left((\Delta_w)^{\mathrm{T}} \mathbf{t}\right), \frac{\sigma_w^2}{n} \right), w \in W . \tag{3.44}$$

The confidence interval of $\overline{S}_w^{(n)}$ at a significance level of 95% can be thus given by

$$\left[S_w\left((\Delta_w)^T \mathbf{t}\right) - 1.96\frac{\sigma_w}{\sqrt{n}}, \quad S_w\left((\Delta_w)^T \mathbf{t}\right) - 1.96\frac{\sigma_w}{\sqrt{n}} \right]. \tag{3.45}$$

Let $E_w = \left| S_w\left((\Delta_w)^T \mathbf{t}\right) - \overline{S}_w^{(n)} \right|$ denote the error of the unbiased satisfaction estimator $\overline{S}_w$. Thus,

$$E_w \leq 1.96\frac{\sigma_w}{\sqrt{n}}, w \in N . \tag{3.46}$$

Let ε_1 be the maximum error allowed for the satisfaction estimation, namely,

$$E_w \leq \varepsilon_1, \text{ for any } w \in W . \tag{3.47}$$

According to eqn. (3.46), the same the sample size can be taken as:

$$n \geq \max_{w \in W}\left(\frac{1.96^2 \times \sigma_w^2}{\varepsilon_1^2} \right), \tag{3.48}$$

where the population variance σ_w^2 can be estimated by the sample variance during the calculation, namely

$$\hat{\sigma}_w^{(n)} = \frac{\sum_{k=1}^{n}\left(\tilde{C}_w^{(k)} - \overline{S}_w^{(n)}\right)^2}{n-1}, w \in W . \tag{3.49}$$

In other words, the following criterion in Step 1.4 can be used:

$$n \geq n_0 = \max_{w \in W} \left(\frac{1.96^2 \times \hat{\sigma}_w^2}{\varepsilon_1^2} \right). \tag{3.50}$$

The lower bound for sample size required in the second stage is then analyzed. As it shows in Step 2.2, if link a is on the shortest path between OD pair w at iteration i, thus $y_{wa}^{(i)} = q_w$; otherwise, $y_{wa}^{(i)} = 0$. Eqn. (3.43) implies that

$$\bar{\psi}_a^{(m)} = \frac{\sum_{i=1}^{m} y_a^{(i)}}{m} = \sum_{i=1}^{m} \frac{\sum_{w \in W} y_{wa}^{(i)}}{m} = \sum_{w \in W} \frac{\sum_{i=1}^{m} y_{wa}^{(i)}}{m} = \sum_{w \in W} \frac{q_w K_{wa}}{m}, a \in A, \tag{3.51}$$

where $K_{wa} \leq m$ is the number of times that link a is on the shortest path between OD pair $w \in W$ among all the m samples.

It can be seen that K_{wa} / m in the right hand side of eqn. (3.51) is a *point estimation* of link usage probability $P_{wa}(\mathbf{t})$. Interestingly, K_{wa} is actually a binomial experiment with the number of trials m and the success probability $P_{wa}(\mathbf{t})$, abbreviated as P_{wa} henceforward. When the sample size is large enough ($mP_{wa} \geq 10$), K_{wa} can be approximated by a normal distribution (Johnson and Wichern, 2002), i.e.

$$K_{wa} \sim N\left(mP_{wa}, mP_{wa}\left(1 - P_{wa}\right)\right). \tag{3.52}$$

In light of eqn. (3.51), $\bar{\psi}_a^{(m)}$ is a linear combination of some normal distributed random variables. Hence, it also follows a normal distribution with the mean and variance as follows:

$$E\left(\bar{\psi}_a^{(m)}\right) = E\left(\sum_{w\in W}\frac{q_w K_{wa}}{m}\right) = \sum_{w\in W}\frac{q_w}{m}E\left(K_{wa}\right) = \sum_{w\in W} q_w P_{wa} = \psi_a\left(\mathbf{t}\right), a\in A, \quad (3.53)$$

$$\begin{aligned}\mathrm{var}\left(\bar{\psi}_a^{(m)}\right) = \mathrm{var}\left(\sum_{w\in W}\frac{q_w K_{wa}}{m}\right) &= \sum_{w\in W}\left(\frac{q_w}{m}\right)^2 \mathrm{var}\left(K_{wa}\right) \\ &+ \sum_{w_1,w_2\in W, w_1\neq w_2}\mathrm{cov}\left(K_{w_1 a}, K_{w_2 a}\right), a\in A\end{aligned}. \quad (3.54)$$

Since each OD demand are loaded independently for each iteration, the covariance proportion in the right hand side of eqn. (3.61) equals to zero. Eqn. (3.61) thence becomes

$$\mathrm{var}\left(\bar{\psi}_a^{(m)}\right) = \sum_{w\in W}\left(\frac{q_w}{m}\right)^2 \mathrm{var}\left(K_{wa}\right) = \sum_{w\in W} q_w^2 \frac{p_{wa}\left(1-p_{wa}\right)}{n}. \quad (3.55)$$

It should be pointed out that the stochastic network loading map estimation shown in eqn. (3.41) is an unbiased estimation according to eqn. (3.53)

Thereby, the confidence interval of the stochastic network loading map estimation $\bar{\psi}_a^{(m)}$ at a significance level of 95% can be given by

$$\left[\psi_a\left(\mathbf{t}\right) - 1.96\sqrt{\sum_{w\in W} q_w^2 \frac{P_{wa}\left(1-P_{wa}\right)}{n}},\quad \psi_a\left(\mathbf{t}\right) + 1.96\sqrt{\sum_{w\in W} q_w^2 \frac{P_{wa}\left(1-P_{wa}\right)}{n}}\right], a\in A. \quad (3.56)$$

Let $E_a = \left|\psi_a\left(\mathbf{t}\right) - \bar{\psi}_a^{(m)}\right|$ denote the error of the unbiased estimation $\bar{\psi}_a^{(m)}$. Eqn. (3.56) implies that

$$E_a \le 1.96\sqrt{\sum_{w\in W} q_w^2 \frac{P_{wa}\left(1-P_{wa}\right)}{m}}, a\in A. \quad (3.57)$$

Let ε_2 be the maximum error allowed for the satisfaction estimation, namely,

$$E_a \le \varepsilon_2, a\in A \quad (3.58)$$

According to eqn. (3.57), a lower bound for the sample size, m_0, can be shown as follows.

$$m \geq m_0 = \max_{a \in A} \left[\frac{1.96^2 \times \sum_{w \in W} q_w^2 P_{wa} \left(1 - P_{wa}\right)}{\varepsilon_2^2} \right]. \tag{3.59}$$

3.5 Three Solution Algorithms

3.5.1 Two Projection-type Self-adaptive Prediction-Contraction Algorithms

The projection-type methods have been commonly used for solving VI models (see, Bertsekas, 1976; Bertsekas and Gafni, 1982; He, 1992 & 1994; Nagurney, 1993; Patriksson, 1994b; Chen et al., 2001; Bekhor and Toledo, 2005, etc.). Since the feasible sets of the two proposed models $VI\left(\mathbf{F}_v, \Omega_v\right)$ and $VI\left(\mathbf{F}_t, \Omega_t\right)$ are Cartesian products of intervals (Meng et al., 2008), the calculation of a projection on such type of feasible sets is largely simplified, thus it is convenient to adopt projection-type methods as solution algorithms. Two Prediction-Contraction (PC) algorithms, the conventional PC algorithm and the self-adaptive PC algorithm, are first introduced as follows.

Let Ω be a nonempty closed convex subset of $\Re^n$, and let $\mathbf{F}$ be a continuous vector mapping from Ω to itself. A general expression for VI models can be given by

$$\mathbf{F}\left(\mathbf{x}^*\right)^{\mathrm{T}} \left(\mathbf{x} - \mathbf{x}^*\right) \geq 0, \forall \mathbf{x} \in \Omega. \tag{3.60}$$

Given a vector $\mathbf{y} \in \Re^n$, its projection onto the set Ω, denoted by $P_\Omega\left(\mathbf{y}\right)$, is defined by

$$P_\Omega\left[\mathbf{y}\right] = \arg\min_{\mathbf{x} \in \Omega} \left(\left\|\mathbf{x} - \mathbf{y}\right\|\right). \tag{3.61}$$

Note that when set $\Omega = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$, which is Cartesian products of intervals, the calculation of projection $\overline{\mathbf{y}} = P_\Omega[\mathbf{y}] = (y_i, i = 1, 2, \cdots, n)^{\mathrm{T}}$ becomes straightforward:

$$\overline{y}_i = \begin{cases} a_i, & y_i < a_i \\ y_i, & a_i \le y_i \le b_i, \\ b_i, & y_i > b \end{cases} \quad i = 1, 2, \cdots, n. \tag{3.62}$$

The conventional PC algorithm, also referred as Korpelevich-Khobotov method (Korpelevich, 1976), works as follows:

Conventional PC Method

Step 0: (Initiation). Set the initial step size $\eta^{(0)} = 1$, parameter $\nu \in (0,1)$, an initial solution $\mathbf{x}^{(0)} \in \Omega$

Step 1: (Prediction). Compute the projection:

$$\overline{\mathbf{x}}^{(n+1)} = P_\Omega\left[\mathbf{x}^{(n)} - \eta^{(n)}\mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]. \tag{3.63}$$

Step 2: (Stop criterion). If $\left\|e\left(\mathbf{x}^{(n)}\right)\right\| = \left\|\mathbf{x}^{(n)} - P_\Omega\left[\mathbf{x}^{(n)} - \eta^{(n)}\mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]\right\| = \left\|\overline{\mathbf{x}}^{(n+1)} - \mathbf{x}^{(n)}\right\| \le \varepsilon_{KK}$, where ε_{KK} is a predetermined tolerance, stop; otherwise, go to Step 3.

Step 3: (Step size adjustment). Perform the following loop to find a proper step size:

Step 3.1: Calculate the ratio:

$$r^{(n)} = \eta^{(n)}\left\|\mathbf{F}\left(\mathbf{x}^{(n)}\right) - \mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right)\right\| \Big/ \left\|\mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)}\right\|. \tag{3.64}$$

Step 3.2: If $r^{(n)} \le \nu$, let $\eta^{(n+1)} = \eta^{(n)}$ go to Step 4;otherwise go to Step 3.3.

Step 3.3: Reduce the step size using the scheme:

$$\eta^{(n)} = \frac{2}{3}\eta^{(n)} \min\left\{1, \frac{1}{r^{(n)}}\right\}. \tag{3.65}$$

Step 3.4: Update the prediction projection:

$$\overline{\mathbf{x}}^{(n+1)} = P_{\Omega}\left[\mathbf{x}^{(n)} - \eta^{(n)}\mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]. \tag{3.66}$$

Go to Step 3.1

Step 4: (Correction). Based on the feasible solution $\overline{\mathbf{x}}^{(n+1)} \in \Omega$ and the step size $\eta^{(n+1)}$ obtained in Step 3, calculate another projection:

$$\mathbf{x}^{(n+1)} = P_{\Omega}\left[\mathbf{x}^{(n)} - \eta^{(n+1)}\mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right)\right]. \tag{3.67}$$

Set $n = n+1$ and go to Step 1.

If the vector function $\mathbf{F}(\mathbf{x})$ is Lipschitz continuous, a step size $\eta^{(n)} > 0$ can be found such that $r^{(n)} \le \nu$ in finite iterations (He and Liao, 2002). In other words, the step size reduction loop of Step 3 will be terminated within limited iterations. More importantly, with the Lipschitz-continuity, it is easy to show that

$$\inf\left\{\eta^{(n)}\right\} = \eta_{\min} > 0. \tag{3.68}$$

The stop criterion in Step 2 comes from a fact that $\mathbf{x}^*$ is the solution of the VI model (3.60) if and only if the error bound

$$\left\|e\left(\mathbf{x}^*\right)\right\| = \left\|\mathbf{x}^* - P_{\Omega}\left[\mathbf{x}^* - \rho\mathbf{F}\left(\mathbf{x}^*\right)\right]\right\| = 0, \tag{3.69}$$

where ρ is an any positive number.

It can be seen that the step size sequence generated by Step 3 of the conventional PC method is monotonically decreasing due to the step size reduction scheme shown by eqn. (3.65). To improve the convergent speed of this conventional PC method, two modifications can be performed on the step size from two aspects (He and Liao, 2002):

(a) For the step size of correction step: Instead of using the step size $\eta^{(n+1)}$ obtained from step 3, determine a more robust step size $\alpha^{(n+1)}$ based on $\eta^{(n+1)}$ and $\mathbf{x}^{(n)}$;

(b) For the step size of prediction step: enlarge $\eta^{(n+1)}$ if it is relatively too small.

Firstly, to illustrate the improvement (a), the following function is defined in the first place:

$$\mathrm{M}\left(\alpha^{(n+1)}\right)=\left\|\mathbf{x}^{(n)}-\mathbf{x}^{*}\right\|^{2}-\left\|\mathbf{x}^{(n+1)}\left(\alpha^{(n+1)}\right)-\mathbf{x}^{*}\right\|^{2}, \tag{3.70}$$

where $\left\|\mathbf{x}^{(n)}-\mathbf{x}^{*}\right\|^{2}$ denotes the distance from $\mathbf{x}^{(n)}$ to the optimum $\mathbf{x}^{*}$, and the mapping $\mathbf{x}^{(n+1)}\left(\alpha^{(n+1)}\right)$ can be obtained from the correction step in terms of any step size value $\alpha^{(n+1)}$:

$$\mathbf{x}^{(n+1)}\left(\alpha^{(n+1)}\right):=P_{\Omega}\left[\mathbf{x}^{(n)}-\alpha^{(n+1)}\mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right)\right]. \tag{3.71}$$

Korpelevich (1976) claimed that when $\alpha^{(n+1)}=\eta^{(n+1)}$ function $\mathrm{M}\left(\alpha^{(n+1)}\right)$ would be positive at each step. Only recently, He and Liao (2002) proved that $\mathrm{M}\left(\alpha^{(n+1)}\right)$ reaches its maximum when $\alpha^{(n+1)}$ takes the following value:

$$\alpha^{(n+1)*} = \eta^{(n+1)} \frac{\left(\mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)}\right)^T \left(\mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)} - \eta^{(n+1)} \left(\mathbf{F}\left(\mathbf{x}^{(n)}\right) - \mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right)\right)\right)}{\left\|\mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)} - \eta^{(n+1)} \left(\mathbf{F}\left(\mathbf{x}^{(n)}\right) - \mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right)\right)\right\|^2}. \quad (3.72)$$

This equation gives the ideal step size, $\alpha^{(n+1)*}$, in the correction step based on given $\mathbf{x}^{(n)}$ and $\eta^{(n+1)}$. According to He and Liao (2002), a relaxation factor $\gamma \in (0,2)$ is usually taken for $\alpha^{(n+1)}$, i.e.

$$\alpha^{(n+1)} = \alpha^{(n+1)*} \times \gamma .$$

Regarding improvement (b), since step size $\eta^{(n)}$ is monotonically non-increasing, the convergent rate may inherently decelerate if $\eta^{(n)}$ is too small. Therefore, $\eta^{(n)}$ is slightly enlarged if it is relatively small enough. The self-adaptive PC algorithm developed by He and Liao (2002) incorporates the aforementioned two step size improvements, which is presented below.

Self-adaptive Prediction-Correction Algorithm

Step 0: (Initiation). Set initial step size $\eta^{(0)} = 1$, parameter $\nu \in (0,1)$, step size relation factor $\gamma \in (0,2)$ and an initial solution $\mathbf{x}^{(0)} \in \Omega$.

Step 1: (Prediction). Compute the projection:

$$\overline{\mathbf{x}}^{(n+1)} = P_\Omega \left[\mathbf{x}^{(n)} - \eta^{(n)} \mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]. \quad (3.73)$$

Step 2: (Stop criterion). If $\left\|e\left(\mathbf{x}^{(n)}\right)\right\| = \left\|\mathbf{x}^{(n)} - P_\Omega \left[\mathbf{x}^{(n)} - \eta^{(n)} \mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]\right\| = \left\|\mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)}\right\| \le \varepsilon$, where ε is a predetermined tolerance, stop; Otherwise, go to Step 3.

Step 3: (Step size adjustment). Perform the following loop to find a proper step size:

Step 3.1: Calculate the ratio:

$$r^{(n)} = \eta^{(n)} \left\| \mathbf{F}\left(\mathbf{x}^{(n)}\right) - \mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right) \right\| \Big/ \left\| \mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)} \right\|. \tag{3.74}$$

Step 3.2: If $r^{(n)} \le \nu$, let $\eta^{(n+1)} = \eta^{(n)}$ go to Step 4;otherwise go to Step 3.3.

Step 3.3: Reduce the step size using the scheme:

$$\eta^{(n)} = \frac{2}{3}\eta^{(n)} \min\left\{1, \frac{1}{r^{(n)}}\right\}. \tag{3.75}$$

Step 3.4: Update the prediction projection:

$$\overline{\mathbf{x}}^{(n+1)} = P_{\Omega}\left[\mathbf{x}^{(n)} - \eta^{(n)}\mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]. \tag{3.76}$$

and go to Step 3.1.

Step 4: (Correction). (i) Estimate step size $\alpha^{(n+1)}$:

$$\alpha^{(n+1)} = \gamma \times \alpha^{(n+1)^*}. \tag{3.77}$$

(ii) Based on the feasible solution $\overline{\mathbf{x}}^{(n+1)} \in \Omega$ and the step size $\alpha^{(n+1)}$, calculate another projection:

$$\mathbf{x}^{(n+1)} = P_{\Omega}\left[\mathbf{x}^{(n)} - \alpha^{(n+1)}\mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right)\right]. \tag{3.78}$$

(iii) Enlarge the step size:

$$\eta^{(n+1)} = \frac{3}{2}\eta^{(n+1)}, \text{ if } r^{(n)} \le \tau; \text{ and } \eta^{(n+1)} = \eta^{(n+1)}, \text{ otherwise}$$

where $\tau \in (0, \nu)$ is a predetermined parameter.

Set $n = n+1$ and go to Step 1.

The difference between the conventional PC algorithm and the self-adaptive PC algorithm lies in Step 4. Step 4 of the latter algorithm reflects the two step size improvements (a) and (b).

3.5.2 Cost-Averaging Algorithm

Eqn. (3.79) shows the recursive function of the conventional MSA method, and when used for solving the PA-SUEED problem, its convergence cannot be guaranteed.

$$\mathbf{v}^{(n+1)} = \mathbf{v}^{(n)} + \frac{1}{n}\left(\boldsymbol{\psi}\left(\mathbf{t}\left(\mathbf{v}^{(n)}\right)\right) - \mathbf{v}^{(n)}\right). \tag{3.79}$$

Cost-averaging (CA) algorithm developed by Cantarella (1997) is the only convergent algorithm in the existing literature for solving the PA-SUEED problem. It is in fact a variation of the MSA. Recursive function of the CA method is given by:

$$\mathbf{t}^{(n+1)} = \mathbf{t}^{(n)} + \frac{1}{n}\left(\mathbf{t}\left(\boldsymbol{\psi}\left(\mathbf{t}^{(n)}\right)\right) - \mathbf{t}^{(n)}\right). \tag{3.80}$$

where the SNL map $\boldsymbol{\psi}\left(\mathbf{t}^{(n)}\right)$ can be calculated by the two-stage Mont Carlo simulation-based method proposed by Section 3.4.

3.5.3 Two Hybrid Prediction-Correction Algorithms

The self-adaptive PC algorithm incorporating the two-stage Monte Carlo simulation based method can be employed for solving two VI models, shown by eqns. (3.25) and (3.26). As the former VI model is defined on the link flow space, the self-adaptive PC algorithm solving this VI model is named FPC algorithm. Similarly, CPC is the abbreviation of the self-adaptive PC algorithm solving the latter VI model defined on the link travel time (or cost) space.

It should be noted that the self-adaptive PC algorithm may encounter a dilemma in practical implementation: step size $\eta^{(n)}$ calculated by Step 3.1 may be extremely small due to the step size reduction scheme shown in eqn. (3.75). In this case, projection operation $\overline{\mathbf{x}}^{(n+1)} = P_{\Omega}\left[\mathbf{x}^{(n)} - \eta^{(n)}\mathbf{F}\left(\mathbf{x}^{(n)}\right)\right]$ of Step 3.4 makes $\overline{\mathbf{x}}^{(n+1)}$ and $\mathbf{x}^{(n)}$ very close, and they may be regarded as identical by the computer. This phenomenon undermines the step size adjustment, since in Step 3.1:

$$r^{(n)} = \eta^{(n)} \left\| \mathbf{F}\left(\mathbf{x}^{(n)}\right) - \mathbf{F}\left(\overline{\mathbf{x}}^{(n+1)}\right) \right\| \Big/ \left\| \mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)} \right\|.$$

The denominator on the right hand side of this equation would be equal to zero, and the PC algorithm is thus disabled.

With a view to overcome this problem, two hybrid PC-CA algorithms are proposed: when denominator of $r^{(n)}$ is small enough:

$$\left\| \mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)} \right\| \le \varepsilon_0 , \tag{3.81}$$

where ε_0 is a predetermined trivial positive value, the FPC and CPC algorithm would be terminated, and then the iteration is continuously carried on by CA algorithm. These new procedures are named as hybrid FPC-CA and hybrid CPC-CA respectively, whose first part is the FPC or CPC algorithm with self-adaptive step size and second part is CA method with predetermined step sizes. Clearly, convergence of the hybrid FPC-CA and hybrid CPC-CA can be theoretically guaranteed.

An integrated procedure of the two hybrid PC-CA algorithms can be obtained by inserting the following Step 3.0 and Step 5 to the Self-adaptive PC Algorithm shown in Section 5.1:

Step 3.0: If $\left\| \mathbf{x}^{(n)} - \overline{\mathbf{x}}^{(n+1)} \right\| \le \varepsilon_0$, then go to step 5; otherwise go to step 3.1

Step 5 : $\mathbf{t}^{(n+1)} = \mathbf{t}^{(n)} + \frac{1}{n}\left(\mathbf{t}\left(\boldsymbol{\psi}\left(\mathbf{t}^{(n)} \right) \right) - \mathbf{t}^{(n)} \right)$.

The recursive function in Step 5 would be iteratively calculated until the stop criterion could be fulfilled. In addition, on the end of Step 3.4, it should be "go to Step 3.0".

3.6 Numerical experiments

Two examples with asymmetric link travel time functions are used to validate the proposed VI models and compare the performance between the two hybrid PC-CA algorithms and CA algorithm. The first example is from Yang (1995) and the second example is a modified Sioux-Fall network (Bar-Gera, 2011).

The variance parameter β in eqn. (3.34) is set to be 0.1 and the tolerance value, ε_0, in eqn. (3.81) is set to be 1×10^{-15}. As to the demand functions, it is assumed that demand between each OD pair $w \in W$ is determined by the following exponential function:

$$q_w = \overline{q}_w \exp\left(-\mu S_w\left(\mathbf{c}_w \right) \right), \tag{3.82}$$

where μ is a constant parameter.

Propositions 3.2 and 3.3 imply that $\mathbf{v}^*$ or $\boldsymbol{\psi}\left(\mathbf{t}^* \right)$ is the probit-based SUE link flow solution if and only if the Euclidean norm of vector $\mathbf{F}_v\left(\mathbf{v}^* \right)$ or $\mathbf{F}_t\left(\mathbf{t}^* \right)$ is equal to zero,

namely, $\left\|\mathbf{F}_v\left(\mathbf{v}^*\right)\right\|=0$ or $\left\|\mathbf{F}_t\left(\mathbf{t}^*\right)\right\|=0$. To compare performance of the three solution algorithms, it is tested whether the Euclidean norm of vector function $\mathbf{F}_v\left(\mathbf{v}^{(n)}\right)$ or $\mathbf{F}_t\left(\mathbf{t}^{(n)}\right)$ is close to zero at iteration n, which is equivalent to the stop criterion defined by the error bound function shown in eqn. (3.69). Stop criteria for the three algorithms are thus defined based on the relative error of $\left\|\mathbf{F}_v\left(\mathbf{v}^{(n)}\right)\right\|$ or $\left\|\mathbf{F}_t\left(\mathbf{t}^{(n)}\right)\right\|$:

For hybrid FPC-CA algorithm in the link flow space:

$$\text{Relative error value}=\frac{\sqrt{\sum_{a\in A}\left(v_a^{(n)}-\psi_a\left(\mathbf{t}\left(\mathbf{v}^{(n)}\right)\right)\right)^2}}{\sqrt{\sum_{a\in A}\left(v_a^{(n)}\right)^2}}\le\varepsilon_f. \tag{3.83}$$

For hybrid CPC-CA and CA algorithms in the link travel cost space:

$$\text{Relative error value}=\frac{\sqrt{\sum_{a\in A}\left(t_a^{(n)}-t_a\left(\boldsymbol{\psi}\left(\mathbf{t}^{(n)}\right)\right)\right)^2}}{\sqrt{\sum_{a\in A}\left(t_a^{(n)}\right)^2}}\le\varepsilon_c. \tag{3.84}$$

where ε_f and ε_c are the positive tolerance values.

It should be noted that most of the computational efforts of these three algorithms (hybrid FPC-CA, hybrid CPC-CA and CA) are devoted into calculating the SNL, so that CPU time used by each of the three algorithms depends on how many times they have invoked the two-stage Monte Carlo simulation based method. This number is called as the total number of SNLs, which can be used as an alternative index to reflect the CPU time.

3.6.1 Example 1

Example 1 (Yang, 1995), shown in Figure 3.1, consists of 6 nodes, 7 links and 2 OD pairs: $1 \to 3$ and $2 \to 4$. It uses the link travel time functions:

$$t_1(\mathbf{v}) = 2 + \frac{v_1^2}{100} + \frac{v_2^2}{200},\ t_2(\mathbf{v}) = 3 + \frac{v_2^2}{100} + \frac{v_1^2}{200},\ t_3(\mathbf{v}) = 10 + \frac{v_3^2}{100} + \frac{v_6^2}{200},\ t_4(\mathbf{v}) = 4 + \frac{v_4^2}{400},$$

$$t_5(\mathbf{v}) = 9 + \frac{v_5^2}{100} + \frac{v_7^2}{200}, t_6(\mathbf{v}) = 2 + \frac{v_6^2}{100} + \frac{v_3^2}{200},\ t_7(\mathbf{v}) = 4 + \frac{v_7^2}{100} + \frac{v_5^2}{200}.$$

As for the OD demand function shown in eqn. (3.82), the following parameters are used: $\mu = 0.1,\ \bar{q}_{1\to3} = \bar{q}_{2\to4} = 30$.

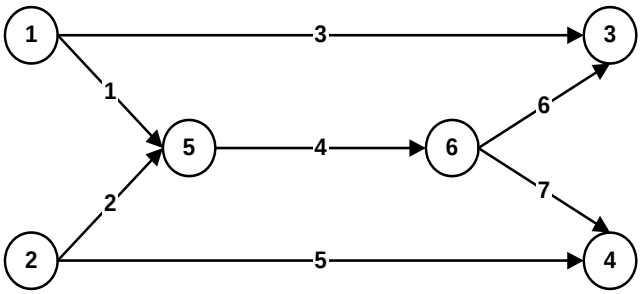

Figure 3.1 Network topology of Example 1

3.6.1.1. Determination of sample sizes

The estimate sample sizes required by the two-stage Monte Carlo simulation based method for calculating the SNL map are first estimated. For the first stage, eqn. (3.50) is taken as the stop criterion by assuming that the maximum error $\varepsilon_1 = 0.5$. Since estimation of population variance $\hat{\sigma}_w^2$, shown by eqn. (3.49), depends on the number of iterations used in the first stage, lower bound for sample size, n_0, of this stage thus cannot be determined in advance.

Lower bound for the sample size of the second stage, m_0, can be determined by eqn. (3.59) before computations. Since $P_{wa}(1-P_{wa}) \le 0.25$, it follows that

$$m_0 = \max_{a \in A}\left[\frac{1.96^2 \times \sum_{w \in W} q_w^2 P_{wa}(1-P_{wa})}{\varepsilon_2^2}\right] \le \frac{1.96^2 \times 0.25 \times \sum_{w \in W} \bar{q}_w^2}{\varepsilon_2^2}. \tag{3.85}$$

Let the maximum error $\varepsilon_2 = 0.5$, and it gives $m_0 \le 768$. Thus, $m_0 = 800$ is taken as sample size for the second stage of Monte Carlo simulation.

3.6.1.2 Computational Results

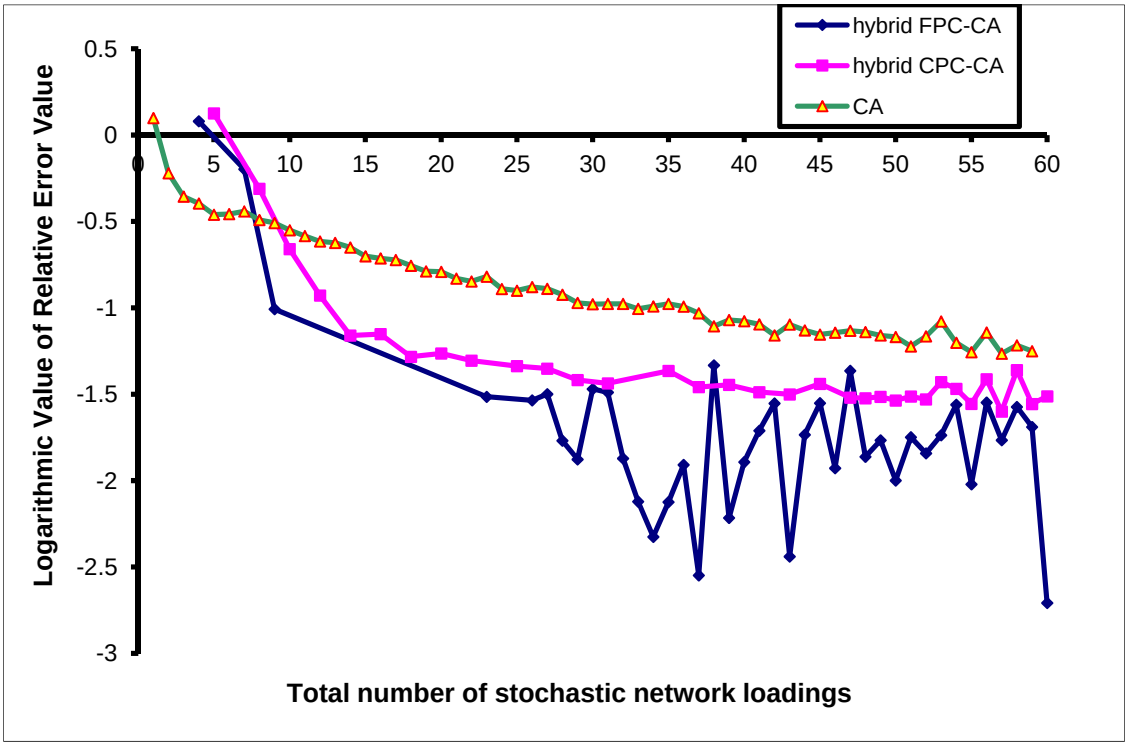

Figure 3.2 Convergent Trends of the Three Algorithms for Solving Example 1

Figure 3.2 depicts the change of relative error values (in logarithmic values) for all three algorithms within 60 SNLs. It shows that (a) the relative error values for all three algorithms are decreasing when the total number of SNL increases and finally approach

to and fluctuate surround a trivial value; (b) the two hybrid PC-CA algorithms both outperform CA algorithm in terms of convergent speed.

To further analyze characteristics of the two hybrid PC-CA algorithms, Figures 3.3 and 3.4 intuitively illustrate their hybrid properties. According to Figure 3.3, it can be seen that after 26 SNLs the FPC part is terminated and CA part takes over the computation for hybrid FPC-CA algorithm. While, Figure 3.4 shows CPC part is stopped after 49 SNLs for hybrid FPC-CA algorithm. It is noted, in passing, that after termination of FPC and CPC, their CA part successively carry on the calculation with a step size equal to reciprocal of the cumulative total number of SNLs. Namely, for this example, step size of CA part, ($\frac{1}{n}$), of FPC-CA and CPC-CA begins with $\frac{1}{27}$ and $\frac{1}{50}$, respectively.

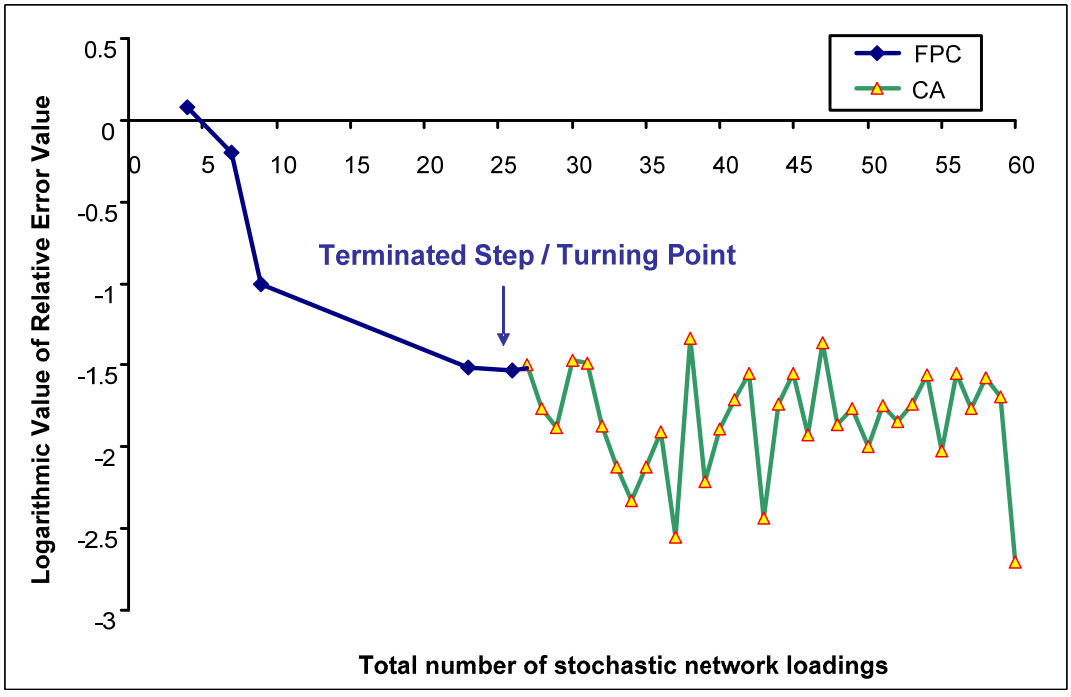

Figure 3.3 Transformation between the FPC and CA Part of Hybrid FPC-CA Algorithm

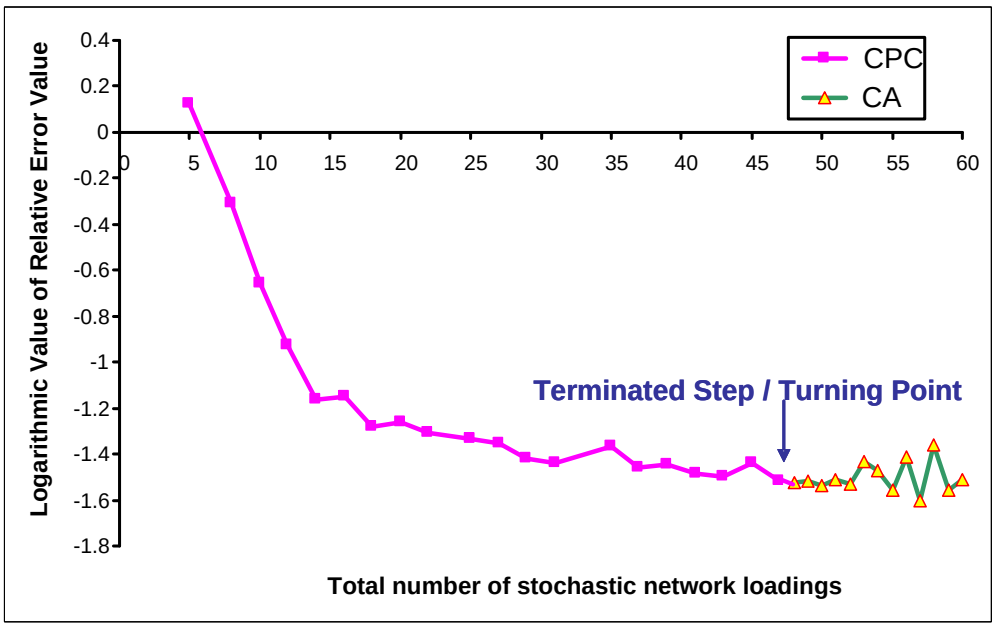

Figure 3.4 Transformation between the CPC and CA Part of Hybrid CPC-CA Algorithm

Table 3.1 gives the CPU time used by the three algorithms for solving Example 1 when different tolerance values are adopted (ε_f for hybrid FPC-CA, while ε_c for CA and hybrid CPC-CA). According to Table 3.1, for any given accuracy level computational time used by the two hybrid PC-CA algorithms is significantly less than that used by the CA algorithm. In particular, for the highest accuracy level $\varepsilon_f = 0.20\%$ or $\varepsilon_c = 0.20\%$, CPU time used by the FPC-CA (CPC-CA) is only 11% (45%) of the time consumed by conventional CA algorithm.

It is worthy pointing out that conventional CA and CA parts of two hybrid algorithms both suffer an erratic fluctuation when the value of relative error achieves 3%. Two CA parts of hybrid algorithms perform better than conventional CA mainly because a

superior initial solution is provided. For this example, CA part can, on average, only slightly improve the accuracy level after termination of PC part, since values of relative error obtained by terminated FPC and CPC are 2.92% and 3.06%, while average value of the relative errors (100 loadings after termination step) obtained by their successive CA part is 1.77% and 2.55%.

Table 3.1 CPU Time of Three Algorithms for Solving Example 1

Tolerance Value ε_f or ε_c	CA (Seconds)	Hybrid FPC-CA (Seconds)	Hybrid CPC-CA (Seconds)
10%	0.45	0.14	0.16
5%	0.85	0.21	0.27
1%	4.33	0.63	2.19
0.50%	9.69	1.58	4.43
0.20%	29.50	3.13	13.30

3.6.1.3 Accuracy investigation of the solutions when the PC parts are terminated

Figures 3.3 and 3.4 both show that the PC parts of two hybrid algorithms terminate after a number of SNLs. This sub-section intends to verify that the resultant solutions obtained by the terminated PC parts are already close enough to the solution of PA-SUEED problem. Tables 3.2 and 3.3 list link flow, link travel time and OD demand values obtained by the terminated PC parts (FPC after 26 SNLs and CPC after 49 SNLs.), respectively. The column entitled "SUE solution" of Tables 3.2 and 3.3 is the final results generated by the hybrid FPC-CA algorithm, which is regarded as a benchmark.

Table 3.2 Link Flows/Times Generated by Two Terminated PC Parts and SUE Solutions for Example 1

Link sequence	Terminated FPC		Terminated CPC		SUE solution	
	Flow	**Time**	**Flow**	**Time**	**Flow**	**Time**
1	13.56	3.02	13.61	3.04	13.59	3.02
2	8.65	3.25	8.71	3.26	8.63	3.25
3	13.47	10.98	13.42	10.96	13.46	10.97
4	22.21	4.91	22.31	4.92	22.23	4.92
5	18.08	11.85	18.03	11.86	18.10	11.86
6	13.56	3.02	13.61	3.04	13.59	3.02
7	8.65	4.34	8.71	4.34	8.63	4.33

Table 3.3 OD demands Generated by the Two Terminated PC Parts and SUE Solution for Example 1

OD Pair	Terminated FPC	Terminated CPC	SUE solution
$1 \to 3$	27.03	27.03	27.12
$2 \to 4$	26.73	26.74	26.81

The relative errors between solutions from the two terminated PC parts and the SUE solution are used to measure the difference of these three sets of results, which are calculated as follows: $E_a^v = \left| \frac{v_a^{PC} - v_a^{SUE}}{v_a^{SUE}} \right| \times 100\%, a \in A$ for link flows; $E_a^t = \left| \frac{t_a^{PC} - t_a^{SUE}}{t_a^{SUE}} \right| \times 100\%, a \in A$ for link travel time; $E_w^q = \left| \frac{q_w^{PC} - q_w^{SUE}}{q_w^{SUE}} \right| \times 100\%, w \in W$ for OD demand, where the superscript PC stands for the two PC parts and SUE represents the SUE solution. The maximal and average values of these relative errors among all the links/OD-pairs are calculated by

$$\hat{E}^v = \max\left\{E_a^v, a \in A\right\},\ \bar{E}^v = \frac{\sum_{a \in A} E_a^v}{|A|};\ \hat{E}^t = \max\left\{E_a^t, a \in A\right\},\ \bar{E}^t = \frac{\sum_{a \in A} E_a^t}{|A|}$$

$$\hat{E}^q = \max\left\{E_w^q, w \in W\right\},\ \bar{E}^q = \frac{\sum_{w \in W} E_w^q}{|W|}.$$

The maximum and average values are tabulated in Table 3.4. It shows that link flow/travel-time and OD demand values generated by the two terminated PC parts are quite close to the SUE solution. In other words, the results obtained by the terminated FPC and CPC parts are already accurate enough for practical applications.

Table 3.4 Differences between the Results of Two Terminated PC Parts and the SUE Solution for Example 1

	Link cost		**Link flow**		**OD demand**	
	$\hat{E}^v$	$\overline{E}^v$	$\hat{E}^t$	$\overline{E}^t$	$\hat{E}^q$	$\overline{E}^q$
FPC	0.459%	0.249%	0.460%	0.349%	0.332%	0.315%
CPC	0.248%	0.121%	0.666%	0.258%	0.332%	0.296%

As shown in Figure 3.2, in the first 15 loadings, two hybrid algorithms possess a faster convergent rate than that of conventional CA. However, after 15 loadings and before the termination of PC parts (e.g. 15 to 49 loadings of CPC-CA) convergent rate of hybrid algorithms evidently slows down, this is because of the computational error of Monte Carlo simulation deteriorates the accuracy of self-adaptive step size mechanism of PC algorithm. Yet, it shows that the accuracy level (value of relative error) obtained at 15 loadings are almost the same with that obtained at the termination of PC parts, thus indicates that when computational speed of PC parts is decelerated, the solution obtained is already accurate enough for practical applications.

3.6.1.4 Sensitivity Test for the Sample Size of the Second-stage Monte Carlo Simulation

Based on many numerical experiments, the estimator in first stage of Monte Carlo simulation all converge quite fast (in around 50 iteration), and the required sample size

for first stage thus is usually quite small (less than 100). However, the estimator in the second stage is in general not stable, and it usually takes hundreds of or even thousands of simulations to converge. Thus, in this sub-section, the estimator for lower bound of the sample size, m_0, for the second-stage is verified using the Example 1.

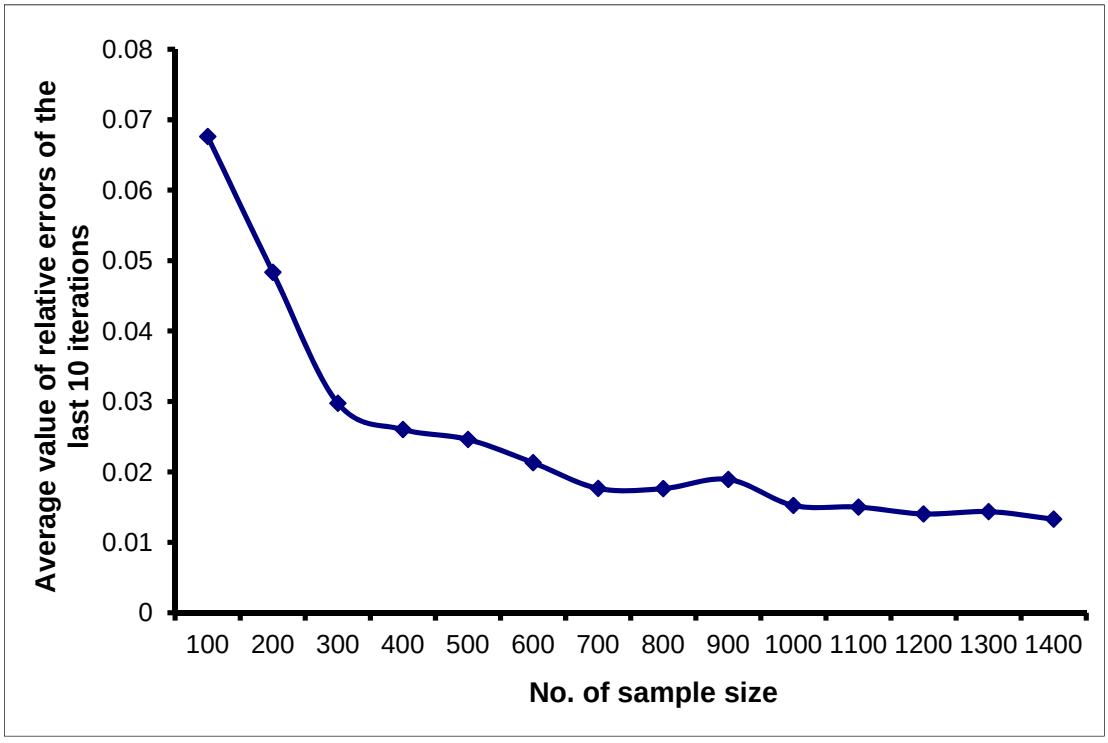

Figure 3.5 Sensitivity Test for the Sample Size of Monte Carlo Simulation

Based on the foregoing analysis, m_0 is determined to be 800 for Example 1. Two tests are conducted here to estimate the impact of this sample size on the accuracy of SUE solutions. Firstly, the sample size m_0 for different experiments is adjusted and the accuracy of corresponding computational results is tested. Without loss of generality, the hybrid CPC-CA algorithm is utilized for this test. Herein, the algorithm is run for 1000

iterations for each of 14 different experiments with simple sizes ranging from 100 to 1400 in each iteration. The average of relative error values in the last 10 iterations is adopted to represent the numerical convergence of each experiment, which is depicted by Figure 3.5. Figure 3.5 indicates that (a) when sample size is less than 400, it largely influences the accuracy, and the accuracy is greatly improved as the sample size increases, which coheres with the study of Daganzo (1979, pp 49-51); (b) When sample size is larger than 600, the improvement in terms of the accuracy is quite modest. This result shows that sample size m_0 is a relatively loose bound, thus a sample size of $m_0 = 800$ is more than sufficient, and it allows some proper relaxations for practical implementations.

Secondly, due to the size of this small-scale example (only 2 paths for each OD pair), a closed-form expression is available for the path choice probability. Thus, based on the OD demand obtained from the first-stage of Monte Carlo simulation method, the link flows can be theoretically calculated. Another test of this sub-section aims to contrast these theoretical link flows with the simulation results, to test the accuracy of the second-stage of Monte Carlo simulation. The computational results of terminated FPC, which is the "SUE solution" in Table 3.2, are adopted for this test. It should be noted that the link flow values provided in the first column of Table 3.2 are calculated by Monte Carlo simulation based on the fixed link travel time values in the second column.

Taking OD pair $1 \to 3$ as an example, there are 2 paths connecting this OD: path 1, link 3; path 2, link 1→4→6. Shown in Table 3.2, path flows simulated for these two paths are $f_1 = 13.47$ and $f_2 = 13.56$, respectively. Thus, the OD demand:

$$q_{1\to 3} = f_1 + f_2 = 27.03.$$

Based on the link travel time value shown in Table 3.2, the perceived path travel times of these two paths follow this form:

$$C_{1\to 3,1}(\mathbf{f}) \sim N(10.98,1) \text{ and } C_{1\to 3,2}(\mathbf{f}) \sim N(10.95,0.8).$$

Thus, the theoretical flow on path 1, denoted by $\overline{f}_1$, equals to:

$$\overline{f}_1 = q_{1\to 3} p_{1\to 3,1} = 27.03 \times \Pr\left(C_{1\to 3,1}(\mathbf{f}) \le C_{1\to 3,2}(\mathbf{f})\right) = 27.03 \times \Phi\left(\frac{10.95-10.98}{\sqrt{1+0.8}}\right) = 13.27$$

For traffic flow on path 1, difference between the theoretical solution and the solution obtained from Monte Carlo simulation is only 0.2 ($13.47-13.27=0.2$), which is less than maximal error $\varepsilon_2 = 0.5$. It also supports the validity of sample size $m_0 = 800$.

3.6.2 Example 2

The Sioux-Falls network has been widely used in transportation studies, which comprises 24 nodes, 76 links and 528 OD pairs. Example 2 is a modified Sioux-Fall network, using the following asymmetric link travel time functions (Bar-Gera, 2011):

$$t_a(\mathbf{v}) = t_a^0\left(1 + 0.15 \times \left(\frac{v_a + k_a \hat{v}_a}{(1+k_a)h_a}\right)^4\right), \tag{3.86}$$

where $\hat{a}$ is the opposite link of a on a two-way street, $\hat{v}_a$ is the link flow on link $\hat{a}$, and t_a^0 is the free-flow travel time of link a and h_a is the traffic flow capacity of link $a \in A$. The data for free-flow travel times and traffic flow capacities of Sioux-Falls network can be found in Bar-Gera (2011). It is not difficult to check that Jacobian matrix of all these link-travel-time functions is asymmetric.

Regarding sample sizes of the two-stage Monte Carlo simulation based method, the same size for the first stage is estimated by eqn. (3.50) with the maximum error $\varepsilon_1 = 0.5$. While for the second stage, the lower bound of sample size, m_0, is estimated by eqn. (3.85), and its tolerance value there is set to be $\varepsilon_2 = \min\{h_a, a \in A\} \times 5\%$, where h_a is flow capacity on link $a \in A$. Accordingly, it gives that $m_0 = 8025$ for this example.

The three algorithms are then utilized to solve the PA-SUEED problem on this example. PC parts of hybrid FPC-CA and hybrid CPC-CA algorithms terminate at 98 SNLs and 151 SNLs, respectively. Figure 3.6 gives the convergent trend of the three algorithms within 200 SNLs. It can be observed that all these three algorithms can converge, and meanwhile before they approach an accurate solution the two hybrid algorithms have much higher convergent speed than that of CA. It should be noted that CA parts of hybrid algorithms, for this example, can only slightly improve the accuracy level of final results, in view that the values of relative error obtained by terminated FPC and CPC are 0.33% and 0.66%, while average value of relative errors obtained by their CA parts in the next 100 loadings are 0.39% and 0.51%.

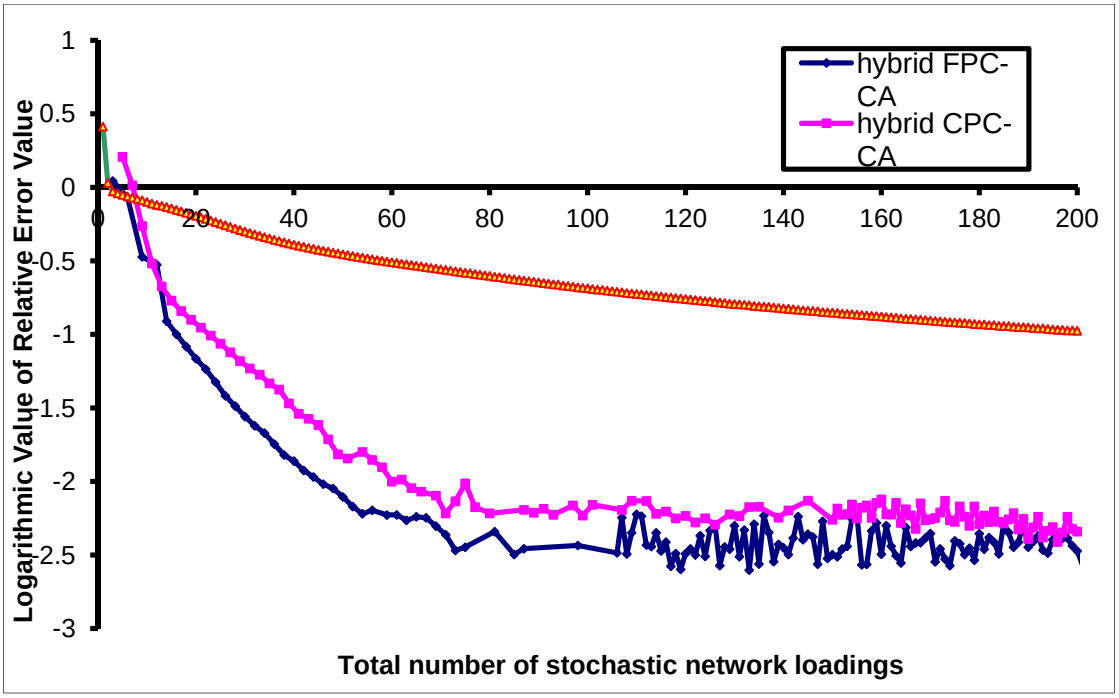

Figure 3.6 Convergent Trend of the Three Algorithms for Solving Example 2

Similar to Table 3.1, Table 3.5 gives the CPU time for each of the three algorithms in terms of different accuracy levels. According to this table, it can be seen that the hybrid CPC-CA and hybrid FPC-CA are also superior to the CA algorithm at any accuracy level for solving this example.

Table 3.5 CPU Time of Three Algorithms for Solving Example 2

Tolerance Value ε_f or ε_c	CA (Seconds)	Hybrid FPC-CA (Seconds)	Hybrid CPC-CA (Seconds)
20%	1235.47	164.23	172.64
15%	1666.02	164.23	195.19
10%	2479.89	209.19	262.83
5%	4711.08	276.61	397.98
1%	N/A	523.97	724.61

Analogous to the data pattern shown in Table 3.4, the maximum and average of the relative errors between link flows, link travel times and OD demands generated by the terminated PC parts and the SUE solution are tabulated in Table 3.6. It shows that when the PC parts are terminated, the resultant link flows, link travel times and OD demands are quite close to the SUE solution. For example 2, as shown in Figure 3.6, convergent rates of two PC parts are decelerated at around 70 SNLs, and the values of relative error obtained there are nearly the same with those obtained at the termination steps, thus similarly to example 1 the SUE solution outputted at those steps when convergent rate of PC are decelerated are already sufficiently accurate for use.

Table 3.6 Differences between the Results of Two Terminated PC Parts and the SUE Solution for Example 2

	Link cost		**Link flow**		**OD demand**	
	$\hat{E}^v$	$\overline{E}^v$	$\hat{E}^t$	$\overline{E}^t$	$\hat{E}^q$	$\overline{E}^q$
FPC	0.483%	0.153%	0.839%	0.216%	0. 453%	0.105%
CPC	0.925%	0.300%	1.71%	0.334%	0. 448%	0.112%

3.7 Conclusions

In this chapter, two VI models were developed for the probit-based asymmetric SUE problem with elastic demand. To solve these two VI models, two hybrid PC-CA algorithms were then put forward. Both the VI models and solution algorithms rely on a stochastic network loading (SNL) map. A link-based two-stage Monte Carlo simulation based method was proposed to calculate the SNL map, where its sample size for each stage was estimated by using statistical analysis techniques. The *two-stage* procedure

rather than an integrated *one-stage* procedure was utilized in this study, due to a better stability of the two-stage case.

The CA method adopts predetermined step sizes. While, PC parts of the two hybrid algorithms use the self-adaptive steps sizes, thus they can provide a superior convergent speed. Superiority of hybrid algorithms was numerical demonstrated by two numerical examples. The computational results for these examples indicated that PC parts of the two hybrid algorithms possess much faster convergent speed than that of conventional CA until they approach a solution. Moreover, CA parts of hybrid algorithms also outperform conventional CA because of better initial solutions.

CHAPTER 4 PA-SUEED WITH LINK CAPACITY CONSTRAINTS

This chapter addresses the PA-SUEED problem with link capacity constraints. Its mathematical definition, named as generalized SUE conditions, is first proposed. A variational inequality (VI) model is then developed whose solutions can fulfill the generalized SUE conditions. Solution existence, Lipschitz-continuous and monotone properties of this VI model are rigorously proven, which ensure the convergence of a projection-based prediction-correction (PC) algorithm to solve this model.

4.1 Background

As per Section 2.2.3, link capacity constraints are un-neglectable for the traffic assignment problem, since the link flow in practice would never exceed its physical capacity. Moreover, in some particular road segments, a threshold is usually added to the link flow by the network authorities to mitigate traffic congestions or to ameliorate its negative impacts on the environment (e.g., Yang and Bell, 1997; Yang and Huang, 2005).

For decades, traffic assignment problem with link capacity constraints has been a sophisticated and complicated research topic, see Section 2.2.3 for a review of relevant studies. For the SUE problem with link capacity constraints, its formulation and computation becomes even more difficult and complicated, and till now only two studies can be observed, which are Bell (1995a) for logit-based SUE with link capacity constraints and Meng et al. (2008) in the case of General SUE. As mentioned at Section 2.2.3, these two existing studies rely on the assumption of separate link travel time functions and fixed demand. Thereby, due to its complicated properties, the PA-SUEED

problem with link capacity constraints is still an open question, despite its superior fitness to the practical conditions.

To solve the PA-SUEED with link capacity constraints, two questions are subsequently brought up: (a) is there a solution for the generalized SUE conditions? and (b) how to find a solution if it does exist. To answer these two questions, this chapter will develop a VI model defined on a non-empty, compact and convex set, and show that any solution of the VI model fulfils the generalized SUE conditions. It is rigorously demonstrated that the proposed VI model is monotone and Lipschitz continuous. These properties of the proposed VI model imply that it has at least one solution, which answers the question (a). More importantly, monotonicity and Lipschitz-continuity of the proposed VI model guarantee the convergence of self-adaptive prediction-correction (PC) algorithm proposed by He and Liao (2002) as a solution algorithm.

Section 4.2 of this chapter gives mathematical definitions for the link flow solution of PA-SUEED with link capacity constraints. Section 4.3 develops a VI model and rigorously proves its monotone and continuous properties. Section 4.4 presents the convergent self-adaptive PC algorithm. Section 4.5 uses a numerical example to validate the proposed methodology. Conclusions are then provided in Section 4.6.

4.2 Generalized SUE Conditions

Based on the strongly connected network $G=(N,A)$ defined at Section 3.2, it is further assumed that each link a in set $\bar{A}$ has the link capacity constraint:

$$v_a \le H_a, a \in \bar{A}, \tag{4.1}$$

where H_a is a given positive threshold/capacity on link $a \in \bar{A}$, and $\bar{A}$ is a subset of A. Note that the capacity constrained links are allowed to be *partial of or all of* the links on the network. Other than using physical capacity of traffic flow on the link, this threshold is sometimes determined to be a relatively smaller value by network authorities so as to mitigate traffic congestion and/or vehicle emission (Yang and Huang, 2005). In this chapter, the link travel time function vector $\mathbf{t}(\mathbf{v})$ is assumed to have a positive definite Jacobian matrix $\nabla_{\mathbf{v}}\mathbf{t}(\mathbf{v})$, which is a sufficient condition of the strict monotonicity (Nagurney, 1993). This assumption of positive definite Jacobian matrix is usually made in previous studies for asymmetric traffic assignment problems (e.g., Dafermos, 1980).

Recall that in Chapter 3, the OD demand is assumed to be a continuously differentiable, non-increasing and bounded function with respect to the satisfaction $S_w(\mathbf{c}_w(\mathbf{v}))$, denoted by $D_w(\bullet)$, namely:

$$q_w = D_w\left(S_w\left(\mathbf{c}_w(\mathbf{v})\right)\right) \le \bar{q}_w, w \in W \tag{4.2}$$

This function was assumed to be generic to the fixed and elastic demand cases in Chapter 3. However, when dealing with the link capacity constrained traffic assignment problem, in the case of fixed demand, the solution may not exist; for instance, considering there is only one link connecting an OD pair, and the OD demand is much larger than the capacity constraint. Clearly, it is impossible to have any solution in such case. Therefore, to cope with the PA-SUEED with link capacity constraints in this chapter, it is reasonably

assumed that travel demand between an OD pair will vanish when the travel time between this OD pair approaches infinity, i.e.,

$$\lim_{x \to +\infty} D_w(x) = 0, w \in W \tag{4.3}$$

Recall that the following equation is proposed in Section 3.2 as a mathematical definition for the link-flow solution of PA-SUEED:

$$v_a = \sum_{w \in W} D_w\left(S_w\left(\left(\Delta_w\right)^{\mathrm{T}} \mathbf{t}(\mathbf{v})\right)\right) P_{wa}\left(\mathbf{t}(\mathbf{v})\right), a \in A, \tag{4.4}$$

and in this section a mathematical definition is also addressed in the first place for the PA-SUEED with link capacity constraints, which is named as generalized SUE conditions.

For the DUE problem with link capacity constraints, it has been widely recognized that it equals to a conventional DUE problem in terms of the *generalized link travel time* (e.g., Larsson and Patriksson, 1995). This generalized link travel time amounts to the summation of actual link travel time and the optimal Lagrangian multiplier of its link capacity constrain. To be consistent with the generalized DUE conditions, the generalized SUE conditions are defined as: a link flow pattern $\mathbf{v} = \left(v_a, a \in A\right)^{\mathrm{T}}$ is the solution for PA-SUEED with link capacity constraints if and only if there exists a vector of Lagrangian multipliers corresponding to link capacity constraints, denoted by $\mathbf{u}^* = \left(u_a^*, a \in \bar{A}\right)^{\mathrm{T}}$, such that

$$v_a = \sum_{w \in W}\left[D_w\left(S_w\left(\left(\Delta_w\right)^{\mathrm{T}} \mathbf{t}(\mathbf{v}) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)\right) \times P_{wa}\left(\left(\Delta_w\right)^{\mathrm{T}} \mathbf{t}(\mathbf{v}) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)\right], a \in A \tag{4.5}$$

$$v_a \le H_a, a \in \bar{A} \tag{4.6}$$

$$u_a^* \times \left(v_a - H_a \right) = 0, a \in \bar{A} \tag{4.7}$$

$$u_a^* \geq 0, a \in \bar{A} \tag{4.8}$$

where vector $\boldsymbol{\lambda}_w \left(\mathbf{u}^* \right)$ is defined by

$$\boldsymbol{\lambda}_w \left(\mathbf{u}^* \right) = \left(\lambda_{wk} \left(\mathbf{u}^* \right) = \sum_{a \in \bar{A}} u_a^* \delta_{ak}^w, k \in R_w \right)^{\mathrm{T}}, w \in W . \tag{4.9}$$

For the sake of presentation, the Lagrangian multipliers satisfying eqns. (4.5)-(4.8) are called as optimal Lagrangian multipliers. Note that eqns. (4.6)-(4.8) are the complementary slackness conditions between the SUE link flow solution and the optimal Lagrangian multipliers. If there is no link capacity constraint, it implies that $u_a^* = 0, a \in \bar{A}$. It should be noted that the flow conservation constraints should also be satisfied here.

When the optimal Lagrangian multipliers are attained, the addressed capacity constrained problem is equivalent to a standard PA-SUEED problem in terms of the generalized link travel time functions:

$$\hat{t}_a \left(\mathbf{v}, \mathbf{u}^* \right) = \begin{cases} t_a \left(\mathbf{v} \right) + u_a^*, a \in \bar{A} \\ t_a \left(\mathbf{v} \right), a \in A \setminus \bar{A} \end{cases}, a \in A, \tag{4.10}$$

which can be simply solved by the algorithms discussed in Chapter 3. To solve the PA-SUEED problem with link capacity constraints thus becomes searching for the optimal Lagrangian multipliers. As aforementioned, the following two questions should be addressed:

(a) Does the optimal Lagrangian multiplier solution exist?

(b) How to find such a solution if it really exists.

To answer these two questions, Section 4.3 first gives a suitable VI model for the generalized SUE conditions and Section 4.4 then introduces a global convergent solution algorithm for solving this VI model.

Remark 4.1: Formulation for the DUE problem with link capacity constraints can be easily handled by directly taking the link capacity constraints as side constraints into a model for DUE problem without link capacity constraint (e.g. Patriksson, 1994a; Larsson and Patriksson, 1999). However, for SUE problems with link capacity constraints, the challenges in formulation are considerably enlarged. For instance, directly adding link capacity constraints into the models for PA-SUEED problem is not effective. Even for the SUE problem with separable link travel time functions, adding link capacity constraints to the optimization model proposed by Daganzo (1982) or Maher and Zhang (2000) does not give us a suitable formulation.

4.3 Mathematical Model

The complementary slackness conditions, eqns. (4.6)-(4.8), can be regarded as a Nonlinear Complimentarity Problem (NCP) of the vector of Lagrangian multipliers $\mathbf{u} = \left(u_a, a \in \bar{A}\right)^{\mathrm{T}}$. It is well known that when the feasible set of $\mathbf{u} = \left(u_a, a \in \bar{A}\right)^{\mathrm{T}}$ is the whole non-negative orthant, this NCP model is equivalent to the following VI model, denoted by $VI\left(\mathbf{\Phi}, \Re_+^{|\bar{A}|}\right)$:

$$\mathbf{\Phi}\left(\mathbf{u}^*\right)^{\mathbf{T}}\left(\mathbf{u}-\mathbf{u}^*\right) \geq 0, \forall \mathbf{u} \in \Re_+^{|\bar{A}|}, \tag{4.11}$$

where $\Re_+^{|\bar{A}|} = \left\{\mathbf{u} \middle| u_a \geq 0, a \in \bar{A}\right\}$ denotes the feasible set and $\mathbf{\Phi}(\mathbf{u})$ is a $|\bar{A}|$-dimensional vector function defined below.

$$\mathbf{\Phi}(\mathbf{u}) = \mathbf{H} - \mathbf{v}_{\bar{A}}(\mathbf{u}) = \left(H_a - v_a(\mathbf{u}), a \in \bar{A}\right)^{\mathbf{T}} : \Re_+^{|\bar{A}|} \to \Re^{|\bar{A}|}, \tag{4.12}$$

where $\mathbf{H} = \left(H_a, a \in \bar{A}\right)^{\mathbf{T}}$ is the vector for all the link capacity constraints, and $\mathbf{v}_{\bar{A}}(\mathbf{u})$ is the sub-vector of vector $\mathbf{v}(\mathbf{u}) = \left(\mathbf{v}_{\bar{A}}(\mathbf{u}), \mathbf{v}_{A\setminus\bar{A}}(\mathbf{u})\right)^{\mathrm{T}}$, where $\mathbf{v}(\mathbf{u})$ is a link flow solution for the PA-SUEED problem with the following generalized link travel time functions (no link capacity constraint):

$$\hat{t}_a(\mathbf{v}, \mathbf{u}) = \begin{cases} t_a(\mathbf{v}) + u_a, a \in \bar{A} \\ t_a(\mathbf{v}), a \in A \setminus \bar{A} \end{cases}, \quad a \in A. \tag{4.13}$$

In other words, vector $\mathbf{v}(\mathbf{u})$ is a solution of the fixed-point model with Lagrangian multiplier vector $\mathbf{u}$:

$$v_a(\mathbf{u}) = \sum_{w \in W} \left[D_w\left(S_w\left(\Delta_w^{\mathrm{T}} \hat{\mathbf{t}}(\mathbf{v}, \mathbf{u})\right)\right) \times P_{wa}\left(\Delta_w^{\mathrm{T}} \hat{\mathbf{t}}(\mathbf{v}, \mathbf{u})\right)\right], a \in A, \tag{4.14}$$

where $\hat{\mathbf{t}}(\mathbf{v}, \mathbf{u}) = \left(\hat{t}_a(\mathbf{v}, \mathbf{u}), a \in A\right)^{\mathrm{T}}$ denotes a vector of all the generalized link travel time functions. $\mathbf{v}(\mathbf{u})$ is referred to as parametric SUE link flow vector. In reality, sub-vector $\mathbf{v}_{\bar{A}}(\mathbf{u})$ is a collection of the parametric SUE link flows on those links with link capacity constraints.

Parametric SUE link flow vector $\mathbf{v}(\mathbf{u})$ is an implicit mapping of Lagrangian multiplier vector $\mathbf{u}$. It can be easily shown that the generalized link travel time function vector $\hat{\mathbf{t}}(\mathbf{v}, \mathbf{u})$ is strictly monotone with respect to link flow vector $\mathbf{v}$ for any given non-

negative Lagrangian multiplier vector $\mathbf{u}$. $\mathbf{v}(\mathbf{u})$ is therefore unique for any given $\mathbf{u}$ (Cantarella, 1997). Thereby, $\mathbf{v}(\mathbf{u})$ can be solved by the solution algorithms discussed in Chapter 3.

4.3.1 Monotone and Continuous Properties of the Vector Function

The existence of solution to VI model $VI\left(\mathbf{\Phi}, \Re_+^{|\bar{A}|}\right)$ as well as the global convergence of its solution algorithm depends on some fundamental properties of the vector function $\mathbf{\Phi}(\mathbf{u})$. These properties mainly include monotonicity and continuity of vector function $\mathbf{\Phi}(\mathbf{u})$ (e.g. Patriksson, 1994b). Thus, the following three important properties of vector function $\mathbf{\Phi}(\mathbf{u})$ are highlighted in Propositions 4.1 to 4.3 with rigorous proofs.

***Proposition 4.1*:** *Vector function* $\mathbf{\Phi}(\mathbf{u})$ *is monotone on* $\Re_+^{|\bar{A}|}$*, namely,*

$$\left(\mathbf{\Phi}(\mathbf{u}') - \mathbf{\Phi}(\mathbf{u}'')\right)^{\mathrm{T}} (\mathbf{u}' - \mathbf{u}'') \geq 0, \forall \mathbf{u}', \mathbf{u}'' \in \Re_+^{|\bar{A}|}. \tag{4.15}$$

Proof.

For any two distinct non-negative Lagrangian multiplier vectors $\mathbf{u}'$ and $\mathbf{u}''$, let $\mathbf{v}(\mathbf{u}')$ and $\mathbf{v}(\mathbf{u}'')$ denote the corresponding SUE link flow solutions. Hence, there should be two SUE path solutions $\mathbf{f}(\mathbf{u}') = \left(\mathbf{f}_w(\mathbf{u}'), w \in W\right)^{\mathrm{T}}$ and $\mathbf{f}(\mathbf{u}'') = \left(\mathbf{f}_w(\mathbf{u}''), w \in W\right)^{\mathrm{T}}$ such that

$$\mathbf{v}(\mathbf{u}') = \Delta \mathbf{f}(\mathbf{u}'), \tag{4.16}$$

$$\mathbf{v}(\mathbf{u}'') = \Delta \mathbf{f}(\mathbf{u}''), \tag{4.17}$$

$$\mathbf{f}_w(\mathbf{u}') = D_w\left(S_w(\hat{\mathbf{c}}'_w)\right) \times \mathbf{p}_w(\hat{\mathbf{c}}'_w), w \in W, \tag{4.18}$$

$$\mathbf{f}_w(\mathbf{u}'') = D_w\left(S_w(\hat{\mathbf{c}}''_w)\right) \times \mathbf{p}_w(\hat{\mathbf{c}}''_w), w \in W, \tag{4.19}$$

where four vectors: $\hat{\mathbf{c}}'_w = \Delta_w^{\mathrm{T}} \hat{\mathbf{t}}\left(\mathbf{v}(\mathbf{u}'), \mathbf{u}'\right)$, $\hat{\mathbf{c}}''_w = \Delta_w^{\mathrm{T}} \hat{\mathbf{t}}\left(\mathbf{v}(\mathbf{u}''), \mathbf{u}''\right)$, $\mathbf{p}_w(\hat{\mathbf{c}}') = \left(p_{wk}(\hat{\mathbf{c}}'_w), k \in R_w\right)^{\mathrm{T}}$ and $\mathbf{p}_w(\hat{\mathbf{c}}'') = \left(p_{wk}(\hat{\mathbf{c}}''_w), k \in R_w\right)^{\mathrm{T}}$.

Since satisfaction function $S_w(\hat{\mathbf{c}}_w)$ is concave (Sheffi, 1985; Cantarella, 1997), it follows that

$$S_w(\hat{\mathbf{c}}'_w) \le S_w(\hat{\mathbf{c}}''_w) + \left(\mathbf{p}_w(\hat{\mathbf{c}}''_w)\right)^{\mathrm{T}} (\hat{\mathbf{c}}'_w - \hat{\mathbf{c}}''_w), w \in W, \tag{4.20}$$

$$S_w(\hat{\mathbf{c}}''_w) \le S_w(\hat{\mathbf{c}}'_w) + \left(\mathbf{p}_w(\hat{\mathbf{c}}'_w)\right)^{\mathrm{T}} (\hat{\mathbf{c}}''_w - \hat{\mathbf{c}}'_w), w \in W. \tag{4.21}$$

After multiplying both sides of eqns. (4.20) and (4.21) by $D_w\left(S_w(\hat{\mathbf{c}}''_w)\right)$ and $D_w\left(S_w(\hat{\mathbf{c}}'_w)\right)$, respectively, it gives

$$D_w\left(S_w(\hat{\mathbf{c}}''_w)\right) \times \left(S_w(\hat{\mathbf{c}}'_w) - S_w(\hat{\mathbf{c}}''_w)\right) \le \left(\mathbf{f}_w(\mathbf{u}'')\right)^{\mathrm{T}} (\hat{\mathbf{c}}'_w - \hat{\mathbf{c}}''_w), w \in W, \tag{4.22}$$

$$D_w\left(S_w(\hat{\mathbf{c}}'_w)\right) \times \left(S_w(\hat{\mathbf{c}}''_w) - S_w(\hat{\mathbf{c}}'_w)\right) \le \left(\mathbf{f}_w(\mathbf{u}')\right)^{\mathrm{T}} (\hat{\mathbf{c}}''_w - \hat{\mathbf{c}}'_w), w \in W. \tag{4.23}$$

Adding up eqn. (4.22) and eqn. (4.23) yields that

$$\begin{aligned} &\left[D_w\left(S_w(\hat{\mathbf{c}}''_w)\right) - D_w\left(S_w(\hat{\mathbf{c}}'_w)\right)\right] \times \left[S_w(\hat{\mathbf{c}}'_w) - S_w(\hat{\mathbf{c}}''_w)\right] \\ &\le \left[\mathbf{f}_w(\mathbf{u}'') - \mathbf{f}_w(\mathbf{u}')\right]^{\mathrm{T}} (\hat{\mathbf{c}}'_w - \hat{\mathbf{c}}''_w), w \in W. \end{aligned} \tag{4.24}$$

namely,

$$\begin{aligned} &\left[D_w\left(S_w(\hat{\mathbf{c}}'_w)\right) - D_w\left(S_w(\hat{\mathbf{c}}''_w)\right)\right] \times \left[S_w(\hat{\mathbf{c}}'_w) - S_w(\hat{\mathbf{c}}''_w)\right] \\ &\ge \left[\mathbf{f}_w(\mathbf{u}'') - \mathbf{f}_w(\mathbf{u}')\right]^{\mathrm{T}} (\hat{\mathbf{c}}'_w - \hat{\mathbf{c}}''_w), w \in W. \end{aligned} \tag{4.25}$$

The monotonicity of OD demand functions implies that

$$\left[D_w\left(S_w\left(\hat{\mathbf{c}}'_w\right)\right)-D_w\left(S_w\left(\hat{\mathbf{c}}''_w\right)\right)\right]\times\left[S_w\left(\hat{\mathbf{c}}'_w\right)-S_w\left(\hat{\mathbf{c}}''_w\right)\right]\le 0, w\in W\,. \tag{4.26}$$

According to eqns. (4.25) and (4.26), it implies that:

$$\left[\mathbf{f}_w\left(\mathbf{u}'\right)-\mathbf{f}_w\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left(\hat{\mathbf{c}}'_w-\hat{\mathbf{c}}''_w\right)\le 0, w\in W\,. \tag{4.27}$$

Because $\hat{\mathbf{c}}'_w=\Delta_w^{\mathrm{T}}\hat{\mathbf{t}}\left(\mathbf{v}\left(\mathbf{u}'\right),\mathbf{u}'\right)$ and $\hat{\mathbf{c}}''_w=\Delta_w^{\mathrm{T}}\hat{\mathbf{t}}\left(\mathbf{v}\left(\mathbf{u}''\right),\mathbf{u}''\right)$, eqn. (4.27) can be rewritten by

$$\left(\mathbf{f}_w\left(\mathbf{u}'\right)-\mathbf{f}_w\left(\mathbf{u}''\right)\right)^{\mathrm{T}}\left[\Delta_w^{\mathrm{T}}\hat{\mathbf{t}}\left(\mathbf{v}\left(\mathbf{u}'\right),\mathbf{u}'\right)-\Delta_w^{\mathrm{T}}\hat{\mathbf{t}}\left(\mathbf{v}\left(\mathbf{u}''\right),\mathbf{u}''\right)\right]\le 0, w\in W\,. \tag{4.28}$$

After rearranging the left hand side of eqn. (4.28), it follows that

$$\begin{aligned}&\left[\mathbf{v}_w\left(\mathbf{u}'\right)-\mathbf{v}_w\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left[\mathbf{t}\left(\mathbf{v}\left(\mathbf{u}'\right)\right)-\mathbf{t}\left(\mathbf{v}\left(\mathbf{u}''\right)\right)\right]+\left[\mathbf{v}_{\bar{A},w}\left(\mathbf{u}'\right)-\mathbf{v}_{\bar{A},w}\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left(\mathbf{u}'-\mathbf{u}''\right)\\&\quad\le 0, w\in W\,.\end{aligned} \tag{4.29}$$

where the four link flow vectors associated with OD pair w: $\mathbf{v}_w\left(\mathbf{u}'\right)=\left(v_{wa}\left(\mathbf{u}'\right),a\in A\right)^T$,

$\mathbf{v}_w\left(\mathbf{u}''\right)=\left(v_{wa}\left(\mathbf{u}''\right),a\in A\right)^T$, $\mathbf{v}_{w,\bar{A}}\left(\mathbf{u}'\right)=\left(v_{wa}\left(\mathbf{u}'\right),a\in \bar{A}\right)^T$, $\mathbf{v}_{w,\bar{A}}\left(\mathbf{u}''\right)=\left(v_{wa}\left(\mathbf{u}''\right),a\in \bar{A}\right)^T$

with the elements:

$$\mathbf{v}_{wa}\left(\mathbf{u}'\right)=\sum_{k\in R_w} f_{wk}\left(\mathbf{u}'\right)\delta_{ak}^w, a\in A, \tag{4.30}$$

$$\mathbf{v}_{wa}\left(\mathbf{u}''\right)=\sum_{k\in R_w} f_{wk}\left(\mathbf{u}''\right)\delta_{ak}^w, a\in A\,. \tag{4.31}$$

According to eqn. (4.29), it gives that

$$\begin{aligned}&\sum_{w\in W}\left\{\left[\mathbf{v}_w\left(\mathbf{u}'\right)-\mathbf{v}_w\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left[\mathbf{t}\left(\mathbf{v}\left(\mathbf{u}'\right)\right)-\mathbf{t}\left(\mathbf{u}''\right)\right]+\left[\mathbf{v}_{w,\bar{A}}\left(\mathbf{u}'\right)-\mathbf{v}_{w,\bar{A}}\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left(\mathbf{u}'-\mathbf{u}''\right)\right\}\\&\quad\le 0.\end{aligned} \tag{4.32}$$

namely,

$$\left[\mathbf{v}_{\bar{A}}\left(\mathbf{u}'\right)-\mathbf{v}_{\bar{A}}\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left(\mathbf{u}'-\mathbf{u}''\right)\le-\left[\mathbf{v}\left(\mathbf{u}'\right)-\mathbf{v}\left(\mathbf{u}''\right)\right]^{\mathrm{T}}\left[\mathbf{t}\left(\mathbf{v}\left(\mathbf{u}'\right)\right)-\mathbf{t}\left(\mathbf{v}\left(\mathbf{u}''\right)\right)\right]. \tag{4.33}$$

The right hand side of eqn. (4.33) is non-negative since link travel time function vector $\mathbf{t}(\mathbf{v})$ is strictly monotone. Thus, if $\mathbf{v}_{\bar{A}}(\mathbf{u}') \neq \mathbf{v}_{\bar{A}}(\mathbf{u}'')$, it gives that

$$\left[\mathbf{v}_{\bar{A}}(\mathbf{u}') - \mathbf{v}_{\bar{A}}(\mathbf{u}'')\right]^{\mathrm{T}} (\mathbf{u}' - \mathbf{u}'') < 0. \tag{4.34}$$

Otherwise, if $\mathbf{v}_{\bar{A}}(\mathbf{u}') = \mathbf{v}_{\bar{A}}(\mathbf{u}'')$, it follows that

$$\left[\mathbf{v}_{\bar{A}}(\mathbf{u}') - \mathbf{v}_{\bar{A}}(\mathbf{u}'')\right]^{\mathrm{T}} (\mathbf{u}' - \mathbf{u}'') = 0. \tag{4.35}$$

According to eqns. (4.34) and (4.35), it is straightforward to see that

$$\begin{aligned} \left(\boldsymbol{\Phi}(\mathbf{u}') - \boldsymbol{\Phi}(\mathbf{u}'')\right)^{\mathrm{T}} (\mathbf{u}' - \mathbf{u}'') &= \left(H - \mathbf{v}_{\bar{A}}(\mathbf{u}') - H + \mathbf{v}_{\bar{A}}(\mathbf{u}'')\right)^{\mathrm{T}} (\mathbf{u}' - \mathbf{u}'') \\ &= -\left(\mathbf{v}_{\bar{A}}(\mathbf{u}') - \mathbf{v}_{\bar{A}}(\mathbf{u}'')\right)^{\mathrm{T}} (\mathbf{u}' - \mathbf{u}'') \geq 0. \end{aligned} \tag{4.36}$$

In other words, vector function $\boldsymbol{\Phi}(\mathbf{u})$ is monotone on $\Re_+^{|\bar{A}|}$. □

***Proposition 4.2*:** *Vector function $\boldsymbol{\Phi}(\mathbf{u})$ is continuously differentiable on $\Re_+^{|\bar{A}|}$.*

Proof.

The *implicit function theorem* is used to prove this proposition. A vector function $\mathbf{g}(\mathbf{v},\mathbf{u})$ on $\Theta \times \Re_+^{|A|}$ is first defined as follows:

$$\mathbf{g}(\mathbf{v},\mathbf{u}) = \left(g_a(\mathbf{v},\mathbf{u}), a \in A\right)^{\mathrm{T}}, \tag{4.37}$$

where

$$g_a(\mathbf{v},\mathbf{u}) = v_a - \sum_{w \in W} \sum_{k \in R_w} \left[D_w\left(S_w\left(\Delta_w^{\mathrm{T}} \hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right)\right) \times p_{wk}\left(\Delta_w^{\mathrm{T}} \hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right) \delta_{ak}^w \right], a \in A. \tag{4.38}$$

Let $\nabla_{\mathbf{v}}\mathbf{g}(\mathbf{v},\mathbf{u})$ and $\nabla_{\mathbf{u}}\mathbf{g}(\mathbf{v},\mathbf{u})$ be Jacobian matrices of vector function $\mathbf{g}(\mathbf{v},\mathbf{u})$ with respect to vector $\mathbf{v}$ and $\mathbf{u}$, respectively. According to eqns. (4.37)-(4.38), these two Jacobian matrices have the explicit expressions:

$$\begin{aligned}\nabla_{\mathbf{v}}\mathbf{g}(\mathbf{v},\mathbf{u}) = \mathbf{I} - \sum_{w\in W}\left\{\frac{\partial D_w(\mathbf{v},\mathbf{u})}{\partial S_w}\times\left[(\Delta_w\mathbf{p}_w)\cdot(\Delta_w\mathbf{p}_w)^{\mathrm{T}}\right]\times\nabla_{\mathbf{v}}\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right\} \\ -\sum_{w\in W}\left\{D_w(\mathbf{v},\mathbf{u})\times\left[\Delta_w\cdot\nabla_{\hat{c}}\mathbf{p}_w\cdot(\Delta_w)^{\mathrm{T}}\right]\times\nabla_{\mathbf{v}}\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right\},\end{aligned} \tag{4.39}$$

$$\begin{aligned}\nabla_{\mathbf{u}}\mathbf{g}(\mathbf{v},\mathbf{u}) = -\sum_{w\in W}\left\{\frac{\partial D_w(\mathbf{v},\mathbf{u})}{\partial S_w}\times\left[(\Delta_w\mathbf{p}_w)\times(\Delta_w\mathbf{p}_w)^{\mathrm{T}}\right]\times\nabla_{\mathbf{u}}\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right\} \\ -\sum_{w\in W}\left[D_w(\mathbf{v},\mathbf{u})\times\left[\Delta_w\cdot\nabla_{\hat{c}}\mathbf{p}_w\cdot(\Delta_w)^{\mathrm{T}}\right]\times\nabla_{\mathbf{u}}\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right],\end{aligned} \tag{4.40}$$

where $\mathbf{p}_w$, S_w, and $D_w(\mathbf{v},\mathbf{u})$ are abbreviations of $\mathbf{p}_w = \left(p_{wk}\left(\Delta_w^T\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right), k\in R_w\right)^{\mathrm{T}}$, $S_w\left(\Delta_w^T\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right)$ and $D_w\left(S_w\left(\Delta_w^T\hat{\mathbf{t}}(\mathbf{v},\mathbf{u})\right)\right)$, respectively. It should be noted that the derivation of eqns. (4.39) and (4.40) uses the following property of satisfaction:

$$\frac{\partial S_w\left(\hat{\mathbf{c}}_w(\mathbf{f})\right)}{\partial\left(\hat{c}_{wk}(\mathbf{f})\right)} = p_{wk}\left(\hat{\mathbf{c}}_w(\mathbf{f})\right), k\in R_w, w\in W. \tag{4.41}$$

For any given $\mathbf{u}_0\in\Re_+^{|\bar{A}|}$, an unique $\mathbf{v}_0 = \mathbf{v}(\mathbf{u}_0)$ can be obtained. Then, I proceed to prove that $\nabla_{\mathbf{v}}\mathbf{g}(\mathbf{v}_0,\mathbf{u}_0)$ is non-singular. It is readily to verify that

$$\nabla_{\mathbf{v}}\hat{\mathbf{t}}(\mathbf{v}_0,\mathbf{u}_0) = \nabla_{\mathbf{v}}\mathbf{t}(\mathbf{v}_0). \tag{4.42}$$

As Jacobian matrix $\nabla_{\mathbf{v}}\mathbf{t}(\mathbf{v}_0)$ is positive definite, $\nabla_{\mathbf{v}}\mathbf{t}(\mathbf{v}_0)$ is thereby non-singular and its inverse $\left[\nabla_{\mathbf{v}}\mathbf{t}(\mathbf{v}_0)\right]^{-1}$ is also positive definite. According to eqn. (4.39), Jacobian matrix $\nabla_{\mathbf{v}}\mathbf{g}(\mathbf{v},\mathbf{u})$ at $(\mathbf{v}_0,\mathbf{u}_0)$ can be rewritten as follows:

$$\nabla_{\mathbf{v}}\mathbf{g}\left(\mathbf{v}_0,\mathbf{u}_0\right)=\left\{\left[\nabla_{\mathbf{v}}\mathbf{t}\left(\mathbf{v}_0\right)\right]^{-1}-\sum_{w\in W}\left\{\frac{\partial D_w\left(\mathbf{v}_0,\mathbf{u}_0\right)}{\partial S_w}\times\left[\left(\Delta_w\mathbf{p}_w\right)\cdot\left(\Delta_w\mathbf{p}_w\right)^{\mathrm{T}}\right]\right\}\right.$$
$$\left.-\sum_{w\in W}\left\{D_w\left(\mathbf{v}_0,\mathbf{u}_0\right)\times\left[\Delta_w\cdot\nabla_{\hat{c}}\mathbf{p}_w\cdot\left(\Delta_w\right)^{\mathrm{T}}\right]\right\}\right\}\times\nabla_{\mathbf{v}}\mathbf{t}\left(\mathbf{v}_0\right). \tag{4.43}$$

There are three terms in the brace of the right-hand-side of eqn. (4.43). The first term $\left[\nabla_{\mathbf{v}}\mathbf{t}\left(\mathbf{v}_0\right)\right]^{-1}$ is positive definite. The second term is positive semi-definite because $\frac{\partial D_w\left(\mathbf{v}_0,\mathbf{u}_0\right)}{\partial S_w}\le 0$ and $\left[\left(\Delta_w\mathbf{p}_w\right)\cdot\left(\Delta_w\mathbf{p}_w\right)^{\mathrm{T}}\right]$ is positive semi-definite. Since $\left[\Delta_w\cdot\nabla_{\hat{c}}\mathbf{p}_w\cdot\left(\Delta_w\right)^{\mathrm{T}}\right]$ is a negative semi-definite matrix (Page 320, Sheffi, 1985) and $D_w\left(\mathbf{v}_0,\mathbf{u}_0\right)\ge 0$, the third term is thus also positive semi-definite. Hence, the whole part in the brace of the right hand side of eqn. (4.43) is a positive definite matrix. In other words, $\nabla_{\mathbf{v}}\mathbf{g}\left(\mathbf{v}_0,\mathbf{u}_0\right)$ equals to a positive definite matrix multiplied by a non-singular matrix. Therefore, $\nabla_{\mathbf{v}}\mathbf{g}\left(\mathbf{v}_0,\mathbf{u}_0\right)$ is non-singular.

According to eqns. (4.39)-(4.40), it can be seen that $\nabla_{\mathbf{v}}\mathbf{g}\left(\mathbf{v},\mathbf{u}\right)$ and $\nabla_{\mathbf{u}}\mathbf{g}\left(\mathbf{v},\mathbf{u}\right)$ both are continuous with respect to $\left(\mathbf{v},\mathbf{u}\right)$. As $\mathbf{v}\left(\mathbf{u}_0\right)$ is the parametric SUE link flow vector, i.e., a solution to the fixed-point model (4.14), it follows that

$$\mathbf{g}\left(\mathbf{v}_0,\mathbf{u}_0\right)=\mathbf{0}. \tag{4.44}$$

Since $\nabla_{\mathbf{v}}\mathbf{g}\left(\mathbf{v}_0,\mathbf{u}_0\right)$ is a non-singular matrix, $\mathbf{v}\left(\mathbf{u}\right)$ is continuously differentiable in a neighborhood of $\mathbf{u}_0$ according to the implicit function theorem (Theorem 5.2.4, Ortega and Rheinboldt, 1970) and it has the gradient:

$$\nabla_{\mathbf{u}}\mathbf{v}(\mathbf{u}) = -\left[\nabla_{\mathbf{v}}\mathbf{g}(\mathbf{v},\mathbf{u})\right]^{-1}\nabla_{\mathbf{u}}\mathbf{g}(\mathbf{v},\mathbf{u}). \tag{4.45}$$

As $\mathbf{v}(\mathbf{u})$ is continuously differentiable on $\Re_+^{|\bar{A}|}$, $\mathbf{\Phi}(\mathbf{u})$ is thus continuously differentiable on $\Re_+^{|\bar{A}|}$ according to its definition shown in eqn. (4.12). □

Since vector function $\mathbf{v}(\mathbf{u})$ is continuously differentiable on $\Re_+^{|\bar{A}|}$, its Jacobian matrix $\nabla_{\mathbf{u}}\mathbf{\Phi}(\mathbf{u})$ is thus continuous on $\Re_+^{|\bar{A}|}$. The 2-norm of $\nabla_{\mathbf{u}}\mathbf{v}(\mathbf{u})$ is therefore bounded from above over *any* non-empty and compact set in $\Re_+^{|\bar{A}|}$ (such a set is denoted by Ψ), namely, there is a positive constant L such that

$$\left\|\nabla_{\mathbf{u}}\mathbf{v}(\mathbf{u})\right\|_2 \le L, \forall \mathbf{u} \in \Psi. \tag{4.46}$$

According to the mean-value theorem (Theorem 3.2.4 of Ortega and Rheinboldt, 1970), it can be seen that

$$\left\|\mathbf{\Phi}(\mathbf{u}') - \mathbf{\Phi}(\mathbf{u}'')\right\|_2 = \left\|\mathbf{v}(\mathbf{u}') - \mathbf{v}(\mathbf{u}'')\right\|_2 \le L\left\|\mathbf{u}' - \mathbf{u}''\right\|, \forall \mathbf{u}', \mathbf{u}'' \in \Psi. \tag{4.47}$$

In other words, the following proposition can be obtained.

***Proposition 4.3*:** *Vector function $\mathbf{\Phi}(\mathbf{u})$ is uniform Lipschitz-continuous on any non-empty and compact set in $\Re_+^{|\bar{A}|}$.*

4.3.2 A Restricted Variational Inequality Model

Though three importance properties of vector function $\mathbf{\Phi}(\mathbf{u})$ are proven, it is still quite difficult to show the existence of a solution to VI model $VI\left(\mathbf{\Phi}, \Re_+^{|\bar{A}|}\right)$ by means of some existing sufficient conditions, e.g., the coercivity condition (Facchinei and Pang, 2003). It

is well known that a VI model has at least one solution if it is continuous over a *compact* set (Corollary 2.25, Facchinei and Pang, 2003). However, VI model $VI\left(\mathbf{\Phi}, \Re_+^{|\bar{A}|}\right)$ does not fulfill the compactness condition because $\Re_+^{|\bar{A}|}$ is an unbounded set. This section thus aims to build a new VI model using the vector function $\mathbf{\Phi}(\mathbf{u})$ and a non-empty, compact and convex set.

According to the assumptions on OD demand, eqns. (3.6) and (4.3), and also the fact that $\min_{a\in\bar{A}}\{H_a\} > 0$, it can be seen that there is a positive number M_1 such that for any number $K \geq M_1$, it gives

$$D_w(\gamma_1 K) \leq \gamma_2 \times \min_{a\in\bar{A}}\{H_a\}, \forall w \in W, \tag{4.48}$$

where $\gamma_1, \gamma_2 \in (0,1)$ are two given parameters, say $\gamma_1 = 0.5$ and $\gamma_2 = 0.9$. Since multivariate error variable $\zeta_w = (\zeta_{wk}, k \in R_w)^{\mathrm{T}}$ for OD pair $w \in W$ has a strictly positive and continuously differentiable probability density function, a positive number M_2 can be attained such that for any number $K \geq M_2$:

$$\sum_{\substack{k\in R_w \\ k\neq l}} \Pr\left(\zeta_{wk} - \zeta_{wl} \leq \sum_{a\in A} t_a\left(\mathbf{v}^{\max}\right) - (1-\gamma_1)K\right) \leq \frac{\gamma_2 \times \min_{a\in\bar{A}}\{H_a\}}{\left(|W| \times \bar{q}_w\right)}, \forall l \in R_w, \forall w \in W, \tag{4.49}$$

where $|W|$ is the number of OD pairs and vector $\mathbf{v}^{\max} = \left(v_a^{\max} = \sum_{w\in W} \bar{q}_w, a \in A\right)^{\mathrm{T}}$. By taking $M = \max(M_1, M_2) > 0$, a non-empty and compact convex set $\Omega \subset \Re_+^{|\bar{A}|}$ is built as follows:

$$\Omega = \left\{ \mathbf{u} \middle| 0 \le u_a \le M = \max\left(M_1, M_2\right), a \in \bar{A} \right\}. \tag{4.50}$$

Accordingly, a restricted VI model is defined on set Ω using vector function $\mathbf{\Phi}(\mathbf{u})$, denoted by $VI(\mathbf{\Phi}, \Omega)$: Find a vector $\mathbf{u}^* \in \Omega$ such that

$$\mathbf{\Phi}\left(\mathbf{u}^*\right)^{\mathrm{T}} \left(\mathbf{u} - \mathbf{u}^*\right) \ge 0, \forall \mathbf{u} \in \Omega. \tag{4.51}$$

As vector function $\mathbf{\Phi}(\mathbf{u})$ is continuous and Ω is a non-empty, convex and compact set, thus the restricted model $VI(\mathbf{\Phi}, \Omega)$ has at least one solution according to Corollary 2.25 of Facchinei and Pang (2003).

However, equivalence between this new VI model $VI(\mathbf{\Phi}, \Omega)$ and the generalized SUE conditions is not clear. This equivalence condition is rigorously proven, which is concluded in Proposition 4.4.

Proposition 4.4**:** $\mathbf{u}^* = \left(u_a^*, a \in \bar{A}\right)$ is a solution of $VI(\mathbf{\Phi}, \Omega)$ if and only if $\mathbf{u}^*$ and $\mathbf{v}\left(\mathbf{u}^*\right)$ fulfill the generalized SUE conditions (4.5)-(4.8).

Proof.

Necessary condition

Suppose that $\mathbf{u}^* = \left(u_a^*, a \in \bar{A}\right)$ is a solution of $VI(\mathbf{\Phi}, \Omega)$ and it is then shown that $\mathbf{u}^* = \left(u_a^*, a \in \bar{A}\right)$ and $\mathbf{v}\left(\mathbf{u}^*\right)$ fulfills the generalized SUE conditions.

It is first proven that $u_a^* < M, \forall a \in \bar{A}$ by using an apagogical approach as follows. Assume that there is at least one link $b \in \bar{A}$ with $u_b^* = M$. Then, define a specific feasible vector $\mathbf{u}' \in \Omega$:

$$\mathbf{u}' = \left(u_a' = u_a^*, a \in \bar{A} \setminus \{b\}, u_b' = 0.5M\right)^{\mathrm{T}}.$$

After substituting $\mathbf{u}$ in the VI model $VI(\mathbf{\Phi}, \Omega)$ with vector $\mathbf{u}'$, it follows that

$$\left(H_b - v_b\left(\mathbf{u}^*\right)\right) \times \left(0.5M - M\right) \geq 0. \tag{4.52}$$

namely,

$$v_b\left(\mathbf{u}^*\right) \geq H_b. \tag{4.53}$$

Then, it is shown that $v_b\left(\mathbf{u}^*\right) < H_b$ by analyzing the following two cases:

Case 1: Suppose there is at least one path between O-D pair $w \in W$ not passing through the particular link b, say $k_0 \in R_w$, with

$$\lambda_{wk_0}\left(\mathbf{u}^*\right) = \sum_{a \in \bar{A}} u_a^* \delta_{ak}^w \leq \gamma_1 M. \tag{4.54}$$

Let $R_{wb} \subset R_w$ be the set of all paths between OD pair $w \in W$ using link b and $v_{wb}\left(\mathbf{u}^*\right)$ be the sum of traffic flow of all the paths between OD $w \in W$ pair passing though the particular link *b*, namely:

$$v_{wb}\left(\mathbf{u}^*\right) = \sum_{k \in R_{wb}} \left[D_w\left(S_w\left(\Delta_w^{\mathrm{T}} \hat{\mathbf{t}}\left(\mathbf{v}\left(\mathbf{u}^*\right), \mathbf{u}^*\right)\right)\right) \times p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)\right]. \tag{4.55}$$

Thus,

$$v_b\left(\mathbf{u}^*\right) = \sum_{w \in W} v_{wb}\left(\mathbf{u}^*\right). \tag{4.56}$$

According the assumption that OD demand function for OD pair $w \in W$ has the upper bound $\bar{q}_w$ and eqn. (4.55), it follows that

$$\begin{aligned} v_{wb}\left(\mathbf{u}^*\right) &= \sum_{k \in R_{wb}} \left[D_w\left(S_w\left(\Delta_w^{\mathrm{T}} \hat{\mathbf{t}}\left(\mathbf{v}\left(\mathbf{u}^*\right), \mathbf{u}^*\right)\right)\right) \times p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)\right] \\ &\le \bar{q}_w \sum_{k \in R_{wb}} p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right). \end{aligned} \tag{4.57}$$

As the path choice probability

$$\begin{aligned} p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right) = \Pr\Big(& c_{wk}\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \lambda_{wk}\left(\mathbf{u}^*\right) + \zeta_{wk} \le \\ & c_{wl}\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \lambda_{wl}\left(\mathbf{u}^*\right) + \zeta_{wl}, \forall l \in R_w \text{ and } l \ne k\Big), \forall k \in R_{wb}. \end{aligned} \tag{4.58}$$

In the right hand side of eqn. (4.58), in the bracket, the perceived travel time on path $k \in R_w$ must be less than or equal to that on each of *all the other paths* between OD $w \in W$, including the particular path $k_0 \in R_w$. Therefore,

$$\begin{aligned} & p_{wk}\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right) \le \\ & \Pr\left(\zeta_{wk} - \zeta_{wk_0} \le \sum_{a \in A} t_a\left(\mathbf{v}\left(\mathbf{u}^*\right)\right)\delta_{ak_0}^w + \lambda_{wk_0}\left(\mathbf{u}^*\right) - \sum_{a \in A} t_a\left(\mathbf{v}\left(\mathbf{u}^*\right)\right)\delta_{ak}^w - \lambda_{wk}\left(\mathbf{u}^*\right)\right), \forall k \in R_{wb}. \end{aligned} \tag{4.59}$$

For any path $k \in R_{wb}$, it can seen that

$$\begin{aligned} & \sum_{a \in A} t_a\left(\mathbf{v}\left(\mathbf{u}^*\right)\right)\delta_{ak_0}^w + \lambda_{wk_0}\left(\mathbf{u}^*\right) - \left[\sum_{a \in A} t_a\left(\mathbf{v}\left(\mathbf{u}^*\right)\right)\delta_{ak}^w + \lambda_{wk}\left(\mathbf{u}^*\right)\right] \\ & \le \sum_{a \in A} t_a\left(\mathbf{v}^{\max}\right) + \lambda_{wk_0}\left(\mathbf{u}^*\right) - \lambda_{wk}\left(\mathbf{u}^*\right) \\ & = \sum_{a \in A} t_a\left(\mathbf{v}^{\max}\right) + \lambda_{wk_0}\left(\mathbf{u}^*\right) - \left(u_b^* + \sum_{a \in \bar{A} \setminus \{b\}} u_a^* \delta_{ak}^w\right) \\ & = \sum_{a \in A} t_a\left(\mathbf{v}^{\max}\right) + \lambda_{wk_0}\left(\mathbf{u}^*\right) - \left(M + \sum_{a \in \bar{A} \setminus \{b\}} u_a^* \delta_{ak}^w\right) \\ & \le \sum_{a \in A} t_a\left(\mathbf{v}^{\max}\right) - \left(1 - \gamma_1\right) M. \end{aligned} \tag{4.60}$$

According to eqns. (4.57), (4.59) and (4.60), it shows that

$$v_{wb}\left(\mathbf{u}^*\right) \le \overline{q}_w \sum_{k \in R_{wb}} \Pr\left(\zeta_{wk} - \zeta_{wk_0} \le \sum_{a \in A} t_a\left(\mathbf{v}^{\max}\right) - \left(1-\gamma_1\right)M \right), \tag{4.61}$$

As per eqn. (4.49), eqn. (4.61) implies that

$$v_{wb}\left(\mathbf{u}^*\right) \le \frac{\gamma_2 \times \min\limits_{a \in A}\left\{H_a\right\}}{|W|} < \frac{H_b}{|W|}. \tag{4.62}$$

Case 2: Assume $\lambda_{wk}\left(\mathbf{u}^*\right)$ for each path $k \in R_w$ is larger than $\gamma_1 M$, i.e.,

$$\lambda_{wk}\left(\mathbf{u}^*\right) = \sum_{a \in \bar{A}} u_a^* \delta_{ak}^w > \gamma_1 M, \forall k \in R_w. \tag{4.63}$$

Therefore, the satisfaction $S_w\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)$ fulfills the condition:

$$S_w\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right) \ge \gamma_1 M. \tag{4.64}$$

According to the assumption that the OD demand function is non-increasing and eqn. (4.48), it can be seen that

$$D_w\left(S_w\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)\right) \le D_w\left(\gamma_1 M\right) \le \frac{\gamma_2}{|W|} \min_{a \in \bar{A}}\left\{H_a\right\} < \frac{H_b}{|W|}. \tag{4.65}$$

In other words,

$$v_{wb}\left(\mathbf{u}^*\right) = \sum_{k \in R_w} f_{wk}\left(\mathbf{u}^*\right)\delta_{bk}^w \le D_w\left(S_w\left(\mathbf{c}_w\left(\mathbf{f}\left(\mathbf{u}^*\right)\right) + \boldsymbol{\lambda}_w\left(\mathbf{u}^*\right)\right)\right) < \frac{H_b}{|W|}. \tag{4.66}$$

Based on eqns. (4.62) and (4.66), it can be concluded that

$$v_b\left(\mathbf{u}^*\right) = \sum_{w \in W} v_{wb}\left(\mathbf{u}^*\right) < H_b. \tag{4.67}$$

However, eqn. (4.67) contradicts eqn. (4.53), thus the hypothesis $u_b^* = M$ is incorrect. In other words,

$$u_a^* < M, \forall a \in \bar{A}. \tag{4.68}$$

To further prove the necessary conditions, the following two specific vectors for any link $a \in \bar{A}$ are developed:

$$\mathbf{u}'(a) = \left(u'_b = u^*_b, b \in A \setminus \{a\}, u'_a = (1-\gamma_1)u^*_a + \gamma_1 M, a \in \bar{A}\right)^{\mathrm{T}}, \tag{4.69}$$

$$\mathbf{u}''(a) = \left(u''_b = u^*_b, b \in A \setminus \{a\}, u''_b = \gamma_1 u^*_a, a \in \bar{A}\right)^{\mathrm{T}}. \tag{4.70}$$

Since these two vectors are in set Ω, substituting vector $\mathbf{u}$ of VI model $VI(\mathbf{\Phi}, \Omega)$ with these two particular vectors, respectively, yields that

$$\left[H_a - v^*_a(\mathbf{u}^*)\right] \times \left[\gamma_1 \left(M - u^*_a\right)\right] \geq 0, \tag{4.71}$$

$$\left[H_a - v^*_a(\mathbf{u}^*)\right] \times \left[(\gamma_1 - 1)u^*_a\right] \geq 0. \tag{4.72}$$

Since $M - u^*_a > 0$, eqn. (4.71) implies that

$$v^*_a(\mathbf{u}^*) \leq H_a, \forall a \in \bar{A}. \tag{4.73}$$

In accordance with the fact that $u^*_a \geq 0$, it gives that

$$\left(H_a - v^*_a(\mathbf{u}^*)\right)u^*_a \geq 0, \forall a \in \bar{A}. \tag{4.74}$$

Hence, eqn. (4.72) in conjunction with eqn. (4.74) imply that

$$\left(H_a - v^*_a(\mathbf{u}^*)\right)u^*_a = 0, \forall a \in \bar{A}. \tag{4.75}$$

In other words, $\mathbf{u}^*$ and $\mathbf{v}(\mathbf{u}^*)$ fulfill the generalized SUE conditions.

Sufficient condition

Suppose that $\mathbf{u}^*$ and $\mathbf{v}(\mathbf{u}^*)$ satisfy the generalized SUE conditions expressed by eqns. (4.5)-(4.8). And using the similar tactic to eqns. (4.52) to (4.68), it can be shown that $0 \le u_a^* < M, \forall a \in \bar{A}$. In addition, eqns. (4.5) to (4.8) imply that

$$\mathbf{H} - \mathbf{v}_{\bar{A}}(\mathbf{u}^*) \ge \mathbf{0}, \tag{4.76}$$

$$\left(\mathbf{H} - \mathbf{v}_{\bar{A}}(\mathbf{u}^*)\right)^{\mathrm{T}} \mathbf{u}^* = 0. \tag{4.77}$$

For any $\mathbf{u} \in \Omega$, it shows that $\mathbf{u} \ge 0$. Thus, according to eqns. (4.76)-(4.77), it follows that

$$\left(\mathbf{H} - \mathbf{v}_{\bar{A}}(\mathbf{u}^*)\right)^{\mathrm{T}} \mathbf{u} \ge \left(\mathbf{H} - \mathbf{v}_{\bar{A}}(\mathbf{u}^*)\right)^{\mathrm{T}} \mathbf{u}^*. \tag{4.78}$$

Rearranging eqn. (4.78) yields that

$$\left(\mathbf{H} - \mathbf{v}_{\bar{A}}(\mathbf{u}^*)\right)^{\mathrm{T}} (\mathbf{u} - \mathbf{u}^*) \ge 0. \tag{4.79}$$

That is,

$$\mathbf{\Phi}(\mathbf{u}^*)^{\mathrm{T}} (\mathbf{u} - \mathbf{u}^*) \ge 0. \tag{4.80}$$

It then proves the sufficient condition. □

Since $VI(\mathbf{\Phi}, \Omega)$ has at least one solution, Proposition 4.4 therefore confirms the existence property of solution for the PA-SUEED problem with link capacity constraints. Moreover, since the first VI model $VI\left(\mathbf{\Phi}, \Re_+^{|\bar{A}|}\right)$ is equivalent to the generalized SUE conditions, thus $VI\left(\mathbf{\Phi}, \Re_+^{|\bar{A}|}\right)$ also has at least one solution, with the aid of $VI(\mathbf{\Phi}, \Omega)$. As vector function $\mathbf{\Phi}(\mathbf{u})$ is only monotone rather than strictly or strongly monotone,

solution to the restricted VI model $VI(\mathbf{\Phi},\Omega)$ may not be unique. In other words, the optimal Lagrangian multipliers involved in generalized SUE conditions are not unique. While, it should be pointed out the SUE link flow solution is unique for any given optimal Lagrangian multiplier solutions.

The three important properties of vector function $\mathbf{\Phi}(\mathbf{u})$ are all available for $VI(\mathbf{\Phi},\Omega)$. Its monotone and uniform Lipschitz-continuous properties are prerequisites for the global convergence of many solution algorithms. To sum up, the new VI model $VI(\mathbf{\Phi},\Omega)$ not only helps in demonstrating the existence of solution to PA-SUEED with link capacity constraint, but also inherits the three important properties proved in Section 4.3.1. These properties can guarantee the global convergence of some algorithms for solving the VI models.

4.4 Solution Algorithm

The projection operation on set Ω of VI model $VI(\mathbf{\Phi},\Omega)$ is also effortless, which is solving a PA-SUEED problem with generalized link travel time functions. Thus, it is convenient to choose the convergent self-adaptive Prediction-Correction (PC) algorithm proposed by He and Liao (2002) for solving $VI(\mathbf{\Phi},\Omega)$, which was previously used in Chapter 3. For completeness, procedures of self-adaptive PC for solving $VI(\mathbf{\Phi},\Omega)$ are summarized as follows:

Step 0: (Initialization). Choose an initial vector $\mathbf{u}^{(1)} = \left(u_a^{(1)} = 0, a \in \bar{A}\right)^{\mathrm{T}}$, three constants $0 < \kappa_2 < \kappa_1 < 1$, $\gamma \in (0,2)$, and initial step size $\eta^{(1)} > 0$. Let the number of iterations $n = 1$.

Step 1: (Projection with step size adjustment). Find vector $\bar{\mathbf{u}}^{(n)}$ with a proper step size $\eta^{(n)}$ through the following procedure:

Step 1.1: For vector $\mathbf{u}^{(n)}$, first calculate the parametric SUE link flow vector $\mathbf{v}\left(\mathbf{u}^{(n)}\right) = \left(v_a\left(\mathbf{u}^{(n)}\right), a \in A\right)^{\mathrm{T}}$, which is a PA-SUEED problem with generalized link travel time functions. Then compute $\mathbf{\Phi}\left(\mathbf{u}^{(n)}\right) = \left(H_a - v_a\left(\mathbf{u}^{(n)}\right), a \in \bar{A}\right)^{\mathrm{T}}$.

Step 1.2: Find vector $\bar{\mathbf{u}}^{(n)}$ by the projection:

$$\bar{\mathbf{u}}^{(n)} = P_{\Omega}\left[\mathbf{u}^{(n)} - \eta^{(n)}\mathbf{\Phi}\left(\mathbf{u}^{(n)}\right)\right]. \tag{4.81}$$

Step 1.3: For the vector $\bar{\mathbf{u}}^{(n)}$ obtained in Step 1.2, calculate parametric SUE link flow solution $\mathbf{v}\left(\bar{\mathbf{u}}^{(n)}\right) = \left(v_a\left(\bar{\mathbf{u}}^{(n)}\right), a \in A\right)^{\mathrm{T}}$ and then calculate $\mathbf{\Phi}\left(\bar{\mathbf{u}}^{(n)}\right) = \left(H_a - v_a\left(\bar{\mathbf{u}}^{(n)}\right), a \in \bar{A}\right)^{\mathrm{T}}$.

Step 1.4: Calculate ratio $r^{(n)}$:

$$r^{(n)} = \eta^{(n)}\left\|\mathbf{\Phi}\left(\mathbf{u}^{(n)}\right) - \mathbf{\Phi}\left(\bar{\mathbf{u}}^{(n)}\right)\right\|_2 \Big/ \left\|\mathbf{u}^{(n)} - \bar{\mathbf{u}}^{(n)}\right\|_2. \tag{4.82}$$

If $r^{(n)} \le \kappa_1$, go to Step 2;otherwise go to Step 1.5

Step 1.5: Reduce the step size according to

$$\eta^{(n)} = \frac{2}{3}\eta^{(n)} \min\left\{1, \frac{1}{r^{(n)}}\right\}, \tag{4.83}$$

and go to Step 1.1

Step 2: (Stop check). If the following condition is fulfilled, then stop; otherwise, go to Step 3.

$$\left\|\mathbf{u}^{(n)} - \bar{\mathbf{u}}^{(n)}\right\| \le \varepsilon_2, \tag{4.84}$$

where ε_2 is a predetermined positive tolerance.

Step 3: (Correction with self-adaptive step size adjustment). Based on $\mathbf{u}^{(n)}$, $\bar{\mathbf{u}}^{(n)}$ and $\eta^{(n)}$, calculate a proper step size $\alpha^{(n)}$ for correction and then get an updated vector $\mathbf{u}^{(n+1)}$:

Step 3.1: Calculate another step size $\alpha^{(n)}$ as per the formula:

$$\alpha^{(n)} = \gamma \times \eta^{(n)} \times \sum_{a\in\bar{A}}\left[\left(u_a^{(n)} - \bar{u}_a^{(n)}\right) \times h_a^{(n)}\right] \Big/ \sum_{a\in\bar{A}}\left(h_a^{(n)}\right)^2. \tag{4.85}$$

where

$$h_a^{(n)} = \left(u_a^{(n)} - \bar{u}_a^{(n)}\right) + \eta^{(n)}\left(\Phi_a\left(\mathbf{u}^{(n)}\right) - \Phi_a\left(\bar{\mathbf{u}}^{(n)}\right)\right), a \in \bar{A}. \tag{4.86}$$

Step 3.2: Update the vector $\mathbf{u}^{(n+1)}$ by this projection:

$$\mathbf{u}^{(n+1)} = P_\Omega\left[\mathbf{u}^{(n)} - \alpha^{(n)}\boldsymbol{\Phi}\left(\bar{\mathbf{u}}^{(n)}\right)\right]. \tag{4.87}$$

Step 3.3: Enlarge step size $\eta^{(n)}$ according to the following scheme:

$$\eta^{(n+1)} = \frac{3}{2}\eta^{(n)} \quad \text{if} \quad \eta^{(n)}\left\|\boldsymbol{\Phi}\left(\mathbf{u}^{(n)}\right) - \boldsymbol{\Phi}\left(\bar{\mathbf{u}}^{(n)}\right)\right\|_2 \Big/ \left\|\mathbf{u}^{(n)} - \bar{\mathbf{u}}^{(n)}\right\|_2 \le \kappa_2. \tag{4.88}$$

Let $n = n+1$ and go to Step 1.

The uniform Lipschitz-continuity of function vector $\boldsymbol{\Phi}(\mathbf{u})$ ensures that condition $r^{(n)} \le \kappa_1$ in Step 1.4 can be fulfilled in the finite iterations, and moreover $\inf\left\{\eta^{(n)}\right\} = \eta_{\min} > 0$ (He and Liao, 2002). The stop criterion utilized in Step 2 comes from the fact that $\mathbf{u}^*$ is a solution of VI model $VI(\boldsymbol{\Phi}, \Omega)$ if and only if (Nagurney, 1993):

$$\mathbf{u}^* = P_\Omega\left[\mathbf{u}^* - \rho\boldsymbol{\Phi}\left(\mathbf{u}^*\right)\right] \tag{4.89}$$

where ρ is an arbitrary positive parameter. In reality,

$$\left\|\mathbf{u}^{(n)} - \overline{\mathbf{u}}^{(n)}\right\|_2 = \left\|\mathbf{u}^{(n)} - P_\Omega\left[\mathbf{u}^{(n)} - \eta^{(n)}\boldsymbol{\Phi}\left(\mathbf{u}^*\right)\right]\right\|_2 \tag{4.90}$$

is an error bound value of $\mathbf{u}^{(n)}$ and it would equal to zero only if $\mathbf{u}^{(n)}$ is a solution of VI model $VI(\boldsymbol{\Phi}, \Omega)$.

4.5 Numerical Experiment

To numerically verify the effectiveness of the proposed VI model and PC algorithm, the Sioux-Falls network is adopted in this section. The parameters involved in the PA-SUEED problem take the same value as those in Section 3.6, if they are not further emphasized here. The asymmetric link travel time functions and demand functions also follow eqn. (3.82) and (3.86), respectively. Data for the OD demands and link attributes of Sioux-Falls network are identical to those in Section 3.6.2.

It is further assumed that only the links in set $\overline{A} = \{3, 10, 15, 30, 50, 70\}$ are restricted by capacity constraints. Let h_a denote the physical capacity constraint on these links, and each link in set $\overline{A}$ is restricted by another threshold constraint, denoted by $H_a^s \le h_a$.

Three scenarios of the threshold constraints are adopted to test their impact on the solutions, in each scenario it is required that:

$$v_a \le H_a^s, a \in \bar{A}, s = 1,2,3 \tag{4.91}$$

where threshold H_a^s of link a for Scenario s ($s = 1,2,3$) as well as its physical capacity h_a are given in Table 4.1.

Table 4.1 Three Scenarios of Link Capacity Constraints

Link ($a \in \bar{A}$)	Scenario 1 Threshold (H_a^1)	Scenario 2 Threshold (H_a^2)	Scenario 3 Threshold (H_a^3)	Physical Capacity (h_a)
3	15000	5000	3000	25900
10	4000	3500	3000	4908
15	4500	4000	3000	4948
30	4500	4000	3000	4993
50	12000	9000	8000	19679
70	4500	4000	3000	5000

As shown in Table 4.1, these three scenarios are all strictly less than the physical capacity of each constrained link. From scenario 1 to scenario 3, the values of threshold are getting smaller implying a more restricted network management. Note that the values listed here are all assumed by the author, while in practice they should be predetermined by the network authorities.

As suggested by He and Liao (2002), the parameters in the self-adaptive PC algorithm are set as: $\zeta = 0.1$, $\omega = 0.9$, $\gamma = 1.8$, $\eta^{(0)} = 1.0$. The tolerance value in its stop criterion is taken as $\varepsilon_2 = 10^{-2}$. In order to see the convergent trend of the self-adaptive PC algorithm

incorporating CA method more precisely and apparently, the following logarithmic value of the error bound shown in eqn. (4.90) is taken as a performance index:

$$\text{Logarithmic value of the error bound} = \log_{10}\left(\left\|\mathbf{u}^{(n)} - \bar{\mathbf{u}}^{(n)}\right\|_2\right) \tag{4.92}$$

Figure 4.1 depicts the logarithmic value of the error bound versus the CPU times used for solving each scenario. It clearly shows that the self-adaptive PC algorithm can monotonically converge to a solution of VI model $VI(\mathbf{\Phi}, \Omega)$.

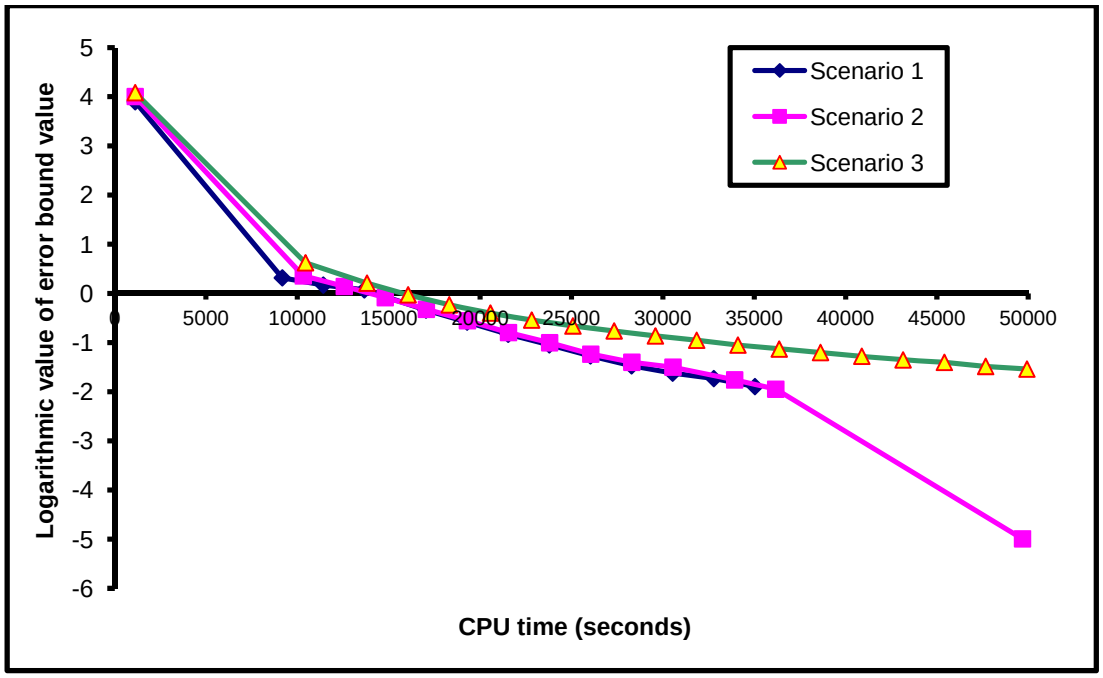

Figure 4.1 Performance of the Self-adaptive PC Algorithm for Solving the Three Scenarios

Table 4.2 gives the resultant Lagrangian multiplier vector $\mathbf{u}^*$ and corresponding flow-threshold ration, $v_a(\mathbf{u}^*)/H_a^s, a \in \bar{A}$, for each scenario $s = 1, 2, 3$. Table 4.2 indicates that the flow-threshold ratio does not exceed 1.0 and the generalized Lagrangian multiplier takes positive value only on the links with a flow-threshold ratio close to 1.0. These two

phenomena tally with eqns. (4.6)-(4.8), which numerically verify the effectiveness of VI model $VI(\mathbf{\Phi},\Omega)$ and self-adaptive PC algorithm. In addition, it can be seen that as the values of threshold are getting larger from Scenario 1 to Scenario 3, the relevant optimal Lagrangian multipliers in general become larger. This implies that it would be more costly to restrict the link flow to a much lower level.

Table 4.2 SUE Link Flows and Optimal Lagrangian Multipliers

Link	**Scenario 1**		**Scenario 2**		**Scenario 3**	
$(a \in \bar{A})$	$v_a(\mathbf{u}^*)/H_a^1$	u_a^*	$v_a(\mathbf{u}^*)/H_a^2$	u_a^*	$v_a(\mathbf{u}^*)/H_a^3$	u_a^*
3	0.31	0.00	0.96	0.00	1.01	4.37
10	1.00	2.95	1.00	1.87	1.00	2.76
15	1.01	4.81	1.00	5.56	1.01	7.84
30	1.00	2.79	1.00	3.36	1.01	4.37
50	1.00	0.93	1.00	1.82	1.00	2.57
70	1.01	4.40	1.01	3.49	1.00	4.48

4.6 Conclusions

This chapter dealt with the PA-SUEED problem with link capacity constraints, which is an open question with considerable theoretical contributions. After defining the generalized SUE conditions in terms of the generalized link travel times, a VI model was defined for the generalized SUE conditions by using the parametric SUE link flow solution. It was rigorously proven that the VI model is equivalent to the generalized SUE conditions. More importantly, this VI model is monotone and uniform Lipschitz-continuous. The convergent self-adaptive PC algorithm was then employed for solving the VI model, which was further validated by a numerical example.

Further efforts are required to extend the methodology proposed in this chapter to some other link capacity constrained problems, including the dynamic SUE problems with link capacity constraints and transit assignment in terms of probit-based SUE with link capacity constraints.

CHAPTER 5 DISTRIBUTED COMPUTING APPROACHES FOR SOLVING PA-SUEED

This chapter aims to further improve the computational speed of solving PA-SUEED by virtue of distributed computing. As shown by Tables 3.1 and 3.5, the two proposed hybrid PC-CA algorithms can reduce the execution time for around 10 times, compared with the existing CA method. Although such an improvement is already impressive, the distributed computing approach can further accelerate the computational speed for over 50 times, as verified by the numerical experiments in this chapter.

5.1 Background

As discussed in Section 3.4, the stochastic network loading (SNL) plays a vital role in the solution algorithms for probit-based SUE problems, which is similar to the all-or-nothing assignment in solution algorithms for DUE problems. A link-based two-stage Monte Carlo simulation-based method has been proposed for the SNL problem in this study. With a sufficient sample size (usually larger than 1000), the accuracy of this method can be guaranteed, yet it has also significantly prolonged the execution time, especially for large-scale networks. Based on some many numerical experiments, it shows that the cumulative computational time of Monte Carlo simulation is at least 95% of the total CPU time used for solving the PA-SUEED problem. Hence, computational time would be largely reduced if the calculation of Monte Carlo simulation could be accelerated.

As claimed in Section 2.1.4, each trial of the Monte Carlo simulation method for solving SNL is independent, and it thus possess a perfect parallelism. Moreover, in each trial of

the Monte Carlo simulation, the computational efforts are in general allocated on solving a shortest path problem between each OD pair, see Steps 1.2 and 2.2 in Section 3.4.1. The shortest path problem is also a popular topic for distributed computing. Thus, calculation for various components of the Monte Carlo simulation can also be executed independently and simultaneously, based on a distributed computing system. All the processors in the distributed computing system would perform the simulation concurrently based on different random numbers. Three approaches are thus proposed in this chapter regarding partitioning workload of the Monte Carlo simulation method for distributed computing. All of these three distributed computing approaches are Single-Program Multiple Data (SPMD) paradigm, where each processor runs the same program based on different subset of data.

It should be noted that this chapter concentrates on the algorithmic issues of the three distributed computing approach in accelerating the computational speed of SNL of PA-SUEED, rather than the implementation issues including: (a) sampling techniques and distribution of random numbers on different processors, (b) hardware architecture of the distributed computing system, (c) and the software aspects for data communication (Ma, 1994). Although these issues can further improve the computational speed of Monte Carlo method, they have similar impacts to the three approaches.

The outline of this chapter is shown as follows. Section 5.2 presents three distributed computing approaches. The computation platform as well as performance measures of the distributed computing approaches are provided in Section 5.3. Three network examples

are employed in Section 5.4 to numerically test the performance of the distributed computing approaches. Conclusions of this chapter are presented in Section 5.5.

5.2 Three Distributed Computing Approaches

Definitions and notation of the transportation network are similar to those in Chapter 3, which are thus not repeated in this section. For conciseness, the three distributed computing approaches are directly introduced as follows, based on the transportation network $G = (N, A)$.

5.2.1 Distributed Loading Approach

As discussed in Section 5.1, a parallel-processing procedure on distributed computing system is an ideal and straightforward solution for the two-stage Monte Carlo simulation based SNL method. This is because each individual sampling and trial of the Monte Carlo simulation is independent, per se, and follows identical procedures. Moreover, during the calculation of each trial, no data communication is required, and therefore the impact of communication time would be quite marginal. Thus, based on a distributed computing system with k processors, we can partition the n_0 (or m_0) trials of the first (or second) stage of Monte Carlo simulation to each processor equably, and execute the calculation in parallel. For the sake of presentation, this approach for the workload partition of the Monte Carlo simulation based on the distributed computing system is called as Distributed Loading (DL).

For this DL approach, two stages of the Monte Carlo simulation are still calculated sequentially. This is because results of all the trials in the first stage are utilized to

calculate the value of travel demand and then taken as an input for all the simulations in the second stage. Suppose k processors in the distributed computing system are available for the calculation, and then all the n_0 (m_0) trials for the first (second) stage Monte Carlo simulation are equally partitioned into k groups. Sample sizes of the simulation, n_0 and m_0, are usually much greater than the number of processors, k. Thus, if k is a sub-multiple number of n_0 and m_0, the Monte Carlo simulation would have a perfect parallelism, in that each processor will conduct the same number of trials and their workloads are identical.

5.2.2 Distributed Shortest-Path Approach

As shown by Steps 1.2 and 2.2 in Section 3.4.1, in each trial of the Monte Carlo simulation, most of the computational efforts are used to calculate many shortest path problems. Calculation of the shortest path problem is also a popular topic for distributed computing in the literature. Hence, the distributed computing system can also be utilized to solve the shortest path problem in parallel, and then sequentially execute each trial of the Monte Carlo simulation. And this second approach for workload partition of the Monte Carlo simulation is referred to as Distributed Shortest-Path (DSP).

Note that calculation of shortest path problems would be largely influenced by the solution algorithm. Thus, it is of considerable significance to find an efficient algorithm for shortest path problems on transportation network. Discussions of shortest path algorithms mainly focus on Dijkstra's Label-setting algorithm (Dijkstra, 1959), with a time complexity of $\mathbf{O}\left(|N|^2\right)$, and Moore's Label-correcting algorithm (Moore, 1957)

whose time complexity is $\mathbf{O}(|A| \bullet |N|)$. Herein, $|N|$ and $|A|$ are, respectively, the total numbers of nodes and links on the network. The time complexity provided here is the worst case execution time. Thus, although $|A| \bullet |N|$ is larger than $|N|^2$ for most of the transportation networks, many previous studies based on numerous experimental tests conclude that Label-correcting algorithm is more efficient than Label-setting algorithm (Pape, 1974; Hribar et al., 2001). Calculation of these two algorithms can be further accelerated by plenty of extensions that use more advanced data structure; for example using balanced binary tree to store the list of labels in Label-setting algorithm. For particular type of network sizes and configurations, some extensions of Dijkstra's Label-setting algorithm are validated to be more preferable. While, the numerical tests conducted by Van Vliet (1978) showed that for both small- and large- scale networks, the D'Esopo algorithm is, in general, more efficient. D'Esopo algorithm is an extension of Moore's Label-correcting algorithm by using two-ended lose-end table (Pollack and Wiebenson, 1960), and each time it can provide shortest paths from one source to all the others nodes on the network, usually referred to as building trees. Accordingly, in this chapter, the D'Esopo algorithm is taken for the calculation of shortest path algorithms.

Two strategies are commonly used for solving the shortest path algorithms in parallel on a distributed computing system: (1) network decomposition (Habbal et al., 1994; Traff, 1995; Hribar et al., 2001) and (2) network replication (Kumar and Singh, 1991; O'Cearbhaill and O'Mahony, 2005). The first strategy, network decomposition, partitions the network into small pieces and then assigns each processor to solve the shortest path problem on each sub-network. Shortest paths of one sub-network are updated if labels of

nodes on a nearby sub-network change. While, the second strategy, network replication, aims to assign the entire network to each processor and each processor then solve the shortest path problems for different sources (origins). Although network decomposition can make use of the cumulative memory of all the processors and its degree of concurrency is not limited by the number of origins, there are high communication overheads for checking node labels and termination detection. Therefore, when the number of processors is not evidently larger than the number of origins, the network replication is found to be superior to the network decomposition strategy (Florian and Gendreau, 2001).

Thereby, in this chapter, the network replication strategy in conjunction with D'Esopo algorithm is selected for the DSP distributed computing approach. Note that the network replication strategy with D'Esopo algorithm was originally used by O'Cearbhaill and O'Mahony (2005) for solving the shortest path problems in the well-known network analysis model SATURN. However, it should be highlighted that when the number of processors is larger than the number of sources, the redundant processors would keep in idle. In addition, since the number of sources is usually much less than the sample size of Monte Carlo simulation, the concurrency level of DSP is thus lower than that of DL.

5.2.3 Integrated Loading Approach

A third approach is proposed, named as Integrated Loading (IL), for workload partition of the Monte Carlo simulation. IL is a mixture of DL and DSP, i.e., both the Monte Carlo trials and shortest path problems in each trial are calculated simultaneously by different processors. Firstly, all the k processors are divided into i groups and let $j_x, x = 1, 2, \cdots, i$

denote the size of group x. Thus, $\sum_{x=1}^{i} j_x = k$. Then, only i trials of Monte Carlo simulation are conducted in parallel by each group of processors. Meanwhile, all the j_x processors in group x simultaneously calculate the shortest paths problems for the Monte Carlo simulation assigned to this group by virtue of the DSP approach. The calculation then will be conducted in turns until all the n_0 (m_0) trials for the first (second) stage of the Monte Carlo simulation are accomplished.

Evidently, level of parallelism of this approach is largely influenced by the number of processors in each group, $j_x, x = 1, 2, \cdots, i$. If this number j_x is equal for all the groups, then workload for each processor is identical. Thus, j_x is decided to be a constant for each group, denoted by $\overline{j}$. In principle, the number of processors in each group $\overline{j}$ should also not exceed the number of origins (sources) in the network. When adopted to calculate the Monte Carlo simulation method for stochastic network loading map, the IL approach contains a better parallelism than DL and DSP, since its concurrency level is a product of that of DL and DSP. Thus, if the number of available processors is greater that the sample size of Monte Carlo simulation as well as number of sources of the transportation network, the proposed IL approach would inherently outperform the other two.

As aforementioned, implementation of PA-SUEED problem on large-scale network is quite restricted, due to the hurdle in solving its stochastic network loading map. All the three approaches discussed in this section can inherently accelerate computing speed of

the Monte Carlo simulation method, in contrast with the sequential procedure. Thus, the three solution algorithms introduced in Chapter 3 incorporating any of these three approaches should be quite efficient for solving the PA-SUEED problem. This chapter aims to search for the most efficient one among these three approaches, which would then be recommended for the calculation of PA-SUEED problem on large-scale networks. However, other than the endogenetic concurrency level, their performance might be also quite sensitive to the number of processors as well as network structure. Thus, an in-depth numerical test is necessary for these three approaches. In the remainder two sections we first discuss about the computing platform as well as performance measures adopted, and then show about the numerical tests of the distributed computing approaches.

Remark 5.1: These three approaches are differentiated by its workload partition strategies. Before the workload partition, the values of random link travel times are sampled by pseudo random numbers from the whole population space. Such a sampling technique is called as Simple Random sampling (Cochran, 1977). Note that accuracy of the Monte Carlo simulation can be further improved, if some advanced sampling techniques are used, for instance, stratified sampling, importance sampling, antithetic sampling (e.g., Fishman and Shaw, 1989; Liu and Meng, 2011). Therefore, if an accuracy level is predetermined, these sampling techniques would reduce the sample size required and hence further reduce the execution time. However, these sampling techniques have identical impacts on the three distributed computing approaches, and it would not change the superiority of any distributed computing approach over the rest. Therefore, for

conciseness and without loss of generality, only the Simple Random sampling is used in this study.

5.3 Computing Platform and Performance Measures

5.3.1 Computing Platform

The computing platform used in this study is a Distributed Computing PC Cluster system in Civil Engineering department at National University of Singapore. This system has 60 computer nodes, and each of which uses Intel® Core i7 940 (Quad Core) processor, with a clock speed of 2.93GHz, 256kB L2 cache per core and 8MB L3 cache and 12GB of 1333MHz DDR3 RAM. The 10G Myrinet technology and corresponding products are utilized to build the Local Area Network, where a convergence at 10-Gigabit/s data rates is available. This network follows IEEE 802.3 10G Ethernet protocol. A 10G Myrinet spine switch is taken as the main network communication channel, which inherently improved the velocity of data communication. All the computer nodes are then connected to this spine switch. The data for inter-node communication are first compressed and then broadcast to the other nodes via the connecting cable and the spine switch.

Operating system on all the computer nodes is Windows High Performance Computing (HPC) Server 2008, which is developed on the x64-based version of the Windows Server 2008 operating system. A HPC Cluster Manager installed on every node is taken as the central tool for configuring, deploying, and administering of the cluster. The distributed computing programs are then submitted to the distributed computing cluster through the HPC Cluster Manger.

For the single program multiple data (SPMD) distributed computing approaches, they can be simply implemented on the HPC cluster. Each node would operate the program synchronously based on its own part of data and communicate with each other by virtue of the well-known Message Passing Interface (MPI) (Gropp et al., 1999). MPI is a message-passing application programmer interface, together with a protocol and semantic specifications for how its features behave in any implementation. Both point-to-point and collective communication are supported. Goals of MPI are high performance, scalability, and portability, and it remains the dominant model used in high-performance computing today. On the HPC PC cluster, a full-featured implementation of the MPI specification called Microsoft MPI is adopted. Microsoft MPI acts as a function library for programming language C, FORTRAN 77, and FORTRAN 90. Without loss of generality, the programs in this chapter are all coded in FORTRAN 90.

5.3.2 Three Performance Measures

Several performance measures are used in the literature to evaluate the performance of parallel (distributed) computing implementations, among which the most commonly used one is called Speed-Up that is a criterion for the execution time. The value of Speed-Up for k processors can be defined as (see, e.g., Nagel and Rickert, 2001; Wong et al., 2001)

$$S(k) = \frac{T_1}{T_k} \tag{5.1}$$

where T_1 is the execution time used by only one processor for solving the algorithm, and T_k is the execution time when k processors involve the calculation. Note that as the number of processors increases, the efforts spent in data communication would keep increasing. Thus, Speed-Up should be a sub-linear function of the number of processor.

Another performance measure employed is the Efficiency of processors, which is calculated by:

$$E(k) = \frac{S(k)}{k} \tag{5.2}$$

Ideally, $E(k)$ should be equal to 1. But due to the additional overhead caused by the communication and synchronization mechanisms, $E(k)$ is less than 1 in practice. When the number of processors exceeds the concurrency level, some processors would be idle. Due to these reasons, value of $E(k)$ usually decreases as the number of processors increases.

The third performance measure used in this chapter is called Relative Burden (Tremblay and Florian, 2001), and it can be expressed by

$$B(k) = \frac{T_k}{T_1} - \frac{1}{k} = \frac{1}{S(k)} - \frac{1}{k} \tag{5.3}$$

The burden of distributed computing refers to all the time overhead spent by the processors; for instance, the inter-node communication time, total idle time of the processors, time spent on calculating the non-parallel parts of the algorithm, etc. In the ideal case, $S(k)$ equals to $\frac{1}{k}$, and the Relative Burden then becomes zero.

Execution time in conjunction with these three performance measures are taken as criterions for the efficiency of the CA method incorporating each of the three distributed

computing approaches respectively for calculating the numerical network examples in Section 5.4.

5.4 Numerical Examples

Although the three distributed computing approaches have different concurrency level, their performance is unpredicted and might be sensitive to the number of processors used. Thus, this section aims to comprehensively test the three approaches using two middle-scale network examples and then select the most efficient approach to accelerate the calculation of a large-scale example. The Sioux-Falls network is first utilized for the test, which has 24 nodes, 76 links and 528 OD pairs. Note that performance of the distributed computing approaches may be influenced by the particular structure of Sioux-Falls network. Thereby, a randomly generated network example is further utilized for the test, which is generic to any network structure.

The most efficient approach is then chosen to calculate the PA-SUEED problem on a large-scale network, called as Anaheim network, which contains 416 nodes, 914 links and 1406 OD pairs. Detailed data for the network attributed of Sioux-Falls network and Anaheim network can be found in the website of transportation network test problems built by Bar-Gera (2011), which are not elaborated here due to the space limit.

It should be noted that the three distributed computing approaches has accelerate the computation of SNL problem, which is taken as a subroutine by any of the three algorithms introduced in Chapter 3. Thus, these distributed computing approaches have identical impacts on the three algorithms for solving PA-SUEED problem. Without loss

of generality, the Cost-Averaging (CA) method is taken for the numerical test in this section due to its concise procedures, while all the conclusions are also effective to the two hybrid PC-CA algorithms.

For the first example, CA method incorporating the three proposed distributed computing approaches will be tested in turns to calculate the probit-based SUE problem. Then, the most efficient approach is selected for the calculation of a large-scale example, Anaheim network. In order to conduct an unbiased comparison, workload for these three approaches should be equivalent. Thus, the number of iterations for the CA method is fixed to be 200, and based on a trial-and-error test, 200 iterations would give rise to a satisfactory convergence level for both of the two examples. Sample sizes of the two-stage Monte Carlo simulations for stochastic network loading are decided based on the estimator proposed in Section 3.4.2. The maximal values for estimation error, ε_1 and ε_2, involved are taken as 5 percent of total demand $\bar{q}_w$ or link-flow capacity h_a, respectively. Then, based on some empirical tests, a sufficient sample size for the first stage of Monte Carlo simulation when solving each numerical example is found to be 100. While, for the second stage in terms of Sioux-Falls network, random graph example and Anaheim network, the sample size is decided to be 8050, 3100 and 1050, respectively.

The asymmetric link travel time functions and OD demand functions still follow those introduced in Section 3.6, and all the parameters involved are also unchanged if they are not further emphasized.

5.4.1 Sioux-Falls Network

In each iteration of CA method, the three different distributed computing approaches are utilized respectively to solve the two-stage Monte Carlo simulation in parallel. In this case, the Distributed Loading (DL) approach will take all the processors involved to simultaneously calculate the 100 and 8050 trials of the two stages of Monte Carlo simulation. The Number of origins for Sioux-Falls network amounts to 24. Thus, within one trial of the simulation, 24 one-to-all shortest-path problems must be handled. The second distributed computing approach, Distributed Shortest-Path (DSP), thus allocates all the processors to synchronously calculate these 24 shortest-path problems. Apparently, concurrency level of DSP approach is inherently restricted by the number of origins, since the number of processors in use cannot exceed 24. While, for the Integrated Loading (IL) approach, all the processors are equally divided into different groups. Each group takes charge of one trial out of 100 or 8050 simulations, and the processors within one group are required to solve the 24 shortest-path problems in parallel. With the purpose of a better concurrency level, group size is determined to be 8, in that it is a sub-multiple of 24 and workload for each processor within one group is thus identical.

Tables 5.1 to 5.3 here illustrate the CPU (execution) time used by the CA method incorporating different distributed computing approaches. Different numbers of processors are sequentially tested for each distributed computing approach. For the test of IL approach, number of processors cannot be less than the group size, i.e., 8. Thus, the value of execution time used by only one processor, T_1, is taken as that used by the DL approach, which is 3307.2 seconds. Based on the CPU time, values of the other three

performance measures (see Section 5.3.2) are calculated for each test. It is worthwhile pointing out that the output link flows for each test of the calculations are identical, which supports the preciseness for the results of each test.

Table 5.1 Test of DL Approach for Solving Sioux-Falls Network Example

No. of processors	CPU time (Seconds)	Speed-Up	Efficiency	Relative Burden
1	3307.200	1.000	1.000	0.000
2	2922.034	1.134	0.567	0.382
4	1461.395	2.267	0.567	0.191
8	731.920	4.526	0.566	0.096
16	367.745	9.007	0.563	0.049
32	184.323	17.970	0.562	0.024
64	93.109	35.575	0.556	0.012
120	50.736	65.286	0.544	0.007
128	47.423	69.847	0.546	0.007

Table 5.2 Test of DSP Approach for Solving Sioux-Falls Network Example

No. of processors	CPU time (Seconds)	Speed-Up	Efficiency	Relative Burden
1	3312.330	1.000	1.000	0.000
2	2915.699	1.136	0.568	0.380
4	1482.071	2.235	0.559	0.197
8	761.178	4.352	0.544	0.105
16	513.781	6.447	0.403	0.093
32	268.218	12.349	0.386	0.050
64	266.518	12.428	0.194	0.065
120	270.499	12.245	0.102	0.073
128	268.045	12.357	0.097	0.073

Table 5.3 Test of IL Approach for Solving Sioux-Falls Network Example

No. of processors	CPU time (Seconds)	Speed-Up	Efficiency	Relative Burden
1	3307.200	1.000	1.000	0.000
16	384.629	8.598	0.537	0.054
32	193.055	17.131	0.535	0.027
64	97.269	34.001	0.531	0.014
120	52.584	62.894	0.524	0.008
128	52.334	63.194	0.494	0.008

We note that due to a large sample size for the second stage of Monte Carlo simulation (8050), even for this small size network, CPU time is around 3300 seconds when only one processor is used for the calculation. This CPU time is hardly acceptable, and it would geometrically increase as the network size expands.

To get a better view for the acceleration of CA method incorporating three distributed computing approaches when using different number of processors, values of Speed-Up in Tables 5.1 to 5.3 are used to chart the trend of Speed-Up over different number of processors, as shown in Figure 5.1. Based on the data shown in Tables 5.1 to 5.3 as well as Figure 5.1, we can see the following phenomena:

(a) compared with T_1, execution time is inherently reduced when multiple processors are used for the calculation;

(b) for each test with the same number of processors, execution time of DL is slightly less than that of IL, while when the number of processors is larger than 24, performance of DL and IL are much superior to that of DSP;

(c) as the number of processors increases, values of Speed-Up for DL and IL approach linearly increase, while Speed-Up value of DSP keeps still when more than 24 processors are used;

(d) Efficiency of DL and IL are, in general, unchangeable, yet efficiency of DSP keeps decreasing especially when the number of processors is larger than 24;

(e) Relative Burden of DL and IL keep decreasing, while value of Relative Burden for DSP first decreases and then goes up.

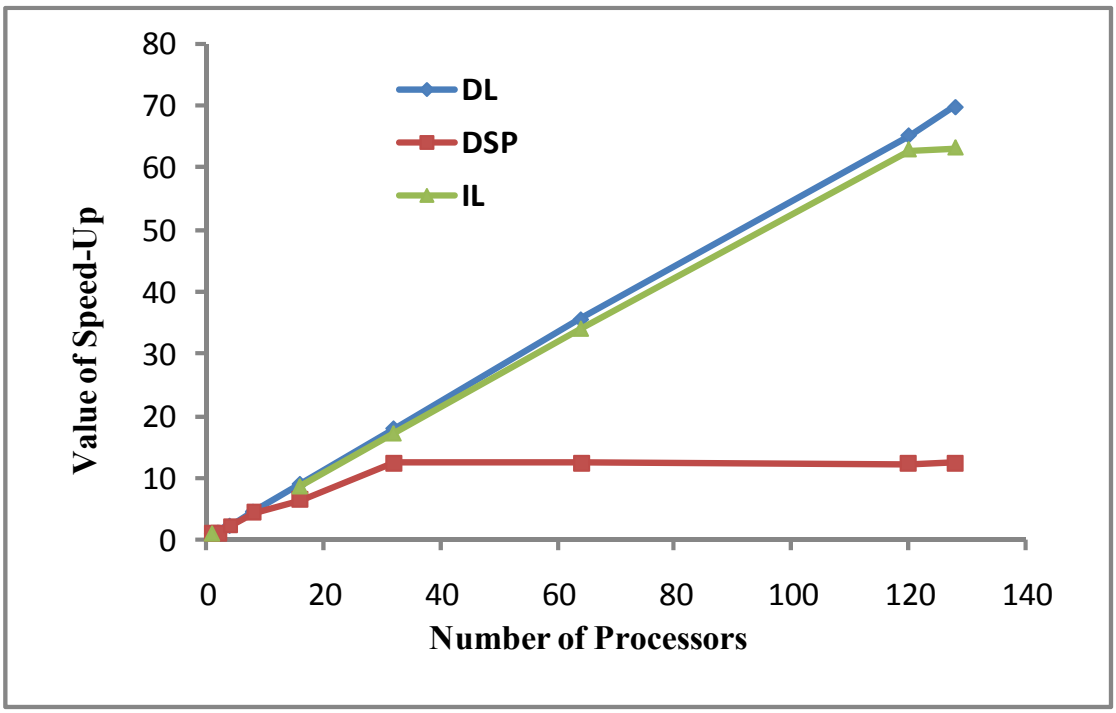

Figure 5.1 Trend of Speed-Up for the Three Distributed Computing Approaches

The explanations and analysis of these five phenomena are then discussed. Phenomenon (a) ensures the validity of three distributed computing approaches, and it implies that these approaches have sound parallelism, e.g., time elapsed in data communication is much less than time used in executing the calculation and the overall idle time is quite trivial.

As to phenomenon (b) and (c), at most 24 processors can be taken in use for DSP approach, thus when the number of processors exceeds 24, CPU time in conjunction with Speed-Up of DSP become quite stable. While, DL and IL approach can further improve

the acceleration with a linearly increasing value of Speed-Up. It can be speculated that if the number of processors reaches the sample size of Monte Carlo simulation, which is 8050, value of Speed-Up for DL would come to an upper bound. Yet, in this case, the IL approach can, in nature, further improve the calculation.

Regarding the execution time of the three approaches, a more important issue should be addressed here is that the performance of DL is always superior to that of IL, albeit they both have a quite high concurrency level. The reasons for this superiority of DL focus on two aspects: firstly, the IL approach needs to define more variables than DL, in that IL requires the grouping of processors in the first place. In addition, during the computation, it is also requires more time to distinguish each group; secondly, a more important reason attributes to the fact that DL has a better job-assignment mechanism, which inherently limited the total idle time. Note that for the IL approach, each 8 processors are grouped together to calculate the shortest path problems for 24 origins in each trial of the Monte Carlo simulation. It implies that for one particular processor, it will keep solving the shortest path problems for the same origins. Computational time of different origins should be slightly different, and this slight difference thus is calculated in each trial, which would be quite considerable after the 8050 simulations. This un-even workload causes idle time to the earlier finished processors. While, for the DL approach, one processor would calculate all the 24 shortest-path problems in each trial, thus the workload for different processors is identical, which gives rise to a perfect job-assignment plan and thence less idle time. It should be noted that the level of superiority of DL is unpredicted and it may be sensitive to the network structure.

Regarding phenomena (d) and (e), the value of Efficiency, $E(k)$, amounts to the slope of a straight line connecting the origin point $(0,0)$ and the value of Speed-Up $S(k)$. Thus since trend of Speed-Up is, in general, a straight line for DL and IL, their values of efficiency should keep still. In terms of Efficiency and Relative Burden, performance of DSP is getting worse when the number of processors is greater than 24. This is because at most 24 processors can be taken in use for DSP, and the redundant processors would remain in idle, yet they also occupied the resources in inter-node communication or data broadcasting.

It should be noted that increase of processors used will also prolong the total communication cost, and the value of Speed-Up shown in Figure 5.1 can properly estimate the trade-off between the communication cost and the computational efforts caused by additional processor added to the calculation. If the Marginal Burden caused by additional communication cost is larger than the Marginal Benefit that one processor can contributed to the speedup of calculation, then no more processor should be added to the computation. Due to the large communication capacity of the distributed computing system, the Marginal Burden for each calculation is quite marginal. Recall that the concurrency level for DL, DSP and IL approach is 8050, 24 and 193200 for the second stage of Monte Carlo simulation. Thus, for DL and IL approaches, all the processors involved in each test can contribute to the Marginal Benefit of the computation, thus their values of Speed-Up keep increasing. Yet, for the DSP approach, when the number of processors is larger than 24 (from 32 to 128), its value of Marginal Benefit is exactly zero,

and in such case even though the Marginal Burden is low the performance of DSP continuous to deteriorate.

In conclusion, the three CA incorporating the three distributed computing approaches can all evidently accelerate the calculation of probit-based SUE problem. The DL approach is demonstrated to be the most efficient one, and it should be recommended for solving the PA-SUEED problem on large-scale network example. Before taking DL approach for the calculation of PA-SUEED on large-scale network, another test is conducted for the stability of the HPC cluster. In this test, the calculation of one test is first conducted for multiple times and then the values of CPU time are recorded. Note that the conclusions made above would be rather suspicious, if an erratic oscillation is observed for the CPU time. Without loss of generality, two calculations of DSP with the number of processors equal to 2 and 8, respectively, are selected for this test, and for the sake of presentation they are referred to as Calculation 1 and Calculation 2. Calculation 1 is examined on the same group of processors, while Calculation 2 is tested on different processors each time. Each calculation is performed for four times, and the CPU times used by these tests are tabulated in Table 5.4.

Table 5.4 Test for the Stability of Calculations in the HPC Cluster

Sequence of tests	**CPU time (Seconds)**	
	Calculation 1	Calculation 2
1	2915.699	761.178
2	2915.651	760.648
3	2915.628	760.435
4	2915.444	763.440

Evidently, CPU times used for the tests of these two calculations are quite steady, which validates the conclusions made above.

5.4.2 Random Graph Example

As claimed in Section 5.1, the DL and IL considerably outperform the DSP approach. In addition, DL is slightly superior to IL due to its better job-assignment mechanism. Nevertheless, the level of this superiority is obscure, and it might be caused by some particularities of the Sioux-Falls network, thus a randomly generated network that is generic to any network structure would be more suitable for the numerical test.

Demeulemeester et al. (1993) has proposed two well-known random graph generators, named as Deletion Method and Addition Method. These two generators, coupled with some other existing random graph generators, are not available for the transportation network examples in this study. This is because the transportation network examples are required to be strongly connected by directed links, namely, there exists at least one path connecting any two nodes on the network. In addition, some links on the network should be accompanied by their opposite links, to guarantee the asymmetric property of link travel times. Let $|N|$ and $|A|$ denote the number of nodes and links on the network, where operator $|\bullet|$ gives the cardinality of any set. When assuming $|A| \geq 2|N| - 2$, a random graph generator, shown as follow, can be designed based on the theory of minimum spanning tree.

Step 0: (Initialization) Define an area with length X and width Y for the network example. Let a link set $\bar{A}$ be empty.

Step 1: (Random node location) For each node $n \in N$, randomly generate its coordinates in the defined area, where $x^{(n)} \in [0, X]$ and $y^{(n)} \in [0, Y]$. This gives the location of each node.

Step 2: (Random Minimum Spinning Tree) Randomly build a minimum spinning tree that connects all the $|N|$ nodes yielded in Step 1 (see the Addition Method in Demeulemeester et al. (1993) for an algorithm of the random minimum spinning tree). Add all the links on the minimum spinning tree to the link set $\bar{A}$.

Step 3: (Stop test) If $|\bar{A}| < |A|$, then go to Step 4, otherwise go to Step 5.

Step 4: (Additional links) Randomly select two nodes n_i and n_j in set N, and build a link that connects n_i to n_j. If this newly built link is not in link set $\bar{A}$, then add it to $\bar{A}$ and go to Step 3; otherwise, redo Step 4.

Step 5: (Network attributes) For each link $a \in \bar{A}$, let n_i and n_j denote the two ends of this link, then the Euclidean distance between n_i and n_j is defined as the free-flow travel time on this link. Capacity for link $a \in \bar{A}$ is randomly attained from the interval $[h'_a, h''_a]$, where h' and h'' are predetermined to be the lower bound and upper bound of the capacity, respectively.

A random graph then can be build following these procedures, and the random OD demand can also be easily obtained based on network. A network example is then

generated, with 100 nodes, 300 links, 16 zones (origins) and 16 OD pairs. Due to the space limit, detailed data for the network attributes are not listed here. This example is subsequently used to test the CA method incorporating DL and IL approaches, respectively. Different scenarios with varying number of processors are tested, and the execution times are tabulated in Table 5.5. It can be seen that in this test the superiority of DL approach is more evident. The randomly generated network can be a good representative of various network structures, and it thus further supports the DL approach, which should be selected for the computation of PA-SUEED problem upon large-scale networks.

Table 5.5 Comparison of DL and IL for the Random Graph Example

No. of processors	CPU time of DL (seconds)	CPU time of IL (seconds)
16	633.773	709.814
32	319.865	361.228
64	163.205	183.824
120	87.824	101.406
128	87.097	93.904

5.4.3 Anaheim Network

As discussed above, the CA method incorporating DL approach is utilized to calculate the PA-SUEED problem on a large-scale network example, called Anaheim network. This network has 416 nodes, 914 links and 1406 OD pairs, which is much larger than Sioux-Falls network. CPU time used by CA method for solving the PA-SUEED problem on this network is indicated by Table 5.6, where different number of processors is adopted the DL approach.

Table 5.6 Test of DL Approach for Solving Anaheim Network Example

No. of processors	CPU time (Hours)	Speed-Up	Efficiency	Relative Burden
1	77.403	1.000	1.000	0.000
2	70.710	1.095	0.547	0.414
4	35.380	2.188	0.547	0.207
8	17.932	4.317	0.540	0.107
16	8.952	8.647	0.540	0.053
32	4.560	16.976	0.530	0.028
64	2.321	33.353	0.521	0.014
128	1.209	64.015	0.500	0.008

As shown in Table 5.6, when only one processor is used, CPU time is as large as 77.4 hours. This execution time is beyond the acceptance, while it can be reduced to 1.2 hours when 128 processors are used. This improvement is highly desirable and of utmost importance to the practical implementations of the algorithm proposed for solving the PA-SUEED problem. The variations of the three performance measures for Anaheim network are quite similar to those for Sioux-Falls network: as the number of processors increases, Speed-Up is linearly increasing; Efficiency is, in general, unchangeable, and Relative Burden goes down to a trivial value. These phenomena therefore support the validity of DL distributed computing approaches for accelerating the convergent speed of CA method.

5.5 Conclusions

This chapter introduced three distributed computing approaches that can largely reduce the execution time of solution algorithms for PA-SUEED problem. A two-stage Monte Carlo simulation method was adopted for solving the stochastic network loading of PA-SUEED problem. The three distributed computing approaches thus all focus on decomposing the workload of Monte Carlo simulation and then conducting the

calculation concurrently. Computational results of the Sioux-Falls network example and a randomly generated network indicated that all these three approaches can inherently accelerate the convergent speed of the solution algorithm, compared with the sequential case. Additionally, when the number of processors employed was larger than the concurrency level of each approach, an evident deterioration can be observed. Based on the numerical tests, the Cost Averaging method incorporating Distributed Loading approach was discovered to be the most efficient solution algorithm for solving the PA-SUEED problem, and it was then employed to solve a large-scale network example, which also showed a sound efficiency.

Regarding its implicit values for future work, this chapter is in general research-oriented. For long, the probit-based SUE problem has been well-recognized as a perfect representative of the SUE principle, yet the computational cost and inaccuracy caused by solving its stochastic network loading have largely prohibits the further investigations about probit-based SUE problem. With the reduced travel time, it becomes possible to analyze some important research topics based on a network with probit-based SUE constraints, for instance, the capacity constrained probit-based SUE problem, road pricing toll design and transportation network design problem, etc. Furthermore, it is also of considerable interest and un-negligible significance to devote some efforts into the implementation issues, including more advanced skills for data communication and generation of random numbers.

As discussed in the Remark in Section 5.2, computational speed of each distributed computing approaches can be further accelerated by virtue of more sophisticated sampling skills, e.g., stratified sampling or importance sampling. Investigations of their usage in the context of PA-SUEED problem are also worthwhile for future research.

The computing platform of tests in current study is described in Section 5.3.1. It would not be burdensome to test the distributed computing approaches on any other computing platform addressed in relevant studies, for example, the platform introduced in Tremblay and Florian (2001), Wong et al. (2001), among many others. Thanks to the development of computer hardware, many personal PC or workstations now have multiple or even tens of processors. Thus, the computation of probit-based SUE can be conducted in parallel in such a shared-memory machine. In view of the prefect parallelism of the DL and IL approach, it would be more straightforward to be tested on any commercial cloud computing system. In such case, the IL may outperform DL since a large number of processors are available and IL has a larger concurrency level.

CHAPTER 6 SPEED-BASED TOLL DESIGN FOR CORDON-BASED CONGESTION PRICING SCHEME

The first part of this dissertation, Chapters 3 to 5, focuses on the mathematical formulations and solution algorithms for the PA-SUEED problem, which is a sound extension of the probit-based SUE traffic assignment problem. The PA-SUEED problem possesses a better fitness to the practical conditions, and such a framework is generic to (a) fixed and elastic demand cases; (b) separable, symmetric or asymmetric link travel time functions; (c) probit-based or logit-based SUE principles (in terms of formulation). Moreover, the efficient PC-CA algorithms as well as distributed computing approaches proposed in Chapters 3 and 5 have significantly reduced the computational burden of solving PA-SUEED problem, making it available for practical implementations.

Traffic congestion is undoubtedly the most focused problem in urban transportation networks. A superior equilibrium framework, like PA-SUEED, could help to predict the traffic congestions, but urban traffic demand management requires to further influence the travel behavior of drivers and to encourage them to make route choices wisely to inhibit congestions. Congestion pricing is an effective tool in terms of adjusting drivers' route choices. Thereby, congestion pricing has attracted numerous attentions from the researchers. As mentioned in Section 2.3.1, existing studies for congestion pricing mainly focus on the first-best and second-best pricing schemes. The first-best pricing is usually realized by the marginal cost pricing. Nevertheless, despite a better theoretical property, the marginal cost pricing has inferior practicality in that it requires charging facilities on every link, thus only second-best pricing schemes is implemented in reality. Regarding as

a special case of second-best pricing, cordon-based congestion pricing scheme is widely adopted in practice. The second part of this dissertation, Chapters 6 and 7, thus works on two practical-oriented pricing tropics, speed-based and distance-based, for cordon-based congestion pricing. Note that the speed-based toll design is originated from the Electronic Road Pricing (ERP) system in Singapore, while the distance-based toll design is taken as the future generation of ERP system. Thus, these two topics are both of considerable practical significance.

This chapter discusses about the speed-based toll design problem, where drivers' travel behavior is assumed to follow the PA-SUEED. It is further assumed that the drivers' value-of-time (VOT) is continuously distributed. Thus, the formulation and algorithm for PA-SUEED with continuously distributed VOT are first addressed. Then, the speed-based toll design problem is formulated as a mathematical programming with equilibrium constraints (MPEC) model, with the objective of maximizing total social benefit (TSB). A Distributed Revised Genetic Algorithm (DRGA) is finally proposed for solving the speed-based toll design problem, which is numerically validated by a network example.

6.1 Background and Relevant Studies

The toll design problem of congestion pricing schemes refers to determination of optimal toll charges according to one or more objectives, based on given charging locations. Some system-wide objectives are usually adopted, for instance, the total social benefit, total travel time and toll revenue. However, compared with these system-wide objectives, the government and network authorities have more concerns about the traffic conditions in the Central Business District (CBD), which is the commercial heart of each city and

traffic congestions in the CBD would cause greater economic losses and worse impacts to the city images. Thus, regarding the practical implementations of congestion pricing schemes, to mitigate the traffic congestions in CBD area is usually taken as a primary target.

Average travel speed is an ideal measure for the traffic conditions within the pricing cordon, in that it is much easier to be observed than traffic column and density (Li, 2002) and moreover travel speed is a better representative of the commuters' driving experience. In view of the cordon-based ERP system in Singapore, its objective is determined as keeping the average speed of vehicles in cordon area within a targeted range, [20, 30] km/hour. The toll charges are adjusted to maintain the average speed in such a range (Olszewski and Xie, 2005). The lower-bound of this range guarantees an expedite traffic condition, while the upper-bounds avoids a waste of the road resources. Herein, to search for a toll charge pattern that can maintain average travel speed of vehicles in the cordon area within a predetermined targeted range is named as speed-based toll design problem. Despite its practical significance, theoretical study for the speed-based toll design problem is still an open question, since few of the existing researches for toll design problems have taken the traffic conditions in CBD as an objective or using travel speed as a criterion for the network performance.

When analyzing the congestion pricing problems, a value-of-time (VOT) is necessitated to convert the toll charges into time unit. The VOT is inherently influenced by the wage rate as well as trip emergency of each commuter/driver (Li, 2002), which is quite diverse

amidst the commuters. As claimed in literature review at Section 2.3.3, it is thus more rational to take VOT as a continuously distributed random variable. The continuous distributed VOT results in another random term in the generalized path travel time, besides the perception error. A computational model is necessitated to describe the PA-SUEED with the continuously distributed VOT due to the inapplicability of the existing models developed for the SUE problems. As mentioned in Section 2.3.3, this chapter aims to propose a link-based solution technique for the PA-SUEED with the continuously distributed VOT, which avoids path generation/enumeration.

A toll-charge pattern that can maintain average travel speed in the cordon area within a predetermined targeted range is regarded as "acceptable". However, for a particular transportation network, there would be more than one acceptable toll-charge patterns. Thus, the optimum among these acceptable patterns that can give rise to a higher total social benefit (TSB) value is regarded as the solution of speed-based toll design problem. A MPEC model is first proposed for the speed-based toll design problem, wherein the equilibrium constraints representing a probit-based SUE problem with continuously distributed VOT. Note that due to the existence of continuously distributed VOT, existing algorithms for solving MPEC model is not available. Thus, a Genetic Algorithm type method is taken as a heuristic for solving the speed-based toll design problem.

Note that the adjustment for toll charges of the ERP system in Singapore is performed in this way (Olszewski and Xie, 2005): a review is launched every three months for the average travel speed in each cordon area, and then toll charges on all the entries to each

cordon are increased (deducted) by a certain amount, if average travel speed is less than lower-bound (greater than upper-bound) of its targeted range. This trial-and-error type toll adjustment approach is quite convenient for practical use, and it is thus combined to the solution algorithm, named as Revised Genetic Algorithm.

Most computational efforts for the Revised Genetic Algorithm are spent on the evaluation of every newly generated chromosome. Yet, due to the complete independency of each evaluation process, it is inspiring to accelerate the Revised Genetic Algorithm by means of distributed computing, and the whole computational procedure is thus called as Distributed Revised Genetic Algorithm (DRGA).

Remainder sections of this chapter are organized as follows. Section 6.2 presents a MPEC model for the speed-based toll design problem. Section 6.3 discusses the DRGA for solving the speed-based toll design problem, which is numerically tested by a network example in Section 6.4. This chapter is then concluded in Section 6.5.

6.2 Problem Statement and MPEC Model for Speed-Based Toll Design

6.2.1 Notation and Definitions

Based on the strongly connected transportation network $G=(N,A)$ defined at Chapter 3, it is assumed that there are several pricing cordons on the network, toll charges are implemented on each entry of the pricing cordons. Let $\bar{A}$ be the set of all the charged links, thus $\bar{A}\subseteq A$. Toll fare on each link $a\in\bar{A}$ is denoted by τ_a, and for the sake of presentation, all the toll fares are grouped into column vector $\boldsymbol{\tau}=\left(\tau_a, a\in\bar{A}\right)^{\mathrm{T}}$. Assuming

that total number of the pricing cordons is I, and each cordon is sequentially numbered by an integer from 1 to I. Any toll fare pattern is regarded as "acceptable" if the average travel speed of vehicles in each pricing cordon can retain in a targeted range. Namely, for the i^{th} pricing cordon (called as cordon i hereafter), $1 \le i \le I$, if the average speed of all the vehicles in this cordon during the morning peak hours is denoted by γ_i, we have that $\underline{\gamma}_i \le \gamma_i \le \overline{\gamma}_i$, where constant $\underline{\gamma}_i$ ($\overline{\gamma}_i$) is a predetermined lower (upper) bound of the targeted range for average speed γ_i.

γ_i would be inherently influenced by the toll charges $\boldsymbol{\tau} = \left(\tau_a, a \in \bar{A}\right)^{\mathrm{T}}$ that affect commuters' route choice plans and eventually change the flows and travel speed on each link in the cordon area. For any variation on the toll fares, the commuters would adjust their route choice plans, and the link flows are assumed to converge to a new equilibrium after a short span of time. Thus, when let $v_a(\boldsymbol{\tau})$ denote the equilibrium flow on link $a \in A$, it should be a function of the toll-charge pattern $\boldsymbol{\tau}$. All the link flows are grouped into a column vector $\mathbf{v}(\boldsymbol{\tau}) = \left(v_a(\boldsymbol{\tau}), a \in A\right)^{\mathrm{T}}$. Likewise, some other attributes on the network should also be functions of the toll-charge pattern $\boldsymbol{\tau}$, including OD demand $\mathbf{q}(\boldsymbol{\tau})$ and path flows $\mathbf{f}(\boldsymbol{\tau})$. In this case, the flow conservation conditions should also be fulfilled.

The cumulative toll charges, denoted by τ_{wk}, on path $k \in R_w$ can be calculated by:

$$\tau_{wk} = \sum_{a \in \bar{A}} \tau_a \delta_{ak}^w . \tag{6.1}$$

To analyze the impacts of these toll fares on commuters' route choice behaviors, τ_{wk} should be converted into time-units by using the commuters' value-of-time (VOT). As aforementioned, the VOT, denoted by α, is regarded as a continuously distributed random variable across the whole population. α is assumed to have constant mean and variance, and its probability density function (PDF) is continuously differentiable. The cumulative distribution function (CDF) of α can be obtained by the empirical curve plotting method if the VOT values of all the drivers are known. In the literature, α is usually assumed to follow uniform distribution, exponential distribution, or log-normal distribution.

The commuters make their per-trip route plans based on their perceived value of the costs on each path, and this perceived cost on path $k \in R_w$ should be expressed as:

$$\bar{C}_{wk}(\mathbf{v}, \boldsymbol{\tau}) = c_{wk}(\mathbf{v}) + \zeta_{wk} + \frac{\tau_{wk}}{\alpha}, \tag{6.2}$$

where ζ_{wk} denotes the commuters' perception error on path travel time, which is a random variable with zero mean and constant variance. ζ_{wk} is assumed to be normally distributed in the context of PA-SUEED. Note that due to the existence of continuously distributed VOT, the models and algorithms proposed in Chapters 6 and 7 are also suitable to DUE or logit-based SUE cases.

As shown by eqn. (6.2), the continuously distributed VOT brought another random term $\frac{\tau_{wk}}{\alpha}$ to the cost. Let $E\left(\frac{\tau_{wk}}{\alpha}\right)$ be the mean of $\frac{\tau_{wk}}{\alpha}$, and for the sake of presentation, eqn. (6.2) is rewritten as:

$$\bar{C}_{wk}\left(\mathbf{v},\boldsymbol{\tau}\right)=c_{wk}\left(\mathbf{v}\right)+E\left(\frac{\tau_{wk}}{\alpha}\right)+\lambda_{wk}, \tag{6.3}$$

where λ_{wk} is termed as generalized perception error, which has the expression:

$$\lambda_{wk}=\frac{\tau_{wk}}{\alpha}-E\left(\frac{\tau_{wk}}{\alpha}\right)+\zeta_{wk}. \tag{6.4}$$

Evidently, the PDF of λ_{wk} is continuously differentiable, and λ_{wk} also has zero mean and constant variance. The generalized perception error on all the paths associated with OD pair $w\in W$ are grouped into column vector $\boldsymbol{\lambda}_w=\left(\lambda_{wk},k\in R_w\right)^{\mathrm{T}}$. $C_{wk}\left(\mathbf{v},\boldsymbol{\tau}\right)$ in eqn. (6.3) is then referred to as generalized path travel time function. Analysis of the equilibrium network flows based on the generalized path travel time function is in nature a PA-SUEED problem with continuously distributed VOT.

Let $S_w\left(\mathbf{v},\boldsymbol{\tau}\right)$ denote the satisfaction function (Sheffi, 1985) in such case, namely:

$$S_w\left(\mathbf{v},\boldsymbol{\tau}\right)=E\left[\min_{k\in R_w}\left\{\bar{C}_{wk}\left(\mathbf{v},\boldsymbol{\tau}\right)\right\}\right]. \tag{6.5}$$

$S_w\left(\mathbf{v},\boldsymbol{\tau}\right)$ here can be used to measure the impedance of traveling between OD pair $w\in W$. Thus, travel demand between any OD pair $w\in W$ is also assumed to be a non-increasing function of $S_w\left(\mathbf{v},\boldsymbol{\tau}\right)$, with the expression:

$$q_w=D_w\left(S_w\left(\mathbf{v},\boldsymbol{\tau}\right)\right)\le\bar{q}_w, w\in W. \tag{6.6}$$

6.2.2 MPEC Model for the Speed-Based Toll Design Problem

With a predetermined targeted range for the average speed of vehicles in each pricing cordon, $\underline{\gamma}_i \le \gamma_i \le \overline{\gamma}_i$, there would be more than one "acceptable" toll fare pattern that generates desirable average travel speed in all the cordons. While, the toll design problem, as shown in Section 6.1, aims to search for the "optimal" toll fare pattern, among all the acceptable solutions, that gives rise to maximal value of total social benefit (TSB). An expression of TSB in the framework of PA-SUEED is thus of intrinsic interests. Assuming constant VOT, a widely used expression for TSB of a transportation network in terms of DUE with elastic demand comes from the Marshallian rule (see, e.g. Maruyama and Sumalee, 2007):

$$\int_0^{q_w} D_w^{-1}(w)dw - \sum_{a\in A} v_a\left(t_a + \frac{\tau_a}{\alpha}\right) + \sum_{a\in A} v_a \frac{\tau_a}{\alpha} = \int_0^{q_w} D_w^{-1}(w)dw - \sum_{a\in A} v_a t_a , \qquad (6.7)$$

where τ_a denotes the toll charge on link $a \in A$. On the left hand side of eqn. (6.7): the first term is the overall benefit gained by the drivers from their journey, the second term represents the overall cost borne by them, and the last term is the total toll revenue, which would be re-invested in the construction and maintenance of infrastructures in transportation network. Note that the first proportion is constant if travel demand is fixed, thus eliminating the first proportion of eqn. (6.7) an index for TSB of DUE with fixed demand can be obtained, which is the Wardrop's second principle, System Optimum.

While, for a pricing scheme in terms of SUE with fixed demand, Bellei et al. (2002) defined TSB as follows:

$$-\sum_{w\in W} q_w S_w\left(\mathbf{c}_w(\mathbf{v}),\boldsymbol{\tau}\right)+\sum_{a\in A} v_a \frac{\tau_a}{\alpha}. \tag{6.8}$$

Based on eqns. (6.7) and (6.8), an expression for the TSB of cordon-based congesting pricing scheme in terms of PA-SUEED principle can be given by:

$$Z_1(\boldsymbol{\tau})=\sum_{w\in W}\int_0^{q_w(\boldsymbol{\tau})} D_w^{-1}(x)dx-\sum_{w\in W} q_w(\boldsymbol{\tau}) S_w(\mathbf{v},\boldsymbol{\tau})+\sum_{a\in \bar{A}} E\left(v_a(\boldsymbol{\tau})\frac{\tau_a}{\alpha}\right), \tag{6.9}$$

where the first term in right-hand-side of eqn. (6.9) is the overall benefits obtained by the commuters from their trips, and the second proportion represents the overall travel costs borne by the commuters. The last proportion is the total toll revenue, and it should be noted that in the last term the mean value is used due to the existence of randomly distributed VOT, α.

Based on eqn. (6.9), the speed-based toll design problem can then be formulated by the following MPEC model:

$$\max_{\boldsymbol{\tau}\in\Omega_\tau} Z_1(\boldsymbol{\tau}) \tag{6.10}$$

Subject to

$$\underline{\gamma}_i \le \gamma_i(\boldsymbol{\tau}) \le \overline{\gamma}_i, i=1,2,\ldots,I \tag{6.11}$$

$$\mathbf{v}(\boldsymbol{\tau}),\ \mathbf{f}(\boldsymbol{\tau}) \text{ and } \mathbf{q}(\boldsymbol{\tau}) \text{ fulfil the equilibrium conditions} \tag{6.12}$$

In eqn. (6.10), Ω_τ denotes the feasible set for toll fare patterns, i.e., $\Omega_\tau=\left\{\boldsymbol{\tau} \mid \underline{\tau}\le\tau_a\le\overline{\tau}, a\in\bar{A}\right\}$, where $\underline{\tau}$ ($\overline{\tau}$) is a predetermined lower (upper) bound for the toll charges. In eqn. (6.12), $\mathbf{v}(\boldsymbol{\tau})$, $\mathbf{f}(\boldsymbol{\tau})$ and $\mathbf{q}(\boldsymbol{\tau})$ denote the vector for link flows, path flows and OD demands, which can be attained by solving a PA-SUEED problem with

continuously distributed VOT. Mathematical model and solution algorithm for the PA-SUEED problem with continuously distributed VOT are briefly introduced in the following sub-section.

6.2.3 PA-SUEED Problem with Continuously Distributed VOT

For any given toll charge pattern $\boldsymbol{\tau} \in \Omega_{\tau}$, its corresponding equilibrium link flows $\mathbf{v}(\boldsymbol{\tau}) = \left(v_a(\boldsymbol{\tau}), a \in A\right)^{\mathrm{T}}$ is in nature the solution of a PA-SUEED problem with continuously distributed VOT.

It is straightforward to show that the probability density function (PDF) of the generalized perception error λ_{wk} is continuously differentiable and strictly positive. Meanwhile, following the proof of Lemma 4.1 in Daganzo (1979), it gives that the conditions shown by Proposition 6.1 is still effective when the continuously distributed VOT is assumed.

Proposition 6.1: Satisfaction function $S_w(\mathbf{v}, \boldsymbol{\tau})$ is concave with respect to path travel times $\mathbf{c}_w(\mathbf{v})$, and moreover $\dfrac{\partial\left(S_w(\mathbf{v}, \boldsymbol{\tau})\right)}{\partial c_{wk}(\mathbf{v})} = p_{wk}(\mathbf{v}, \boldsymbol{\tau})$, thus

$$S_w(\mathbf{v}', \boldsymbol{\tau}) - S_w(\mathbf{v}'', \boldsymbol{\tau}) \le \mathbf{p}_w(\mathbf{v}'', \boldsymbol{\tau})^{\mathrm{T}}\left(\mathbf{c}_w(\mathbf{v}') - \mathbf{c}_w(\mathbf{v}'')\right), \forall \mathbf{v}', \mathbf{v}'' \in \Omega_{\mathbf{v}}, \tag{6.13}$$

where $\mathbf{p}_w(\mathbf{v}'', \boldsymbol{\tau}) = \left(p_{wk}(\mathbf{v}'', \boldsymbol{\tau}), k \in R_w\right)^{\mathrm{T}}$ is a column vector of choice probability of all the paths associated with OD pair $w \in W$.

These properties guarantees that the path choice probability in the case of PA-SUEED with continuously distributed VOT is still *regular* (see Section 2.4 of Cantarella, 1997), thus the solution framework addressed in Chapter 3 is still effective for such a problem. Namely, it can be formulated as the following fixed-point model over the feasible set of link flows:

$$v_a(\boldsymbol{\tau}) = \sum_{w \in W} \sum_{k \in R_w} \left[D_w\left(S_w(\mathbf{v}, \boldsymbol{\tau})\right) \times p_{wk}(\mathbf{v}, \boldsymbol{\tau}) \delta_{ak}^w \right], a \in A. \tag{6.14}$$

The equilibrium conditions in eqn. (6.12) thus can be replaced by this fixed-point model (6.14). As per Theorems 1 and 2 of Cantarella (1997), this fixed-point model has one unique solution in terms of PA-SUEED with continuously distributed VOT. Thus, the solution algorithms discussed in Chapter 3 are all effective for solving this problem. Note that the procedures of the link-based two-stage Monte Carlo simulation method for solving SNL should be modified in this case, since the value of VOT should also be sample in each run. Yet, such a change is straightforward, thus the detailed procedures of the modified Monte Carlo simulation method are not provided here.

6.3 Solution Algorithm for the Speed-based Toll Design

In view that the proposed MPEC model is not convex and also considering the continuously distributed VOT, existing algorithms (see Section 2.3.1) are not available for solving this MPEC model. Note that a theoretically effectual method is to enumerate all the feasible toll patterns, assess their corresponding value of total social benefit (TSB) and average speed $\gamma_i(\boldsymbol{\tau})$, and then choose the optimum with maximal TSB value, among those toll patterns that can fulfill the desired speed interval, eqn. (6.11). This brute force method is extremely time-consuming, per se, and would be unacceptable even for a

middle size example. Consequently, in this chapter a genetic algorithm type method is adopted to solve the proposed model, which is an approximation of the brute force method.

GA is one of the most well-known search heuristics for solving optimization problems (e.g., Goldberg, 1989; Gen and Cheng, 1997). Chromosomes of the genetic algorithm are designed in this way: all the tolled links on the network are successively numbered, and each gene in one chromosome represents the toll charge on the corresponding tolled link. For the chromosomes in the initial generation, all their genes are randomly generated between $\underline{\tau}$ and $\overline{\tau}$.

To cope with the speed constraint (6.11), a penalty cost is added to the objective function, thus the model (6.10) is approximated by the following model:

$$\max_{\boldsymbol{\tau}\in\Omega} Z_2(\boldsymbol{\tau}) = Z_1(\boldsymbol{\tau}) - c\left(\sum_{i=1}^{I}\left(\max\left(0, \underline{\gamma}_i - \gamma_i\right)\right)^2\right) - c\left(\sum_{i=1}^{I}\left(\max\left(0, \gamma_i - \overline{\gamma}_i\right)\right)^2\right) \quad (6.15)$$

Subject to:

$$v_a(\boldsymbol{\tau}) = \sum_{w\in W}\sum_{k\in R_w}\left[D_w\left(S_w(\mathbf{v},\boldsymbol{\tau})\right)\times p_{wk}(\mathbf{v},\boldsymbol{\tau})\delta_{ak}^{w}\right], a\in A \quad (6.16)$$

where penalty parameter c is a large positive number.

6.3.1 Revised Genetic Algorithm

As shown in Section 6.1, the Land Transport Authority in Singapore adjusts the toll charges based on a regular survey on the travel speed. In accordance with this strategy for toll adjustment, for any chromosome in a particular generation of the genetic algorithm, if

its corresponding average speed is not in the targeted range, a similar adjustment would be conducted on the chromosomes. This toll adjustment strategy would subsequently produce a new chromosome. Together with those from the crossover and mutation processes, all the newly generated chromosomes will be considered for the selection of next generation. Such a solution algorithm for the speed-based toll design problem is called as Revised Genetic Algorithm, shown as follows:

Step 1: (Initial population). Set the size of population to be n. Randomly generate initial population of the chromosomes, which contains toll fares on each tolled link. Let number of generation $m = 1$.

Step 2: (Crossover). Randomly choose some parents from the survivors, and conduct pairing between each parent, which yields some new chromosomes.

Step 3: (Mutation). With a lower probability, randomly choose some genes from all the chromosomes in current generation, and then modify value of these genes by a pseudo random number between $\underline{\tau}$ and $\overline{\tau}$. This process also generates some new chromosomes.

Step 4: (Evaluation). For each newly generated chromosome, solve a corresponding PA-SUEED problem, and then record its corresponding TSB value in terms of the objective function (6.15).

Step 5: (Toll Adjustment). Based on the existing individuals in current generation, perform a one-off adjustment on their toll fares in turns: for a chromosome with the toll fares equal to $\boldsymbol{\tau} = \left(\tau_a, a \in \overline{A}\right)^{\mathrm{T}}$, check the corresponding average speed of vehicles in pricing cordon i, denoted by $\gamma_i(\boldsymbol{\tau})$, and if $\gamma_i(\boldsymbol{\tau})$ is less than the predetermined lower bound $\underline{\gamma}_i$, then increase toll fares on all the entry links to

this cordon by π; otherwise if $\gamma_i(\boldsymbol{\tau})$ is greater than its upper bound $\overline{\gamma}_i$, deduct the toll fares on its entry links by π. Here, the increment π is predetermined and fixed. This adjustment produces some new chromosomes, and then the TSB values of these new chromosomes are evaluated.

Step 6: (Selection). Among all the existing individuals, choose the top *n* individuals with larger TSB value, and then set these *n* individuals as survivors for next generation.

Step 7: (Stop Test). If a stop criterion is fulfilled, stop; otherwise, set $m = m + 1$ and go to Step 2.

6.3.2 Decomposition of Revised Genetic Algorithm for Distributed Computing

It can be seen from the procedures above for Revised Genetic Algorithm that evaluation process of each chromosome mainly requires solving a PA-SUEED problem with continuously distributed VOT based on given toll pattern $\boldsymbol{\tau} \in \Omega_\tau$.

In each generation of the Revised Genetic Algorithm, there would be tens of newly generated chromosomes. Most of the computational efforts are devoted onto the evaluation process. Evaluation of each chromosome is totally independent and follows identical procedure, thus it also has a perfect parallelism. Therefore, it is quite straightforward to inherently accelerate the computation of the Revised Genetic Algorithm by virtue of distributed computing, and such a computational procedure is named as Distributed Revised Genetic Algorithm (DRGA). Regarding the parallel (distributed) GA type methods used in transportation studies, Wong et al. (2001)

proposed a parallel GA for the calibration of Lowry model, and only recently Cipriani et al. (2010) has used a parallel GA on a personal computer with dual-core processor for solving the transit network design problem.

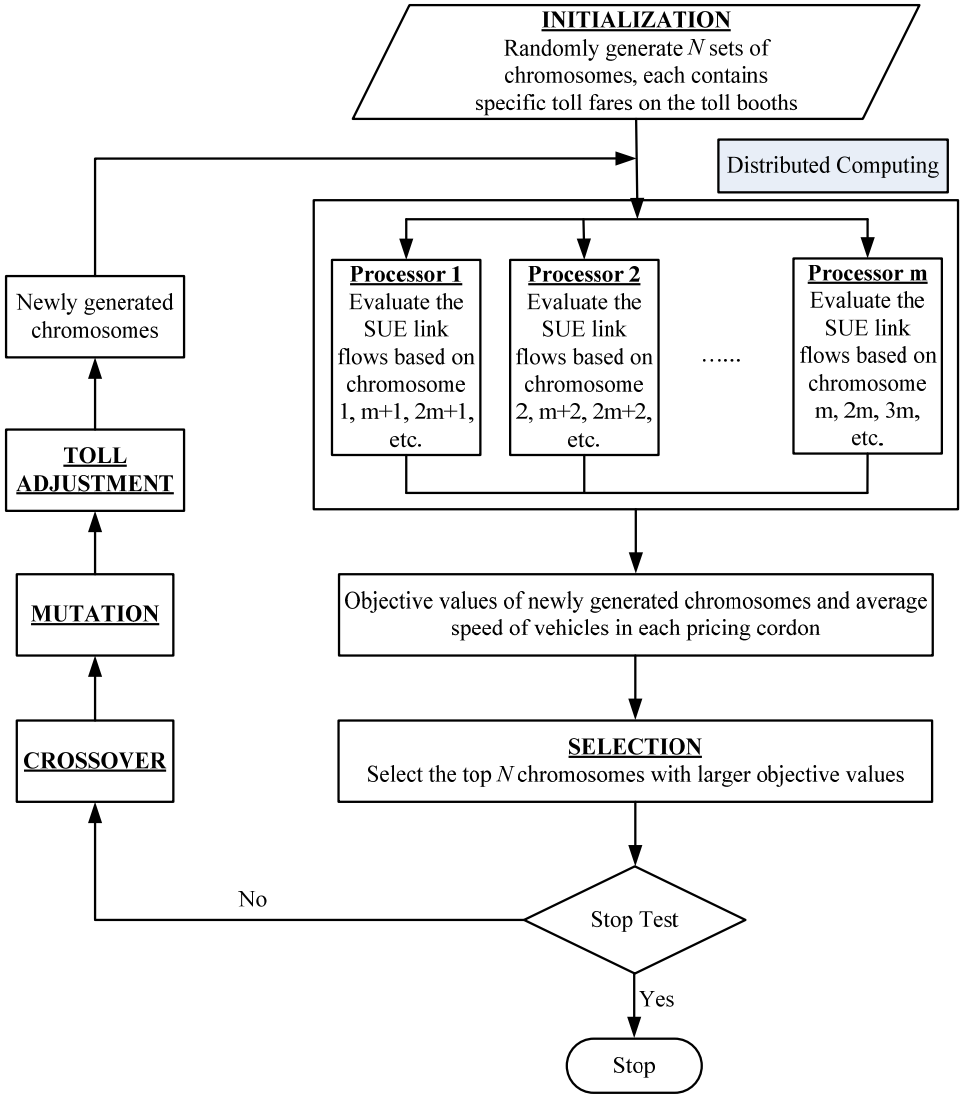

Figure 6.1 Flowchart of Distributed Revised Genetic Algorithm

Figure 6.1 shows the procedures of DRGA. It can be seen that in each iteration new chromosomes are yield by three processes: crossover, mutation and toll adjustment. Then, all the newly generated chromosomes are taken for evaluation, which possesses more

than 90% of the total CPU time. As mentioned earlier, the evaluation process for each chromosome is conducted by different processors in the distributed computing system synchronously. Suppose the total number of processors equal to k, and all the newly generated chromosomes are evenly assigned to these k processors. Then, all the processors would work in parallel, which largely reduced the total execution time. After the evaluation, computational results for each newly generated chromosome are sent to the main processor for selection and stop test.

6.4 Numerical Example

To numerically validate the proposed model and algorithm for the speed-based toll design problem, a network example is build based on the cordon-based Electronic Road Pricing (ERP) system in downtown Orchard Road of Singapore.

Figure 6.2 Locations of ERP System on the Orchard Road of Singapore

Figure 6.2 downloaded from the website of Singapore Land Transport Authority (2011) shows the charging locations on Orchard Road, and clearly each entry to the cordon is charging. Based on the map shown by Figure 6.2, a grid network example with 33 nodes and 104 links is built in Figure 6.3. The pricing cordon is highlighted by a dotted ellipse, and all the 12 entries are indicated by bold lines. All the entries are grouped into set $\overline{A}$, which are:

$$\overline{A} = \{24, 25, 27, 29, 34, 47, 79, 82, 84, 86, 88, 90\} \tag{6.17}$$

It is noted, in passing, that the entries shown in eqn. (6.17) are sequentially used to build the chromosomes in the DRGA. The targeted range for average speed of vehicles in Orchard Road cordon is decided by Singapore Land Transport Authority to be [20, 30] km/hour. In light of the speed-based toll design problem, toll charges on each entry should be adjusted to keep the average travel speed in this range and meanwhile achieve a better total social benefit. The increment π of the toll adjustment procedure (see Step 5 of the Revised Genetic Algorithm) is currently taken as 1.0 Singapore Dollar (S$).

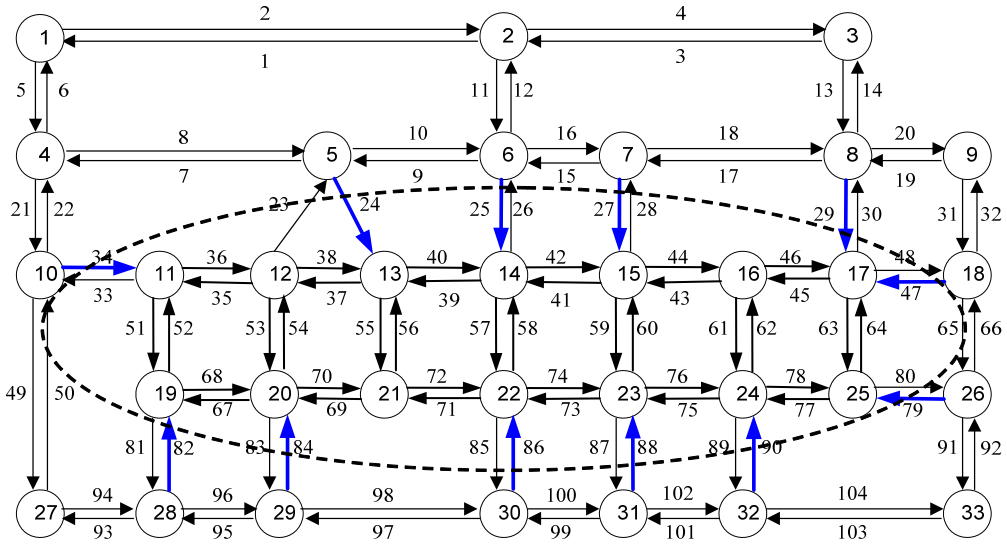

Figure 6.3 Topological Structure of the Orchard Road Network

It is assumed that 12 OD pairs exist on this network. Table 1 shows the origin and destination node of each OD pair as well as the upper bound of its travel demand. The actual travel demand between OD pair $w \in W$ is assumed to be determined by the eqn. (3.82) with the parameter μ equal to 0.001 for this example.

Table 6.1 Parameters in the Travel Demand Function

OD pair w	**Upper bound of travel demand** $\overline{q}_w$ **(vehicles/hours)**
$1 \to 33$	5000
$9 \to 1$	4000
$3 \to 27$	5000
$27 \to 9$	5000
$2 \to 29$	6000
$18 \to 28$	6000
$4 \to 24$	3000
$32 \to 14$	5000
$33 \to 3$	5000
$25 \to 4$	5000
$28 \to 6$	8000
$7 \to 23$	8000

The asymmetric link travel time function still follows eqn. (3.86). Data for the relevant link attributes are tabulated in Table 6.2.

Table 6.2 Data for the Link Attributes

Link No.	**Start node**	**End node**	**Free-flow travel time** t_a^0 **(Seconds)**	**Capacity** h_a **(vehicles /hours)**
1	2	1	60	5400
2	1	2	60	5400

Link No.	**Start node**	**End node**	**Free-flow travel time** t_a^0 **(Seconds)**	**Capacity** h_a **(vehicles /hours)**
4	2	3	100	5400
5	1	4	40	5400
6	4	1	40	5400
7	5	4	90	3600
8	4	5	90	3600
9	6	5	42	3600
10	5	6	42	3600
11	2	6	80	7200
12	6	2	80	7200
13	3	8	72	3600
14	8	3	72	3600
15	7	6	160	1800
16	6	7	160	1800
17	8	7	120	1800
18	7	8	120	1800
19	9	8	60	3600
20	8	9	60	3600
21	4	10	80	5400
22	10	4	80	5400
23	12	5	55	3600
24	5	13	55	3600
25	6	14	80	7200
26	14	6	80	7200
27	7	15	120	1800
28	15	7	120	1800
29	8	17	60	3600
30	17	8	60	3600
31	9	18	90	3600
32	18	9	90	3600
33	11	10	16	5400
34	10	11	16	5400
35	12	11	40	5400
36	11	12	40	5400
37	13	12	24	5400
38	12	13	24	5400
39	14	13	48	5400
40	13	14	48	5400
41	14	15	40	5400
42	15	14	40	5400
43	16	15	12	5400
44	15	16	12	5400
45	17	16	60	5400

Link No.	**Start node**	**End node**	**Free-flow travel time t_a^0 (Seconds)**	**Capacity h_a (vehicles /hours)**
47	18	17	60	5400
48	17	18	60	5400
49	10	27	20	5400
50	27	10	20	5400
51	11	19	24	1800
52	19	11	24	1800
53	12	20	20	5400
54	20	12	20	5400
55	13	21	30	3600
56	21	13	30	3600
57	14	22	12	7200
58	22	14	12	7200
59	23	15	12	3600
60	15	23	12	3600
61	16	24	12	3600
62	24	16	12	3600
63	17	25	14	3600
64	25	17	14	3600
65	18	26	20	5400
66	26	18	20	5400
67	20	19	80	1800
68	19	20	80	1800
69	20	21	20	5400
70	21	20	20	5400
71	22	21	48	5400
72	21	22	48	5400
73	23	22	20	5400
74	22	23	20	5400
75	24	23	28	5400
76	23	24	28	5400
77	25	24	60	5400
78	24	25	60	5400
79	26	25	60	5400
80	25	26	60	5400
81	19	28	24	1800
82	28	19	24	1800
83	20	29	12	5400
84	29	20	12	5400
85	22	30	18	7200
86	30	22	18	7200
87	23	31	16	3600
88	31	23	16	3600

Link No.	Start node	End node	Free-flow travel time t_a^0 (Seconds)	Capacity h_a (vehicles /hours)
90	32	24	18	5400
91	26	33	60	5400
92	33	26	60	5400
93	28	27	20	5400
94	27	28	20	5400
95	29	28	40	5400
96	28	29	40	5400
97	30	29	90	3600
98	29	30	90	3600
99	31	30	30	3600
100	30	31	30	3600
101	32	31	60	3600
102	31	32	60	3600
103	33	32	96	5400
104	32	33	96	5400

As mentioned earlier that commuters' VOT is assumed to be continuously distributed, it is assumed that the VOT is uniformly distributed, ranging from 18.0 to 72.0 S$ per hour.

6.4.1 Simulation Method for the Average Travel Speed in Each Cordon

As per the speed-based toll design method, toll charges on each entry of pricing cordon i should be adjusted in terms of the average journey speed of all the vehicles in this cordon during the decision period. Herein, the decision period is defined as one hour in the morning peak. In reality, after the implementation of any toll-charge pattern the average journey speed of vehicles in the cordon can be obtained by a survey using the probe vehicles. While, in this numerical example, the corresponding average speed γ_i of cordon i is approximated by an area-wide speed-flow model proposed by Olszeski et al. (1995) for the downtown area of Singapore:

$$Q_i = 80.645\gamma_i \left(44.9 - 12.0 \ln \gamma_i\right)^{1.563} - 2121.8, i = 1, 2, \ldots I \qquad (6.18)$$

where Q_i denotes the total in-bound and out-bound volume to cordon i, which equals to the summation of flows on all the entries and exits. With any given toll-charge pattern $\boldsymbol{\tau} = \left(\tau_a, a \in \overline{A}\right)^{\mathrm{T}}$, the equilibrium link flows on all the entries and exits of cordon i can be obtained by solving a PA-SUEED problem with continuously distributed VOT, and consequently its total in-bound and out-bound volume $Q_i\left(\boldsymbol{\tau}\right), i = 1, 2, \ldots, I$ equals to the flow on all the entries and exits of the pricing cordon. Taking $Q_i\left(\boldsymbol{\tau}\right), i = 1, 2, \ldots, I$ into eqn. (6.18) gives an estimation of the average travel speed $\gamma_i\left(\boldsymbol{\tau}\right)$, and it is then used to adjust the toll charges on each entry of cordon i.

6.4.2 Computational Results of Distributed Revised Genetic Algorithm

The Distributed Revised Genetic Algorithm (DRGA) is then used to solve the speed-based toll design problem. The computing platform of this distributed computing approach is identical to that described at Section 5.3, thus is not repeated here.

The DRGA is tested in different scenarios, wherein different numbers of processors in the distributed computing system are used. The execution time as well as the value of Speed-Up (Nagel and Rickert, 2001; Wong et al., 2001), are used to evaluate its performance in each scenario. Recall that the value of Speed-Up can be calculated as follows:

$$S\left(k\right) = \frac{T_1}{T_k} \tag{6.19}$$

where T_1 denotes the execution time used by only one processor, and T_k is the execution time when k processors are used.

For the DRGA, its values for population and generation are both decided to be 50. The computation will be terminated after 50 generations, which is taken as a stop criterion. Whilst, the mutation and crossover rates are set to be 0.01 and 0.25, respectively. Sample size of each stage of the two-stage Monte Carlo simulation is determined to be 100 and 1000 for this example. In addition, the penalty parameter c in eqn. (6.15) is set to be 1.0×10^{6}.

Upper-bound $\overline{\tau}$ and lower-bound $\underline{\tau}$ for the positive toll charges on each entry are taken as 10.0 S$ and 0.5 S$, respectively. Initial generation of the DRGA is then produced by independently setting a random number between [0.5, 10.0] to each gene of the chromosomes. It should be pointed out that due to the toll adjustment process (Step 5 in Section 6.3.1), toll charges may overstep the range of [0.5, 10.0] S$.

Table 6.3 Resultant Toll Charges on Each Entry

Entry No.	**24**	**25**	**27**	**29**	**34**	**47**
Optimal Toll Charge (S$)	1.1	0.5	2.1	1.0	3.1	5.0
Entry No.	**79**	**82**	**84**	**86**	**88**	**90**
Optimal Toll Charge (S$)	1.8	6.1	1.5	2.5	3.7	2.5

Table 6.3 indicates the resultant optimal toll charges on each entry to the Orchard Road Cordon, denoted by $\boldsymbol{\tau}^{*}$. The corresponding total social benefit (TSB) (in terms of eqn. (6.15)) and the average travel speed $\gamma_i\left(\boldsymbol{\tau}^{*}\right)$ in Orchard Road cordon are 4.79×10^{7} and

24.4 km/hour, respectively. It can be seen that $\gamma_i(\boldsymbol{\tau}^*)$ locates in the targeted range [20, 30] km/h decided by the Singapore Land Transport Authority.

Table 6.4 Execution Time and Speed-Up with Different Number of Processors

No. of Processors	Execution Time (Seconds)	Speed-Up
1	107580	1.00
2	56736	1.90
3	41330	2.60
4	37116	2.90
5	28836	3.73
6	27427	3.92
7	26458	4.07
8	22950	4.69
9	20399	5.27
10	19106	5.63
15	14501	7.42
20	11156	9.64
25	10080	10.67
30	9826	10.95

To comprehensively see the impact of toll charges on the network conditions, two additional tests are conducted for the cases with null toll (toll charges all equal to zero) and highest toll (the upper-bound 10.0 S$ is levied on each entry). It gives that for the un-tolled case: the TSB value is -5.17×10^7 and average speed in the cordon is 10.1 km/hour; for the case with highest toll: the TSB value is 2.56×10^7 and the average speed is 34.2 km/hour. Computational results of these two extreme cases verified that the cordon-based toll charges would inherently influence the network users' route choice behavior to mitigate traffic congestions within the cordon area. For the un-tolled case, an average speed of 10.1 km/hour implies a congested road condition, which is much worse than the expectation of the network authorities. While, in the case of highest toll charges,

a fast average speed of 34.2 km/hour implies that the quite few vehicles are traveling in the area, which is a waste of the road resources. The TSB value for the un-tolled case is negative due to its high penalty cost for the unacceptable average travel speed. Performance of the DRGA would be affected by the number of processors used. Hence, a sensitivity test is conducted for its impact on the total execution time as well as the value of Speed-Up, shown in Table 6.4. It can be seen that when only one processor is used for the calculation T_1, the execution time is as large as 107580 seconds, approximately 30 hours, which is beyond the acceptance. The execution time, however, is sharply shortened when more processors are used. In the optimal case, when 30 processors are used, the computation can be accelerated for nearly 11 times, with an execution time of around 2.7 hours.

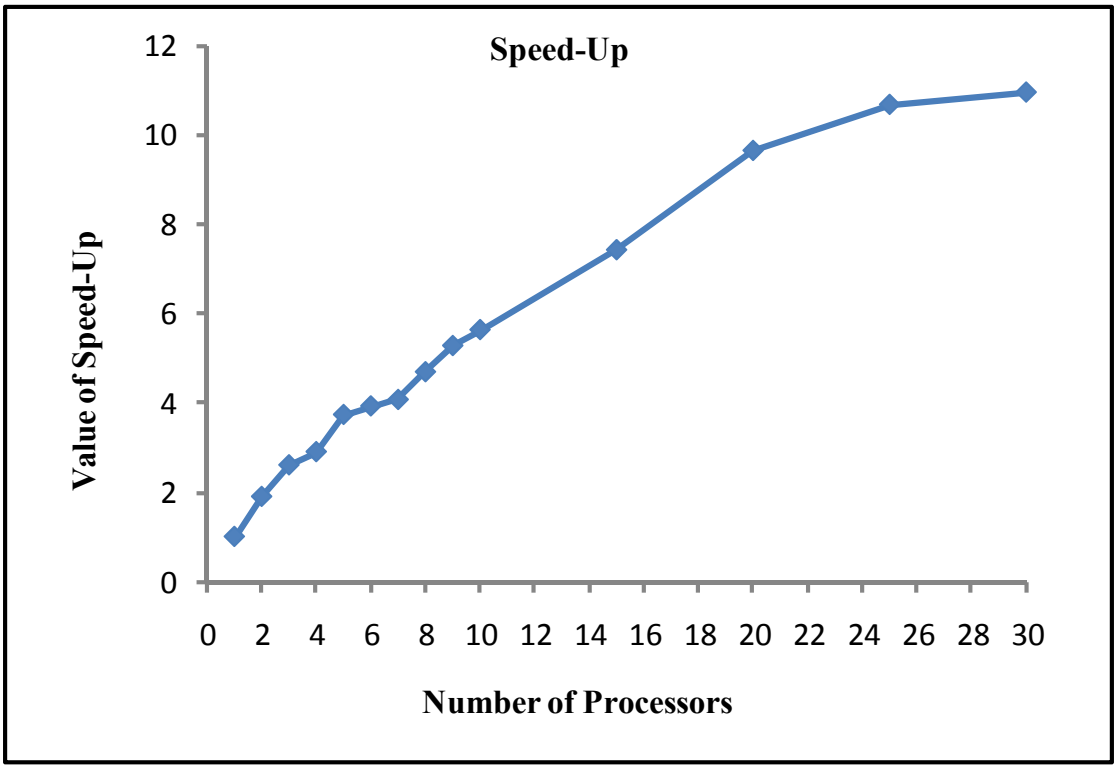

Figure 6.4 Value of Speed-Up in terms of Different Number of Processors

To get a better view of the trend of this sensitivity test, the values of Speed-Up in terms of different processors are indicated by Figure 6.4. An interesting phenomenon shown by Figure 6.4 is that when the number of processors used is less than 10, the value of Speed-Up is linearly increasing, while this increase is decelerated when more than 10 processors are used. This phenomenon can be ascribed to two reasons: (a) when a new processor is added, there should be a trade-off between its marginal computation effort and the marginal cost resulting from additional data communication load. Yet, thanks to the advanced 10G Myrinet technology adopted in the distributed computing system, the total data communication time is quite trivial, thus each additional processor can fully contributed to the speedup of computation, and this results to an approximately linear Speed-Up increase; (b) in each generation of the DRGA, the number of newly generated chromosomes varies from 10 to 40. If this number is less than the number of processors used, the redundant processors would be in idle, which can not contribute to the speedup of the calculation. Thereby, where the number of processors is larger than 10, the total idle time would dramatically increase, and performance of the distributed computation would be deteriorated.

6.5 Conclusions

This chapter dealt with the speed-based toll design problem for cordon-based congestion pricing scheme, where improving the traffic conditions in the cordon area was taken as an objective. Average travel speed was taken as an index of the traffic conditions, and a targeted range should be predefined for the average travel speed. Any toll-charge pattern that can maintain the travel speed in this targeted range is regarded as acceptable. A

MPEC model was proposed for optimal toll-charge pattern, among the acceptable ones, with maximal total social benefit value. The MPEC model takes a fixed-point model, formulated for the PA-SUEED problem with continuously distributed VOT.

A Distributed Revised Genetic Algorithm (DRGA) was then proposed for solving this speed-based toll design problem. It showed that DRGA can successfully achieve a toll-charge pattern that maintains the corresponding average travel speed in the targeted range and also attain a higher total social benefit. The numerical example further indicated that the computation can be speeded up for more than ten times.

This study is an initial step of combining the traffic conditions in CBD area as well as the issue of average travel speed into congestion pricing toll design problem. A promising research topic would be an in-depth investigation about the effect of any toll-charge pattern on the average travel speed in the cordon area. Further efforts are also required to investigate the impacts of different distribution types of the VOT on the optimal toll-charge pattern. Calibration of the VOT distribution based on practical survey data is also of considerable significance to this research topic.

CHAPTER 7 DISTANCE-BASED TOLL DESIGN FOR CORDON-BASED CONGESTION PRICING SCHEME

This chapter investigates about the distance-based toll design, which is an extension of the pay-per-entry basis charge, e.g., the toll charge pattern in Chapter 6. In reality, Land Transport Authority of Singapore intends to upgrade its Electronic Road Pricing (ERP) system from the current pay-per-entry charge to a distance-based charge, which is believed to perform better in terms of equity measures for commuters. Thus, toll design for the distance-based charge is a timely research topic with practical significance. The distance-based charges on each path within the cordon are uniquely decided by a charge function. Thus, toll design for the distance-based charge aims to find the optimal charge function that gives rise to maximal total social benefit (TSB), which is also formulated as a MPEC model. The charge function is assumed to be generic to any functional form. This property in conjunction with the non-additivity of distance-based charge has enlarged the challenges of solving the toll design problem.

7.1 Background and Relevant Studies

As claimed in Section 6.1, cordon-based pricing is apparently dominant in the practical implementations of congestion pricing schemes. All the implemented cordon-based congestion pricing schemes currently use a flat toll-charge method including the daily licensing basis charge (Santos, 2008) and the pay-per-entry basis charge, regardless of the travel distance or time in the pricing cordons. This flat toll-charge method, however, is inequitable because it undercharges the longer journeys and over-restrains the shorter ones (May et al., 2008).

To cope with this inequity issue, May and Milne (2000) examined three possible alternative toll-charge methods: (a) *time-based* method according to the time consumed in traversing a cordon; (b) *congestion-based* method in accordance with the additional travel time spent in congestion; and (c) *distance-based* method relied on the distance travelled. May and Milne (2000) concluded that these three methods outperformed the flat toll-charge method in terms of traffic congestion mitigation. The first two toll-charge methods, to some extent, encourage aggressive driving behaviors and may cause more traffic safety issues. They are hence not adopted in practical trials (Richards et al., 1996). The technology for distance-based toll-charge method is already available, and it can be efficiently implemented with the aid of the global positioning system (GPS) and an in-vehicle unit integrating a GPS receiver, a digital map and a general packet radio service (GPRS) communication device. It is more preferable for the next generation of congestion pricing schemes taking into account the inequity issue. Note that toll charges for the distance-based toll-charge method should be a function of the travel distance in each pricing cordon, which is termed distance-based toll-charge function. It thus makes the toll charges, in most cases, non-additive.

Land Transport Authority (LTA) of Singapore has updated the bus fares for public transport system to be distance-based. The bus fares are determined by a universal fare structure table, and as shown by Figure 7.1 the bus fare in such case is a nonlinear function of travel distance. Likewise, LTA intends to update its Electronic Road Pricing (ERP) system from the current pay-per-entry charge to the distance-based charge termed

as second generation ERP system (Ohno, 2007). Despite the communication technology challenges, it is essential for the LTA of Singapore to determine a proper distance-based toll-charge function for the second generation ERP. It is therefore a new research issue with practical importance to estimate a distance-based toll-charge function that maximizes the total social benefit (TSB). To better reflect the commuters' travel behavior, PA-SUEED with continuously distributed VOT is also assumed as a framework for the analysis of network equilibrium in this chapter.

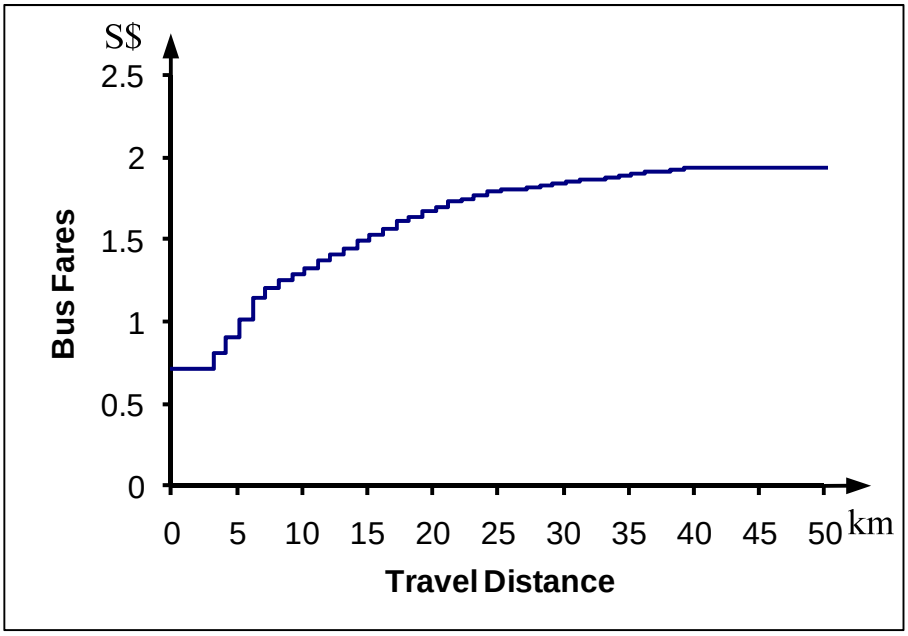

Figure 7.1 Bus Fare Structure for Public Transport System in Singapore

In addition to the second generation ERP system in Singapore, the importance and significance of the optimal distanced-based toll design addressed in this study can be further demonstrated by the experiments of a kilometer (KM)-based toll-charge method conducted in several European regions including Leeds, UK (Mitchell et al., 2005; Namdeo and Mitchell, 2008), Scotland (O'Mahony et al., 2000), Netherlands (Ubbels et

al., 2002) and Germany (Hensher and Puckett, 2007; Link, 2008). As a special distance-based toll-charge method, the KM-based toll-charge method assumed that toll charge for each vehicle was linearly proportional to its travel distance in the pricing area (May and Milne, 2000), making the toll charge additive. Yet, previous studies and trials for KM charge all arbitrarily set a charging rate per kilometer, which is unlikely to achieve the maximal TSB. Therefore, this chapter also takes the toll design problem for KM-based toll-charge method into analysis, which can be briefly solved by a brute-force method.

In spite of its simplicity in modeling and calculation, KM-based toll-charge function is unlikely to be the optimal for all the transportation networks. Yet, no practical data is available for the analysis of a proper functional form of the toll-charge function or to calibrate such a function. It is therefore more reasonable to assume that the toll-charge function is generic to any positive and non-decreasing function, which includes the KM-based toll-charge function, linear function, and nonlinear functions.

Thereby, this chapter aims to propose a methodology that is able to estimate a positive and non-decreasing toll-charge function with maximal TSB, which is formulated as a MPEC model. Likewise to the speed-based toll design, the fixed-point model, proposed for the PA-SUEED problem with continuously distributed VOT, is also taken as a constraint to the MPEC model. Since the distance-based toll-charge function is allowed to be any functional form, it is prohibitive to solve this MPEC model. Thus, a piecewise-linear function is proposed to approximate any form of the toll-charge function, which gives another mixed-integer MPEC model for the convenience of calculation.

For solving the mixed-integer MPEC model, due to the random VOT, existing algorithms for the MPEC model, see Section 2.3.1, are not available. On the other hand, considering the integer property of the piecewise-linear approximation function, it is convenient to adopt the Genetic Algorithm (GA) as a solution method.

For each given toll-charge function, its impacts on the network equilibrium flows are analyzed by solving a PA-SUEED problem with continuously distributed VOT. However, the non-additivity of distance-based toll charge requires path enumeration/generation, thus increases the computational costs and also prohibits the link-based Monte Carlo simulation method to solve the SNL problem. To cope with this hurdle, a network transformation technique is proposed, which enumerates the cycle-free paths in the cordon, and therefore enables the link-based solution algorithm.

The remaining sections are organized as follows. Section 7.2 defines the optimal distance-based toll design problem and Section 7.3 put forward the network transformation technique that enables a link-based Monte Carlo simulation for solving the SNL. Section 7.4 first develops a MPEC model for the toll design problem, and then introduces a mixed-integer MPEC model based on the piecewise-linear approximation function. Solution algorithms and numerical experiments are presented in Sections 7.5 and 7.6. Conclusions are finally provided in Section 7.7.

7.2 Toll-Charge Function and Optimal Distance-based Toll Design

Similarly to the previous chapters, PA-SUEED problem is also assumed to depict the

commuters' travel behavior on the strongly connected network $G=(N,A)$. Let I be the number of congestion pricing cordons in network G. Network $G=(N,A)$ is then divided into one external network, denoted by $\hat{G}=(\hat{N},\hat{A})$, and several cordon networks, denoted by $\bar{G}_i=(\bar{N}_i,\bar{A}_i)\ (i=1,2,\cdots,I)$. The links (nodes) in any cordon network are termed as internal links (nodes), while the other links (nodes) in external network $\hat{G}=(\hat{N},\hat{A})$ are termed as external links (nodes). Note that the boundary nodes of any cordon i exist both in its cordon node set $\bar{N}_i \subseteq N$ and the external node set $\hat{N} \subseteq N$. Let E_i be the set of all the entry-exit pairs of cordon $i \in I$, and $d_k^{w,i}$ denote the length of the portion of path $k \in R_w$ in the congestion pricing cordon i and it can be expressed as follows:

$$d_k^{w,i}=\sum_{e\in E_i}\sum_{a\in A} l_a \delta_{ak}^w \delta_{ek}^w, k\in R_w, w\in W, \tag{7.1}$$

where l_a is the length of link $a \in A$ and δ_{ek}^w is entry-exit pair/path incidence indicator, i.e., $\delta_{ek}^w=1$ if path $k \in R_w$ between OD pair $w \in W$ passes through entry-exit pair $e \in E_i$ of cordon i and $\delta_{ek}^w=0$, otherwise.

A distance-based toll-charge method for the cordon-based congestion pricing scheme can be expressed by a distance-based toll-charge function $\phi(d)$ where d is the distance traveled in a pricing cordon. It is assumed that the generic function $\phi(d)$ is positive and non-decreasing but may not be continuous. As shown by Figure 7.1, the toll-charge

function for distance-based bus fare in Singapore is not continuous. $\phi(d)$ is also allowed to follow any function form, including, linear function, quadratic function, exponential function, power function, etc. The kilometer (KM) based toll-charge function is a special case of $\phi(d)$, where only the slope is a decision variable, shown as follows:

$$\phi(d) = \rho d, \tag{7.2}$$

where ρ is the slope of KM based toll-charge function, which is the charging rate per kilometer.

For any given toll-charge function $\phi(d)$, the toll charge imposed on path $k \in R_w$ between OD pair $w \in W$ can be calculated as follows:

$$\tau_{wk}(\phi) = \sum_{i=1}^{I} \phi\left(d_k^{w,i}\right). \tag{7.3}$$

Toll charge $\tau_{wk}(\phi)$ shown in eqn. (7.3) can be converted into time-units by virtue of the drivers' value of time (VOT), denoted by α. α is assumed to be continuously distributed with a continuously differentiable probability density function. The generalized path travel time in this case can be defined as follows, which is similar to that defined in Chapter 6 at eqn. (6.2).

$$\bar{C}_{wk}\left(\mathbf{v}, \boldsymbol{\tau}_w(\phi)\right) = c_{wk}(\mathbf{v}) + \zeta_{wk} + \frac{\tau_{wk}(\phi)}{\alpha}, \tag{7.4}$$

And the generalized perception error can also be defined as:

$$\lambda_{wk} = \frac{\tau_{wk}(\phi)}{\alpha} - E\left(\frac{\tau_{wk}(\phi)}{\alpha}\right) + \zeta_{wk}. \tag{7.5}$$

All the toll charges on the network are determined by the toll-charge function shown in eqn. (7.3). Different toll-charge functions will result in different network equilibrium

flows and consequently different total social benefit (TSB) values on the entire network. Let Φ be the set of all these positive and non-decreasing toll-charge functions. Assuming the PA-SUEED for drivers' route choice behavior, the distance-based toll design problem aims to identify or estimate a proper toll-charge function with the maximal TSB from the set Φ.

The number of feasible toll charge function in Φ is infinity. For each individual toll charge function $\phi \in \Phi$, it would influence the drivers' route choice plans, and thus its impacts on the network are usually analyzed by solving a route choice problem, which in this case is solving a PA-SUEED problem. Equilibrium flows of the PA-SUEED problem then can be used to calculate the corresponding TSB value of this $\phi \in \Phi$. Since the distance-based toll charge is not additive to the links, it is more challenging to solve such a PA-SUEED problem, for which the mathematical models and solution algorithms are discussed in the following sub-section.

7.3 PA-SUEED Problem with Non-additive Distance-based Charge

As discussed in Section 6.2.3, the PA-SUEED problem with continuously distributed VOT can be also formulated by means of the following fixed-point model:

$$v_a(\phi) = \sum_{w \in W} \sum_{k \in R_w} \left[D_w\left(S_w\left(\mathbf{c}_w(\mathbf{v}), \boldsymbol{\tau}_w(\phi)\right)\right) \times p_{wk}\left(\mathbf{c}_w(\mathbf{v}), \boldsymbol{\tau}_w(\phi)\right) \delta_{ak}^w \right], a \in A. \quad (7.6)$$

Wherein, the fixed-point model in this case also has a unique solution. Solution algorithms, see Chapter 3, require the calculation of SNL in each iteration, which can be solved by the link-based two-stage Monte Carlo simulation method proposed in Section 3.4. Nevertheless, in this chapter, due to the existence of non-additive distance-based

charge, the link-based Monte Carlo simulation method is not available. Thus, to tackle this problem, a network transformation technique is first proposed as follows.

7.3.1 Network Transformation for Non-additive Path Toll Charges

A pricing cordon usually has limited size, and in such cases it is effortless to enumerate all the simple paths (acyclic paths) between any entry-exit pair $e \in E_i$, which are termed as internal paths. Thus, a network transformation can be efficiently conducted to convert the non-additive path-based toll charges into link-based: firstly, each internal path between entry-exit pair $e \in E_i$ is represented by a dummy link connecting its entry and exit node; secondly, replace all the cordon networks $\bar{G}_i = \left(\bar{N}_i, \bar{A}_i\right)$ $\left(i = 1, 2, \cdots, I\right)$ by corresponding dummy links. Hereby, external network $\hat{G} = \left(\hat{N}, \hat{A}\right)$ and all the dummy links constitute a new composite network, denoted by $G' = \left(N', A'\right)$. We can see that this network transformation converts the "path-based" toll charges to be "link-based" on each dummy link.

An illustrative example. A small example shown in the upper part of Figure 7.2 is used to illustrate the network transformation procedure. Links 1 to 6 and Nodes 1 to 5 constitute the external network, $\hat{G} = \left(\hat{N}, \hat{A}\right)$, of this example. The area surrounded by a dotted ellipse is a pricing cordon, and therefore Links 7 to 12 and Nodes 2, 4, 6 compose the cordon network, $\bar{G} = \left(\bar{N}, \bar{A}\right)$. Entry-exit pairs of the pricing cordon comprise $2 \to 4$ and $4 \to 2$, and any trip traversing this cordon is imposed by a distance-based toll charge.

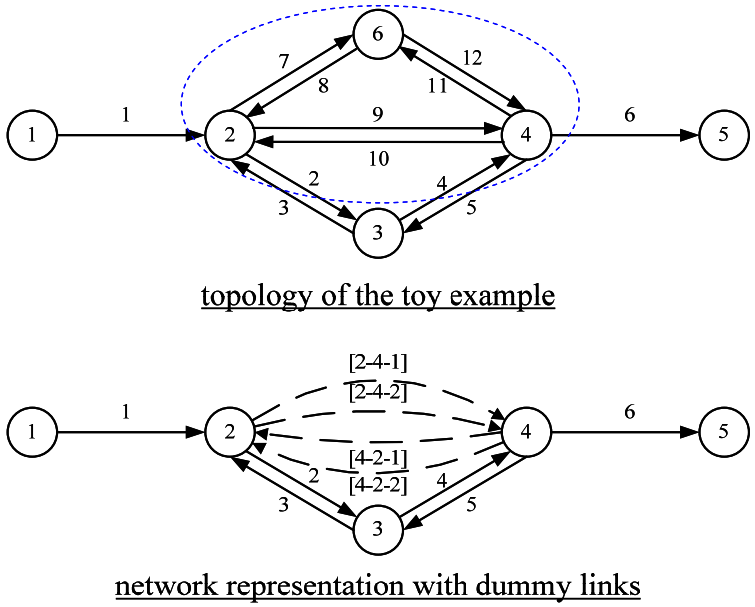

Figure 7.2 Illustrative Example for Network Transformation

A new network, $G' = (N', A')$, modified by the network transformation procedure is shown in the lower part of Figure 7.2. It can be seen that the pricing cordon is replaced by four dummy links (dotted arcs), and as shown in Table 7.1 each of these dummy links represents a simple path.

Table 7.1 List of Simple Paths and Corresponding Dummy Links

Entry-exit pairs	Simple paths	Represented by dummy link
$2 \to 4$	$2 \to 4$	$[2-4-1]$
	$2 \to 6 \to 4$	$[2-4-2]$
$4 \to 2$	$4 \to 2$	$[4-2-1]$
	$4 \to 6 \to 2$	$[4-2-2]$

All the internal paths between entry-exit pair, $e \in E_i$ are grouped in set $\overline{R}_e$. Evidently, based on a given toll-charge function, ϕ, toll charge on each dummy link of composite

network $G' = (N', A')$ is a flow-independent constant, and generalized travel time on any path is additive to the links.

Remark 7.1: In the case of a realistic network with large pricing cordons, there could be quite a number of simple paths between one entry-exit pair, thus considerably increasing the computational burden. Previous studies for path-based traffic assignment problems are confronted by this hurdle, see, e.g., Lo and Chen (2000), Bekhor and Toledo (2005). Instead of enumerating all the simple paths, these studies only examine partial paths (a) explicitly in a pre-generated path choice set or (b) implicitly based on the column generation method during iterative computations. These techniques of using partial paths are also recommended to cope with the internal paths on a realistic network.

7.3.2 A Monte Carlo Simulation Method on the Composite Network

Based on the composite network $G' = (N', A')$, the link-based two-stage Monte Carlo simulation method then can be used for solving the SNL of PA-SUEED. Procedures of this Monte Carlo simulation method are more or less different from the procedures introduced in Section 3.4.2, thus they are also summarized as follows, for the sake of completeness.

Stage 1: Monte-Carlo simulation for travel demand estimation

Step 1.0: (Initialization) Let the number of simulations be $n = 1$ and the initial estimated satisfaction be $\bar{S}_w^{(0)} = 0, w \in W$.

Step 1.1: (Sampling of link travel time) For each link $a \in A$, sample a value for the perception error $\overline{\xi}_a^{(n)}$ from the normally distributed population $N\left(0, \beta t_a^0\right), a \in A$. And then calculate the link travel time $\tilde{T}_a^{(n)} = t_a + \overline{\xi}_a^{(n)}$.

Step 1.2: (Generalized travel time of dummy links) Sample a value for VOT $\alpha^{(n)}$ from its distribution function and calculate the generalized travel time on all the internal paths $l \in \overline{R}_e, e \in E$ by

$$\tilde{T}_{el}^{(n)} = \sum_{a \in \overline{A}} \tilde{T}_a^{(n)} \delta_{al}^e + \frac{\phi\left(d_{el}\right)}{\alpha^{(n)}}, \tag{7.7}$$

where δ_{al}^e equals to 1 if internal path $l \in \overline{R}_e$ contains internal link $a \in \overline{A}$, and 0, otherwise. d_{el} is the length of internal path $l \in \overline{R}_e$. $\tilde{T}_{el}^{(n)}$ is recorded as the travel time on the corresponding dummy link of composite network $G' = \left(N', A'\right)$.

Step 1.3: (Shortest path time calculation). With link travel time pattern $\left\{\tilde{T}_a^{(n)}, a \in A'\right\}$ on the composite network $G' = \left(N', A'\right)$, find the shortest path between each OD pair $w \in W$, and record the total travel time on this path as $\tilde{C}_w^{(n)}$.

Step 1.4: (Satisfaction estimation). Estimate the satisfaction for each OD pair $w \in W$ by the following average scheme:

$$\overline{S}_w^{(n)} = \frac{\left(n-1\right)\overline{S}_w^{(n-1)} + \tilde{C}_w^{(n)}}{n}, w \in W. \tag{7.8}$$

Step 1.5: (Accuracy checking). If the number of iterations $n \geq n_0$, where n_0 is a predetermined sample size, go to Step 1.6; otherwise, set $n = n + 1$ and go to Step 1.1.

Step 1.6: (OD demand calculation). Calculate OD travel demand pattern by the formulae:

$$q_w = D_w\left(\overline{S}_w^{(n)}\right), w \in W . \tag{7.9}$$

Stage 2: Monte-Carlo simulation for calculating the generalized probit-based SUE link flow

Step 2.0: (Initialization). Set the initial link travel flow vector be $v_a^{(0)} = 0, a \in A$ and $m = 1$

Step 2.1: (Sampling). Sample the perception error $\overline{\xi}_a^{(m)}$ from $N\left(0, \beta t_a^0\right), a \in A$ based on normally distributed random number series, and then calculate the link travel time $\tilde{T}_a^{(m)} = t_a + \overline{\xi}_a^{(m)}, a \in A$.

Step 2.2: (Sampling of VOT). Sample a value for VOT $\alpha^{(m)}$ from its distribution function and calculate the travel time on all the internal paths $l \in \overline{R}_e, e \in E$ by

$$\tilde{T}_{el}^{(m)} = \sum_{a \in \overline{A}} \tilde{T}_a^{(m)} \delta_{al}^e + \frac{\phi\left(d_{el}\right)}{\alpha^{(m)}}. \tag{7.10}$$

Then, record $\tilde{T}_{el}^{(m)}$ as the travel time on corresponding dummy link of the composite network $G' = \left(N', A'\right)$.

Step 2.3: (All-or-nothing assignment). Based on link travel time pattern $\left\{\tilde{T}_a^{(m)}, a \in A'\right\}$ on the composite network $G' = \left(N', A'\right)$, find the shortest path for each OD pair w, then assign q_w to the shortest path. Flow on link $a \in A'$ is therefore a

summation of the travel demands of all the OD pairs whose shortest path uses link $a \in A'$. Then assign flows on all the dummy links to the links on their corresponding internal paths $l \in \bar{R}_e, e \in E$. This would generate an auxiliary flow pattern on all the links $\left\{Y_a^{(m)}, a \in A\right\}$. Calculate the simulated link flow by

$$v_a^{(m)} = \frac{(m-1)v_a^{(m-1)} + Y_a^{(m)}}{m}, a \in A. \tag{7.11}$$

Step 2.4: (Convergence test). If the number of iterations $m \geq m_0$, where m_0 is also a predetermined sample size, stop; otherwise, set $m = m + 1$ and go to step 2.1.

Accordingly, the three solution algorithms, CA method and two hybrid PC-CA algorithms, discussed in Chapter 3 incorporating this Monte Carlo simulation-based SNL method can be adopted to solve the PA-SUEED problem with continuously distributed VOT, based on any given toll-charge function $\phi \in \Phi$.

7.4 Two MPEC Models for the Optimal Distance-Based Toll Design

7.4.1 Total Social Benefit and the Exact MPEC Model

The distance-based toll design problem is also formulated as a MPEC model with the objective of maximizing total social benefit (TSB). In the framework of PA-SUEED with continuously distributed VOT, the TSB is defined as follows:

$$\begin{aligned} Z(\phi) = & \sum_{w \in W} \int_0^{q_w(\boldsymbol{\tau})} D_w^{-1}(x)dx - \sum_{w \in W} q_w(\boldsymbol{\tau}) S_w\left(\mathbf{c}_w\left(\mathbf{v}(\boldsymbol{\tau})\right), \boldsymbol{\tau}_w(\phi)\right) \\ & + \sum_{e \in E} \sum_{l \in \bar{R}_e} E\left(\frac{\bar{f}_{el}(\boldsymbol{\tau})\tau_{el}(\phi)}{\alpha}\right), \end{aligned} \tag{7.12}$$

where $E(\bullet)$ is the expectation operator, and $\boldsymbol{\tau} = \left(\boldsymbol{\tau}_w(\phi), w \in W\right)^{\mathrm{T}}$. $\mathbf{q}(\boldsymbol{\tau}) = \left(q_w(\boldsymbol{\tau}), w \in W\right)^T$, $\mathbf{v}(\boldsymbol{\tau}) = \left(v_a(\boldsymbol{\tau}), a \in A\right)^T$, and $\bar{\mathbf{f}}(\boldsymbol{\tau}) = \left(\bar{f}_{el}(\boldsymbol{\tau}), e \in E, l \in \bar{R}_e\right)^T$ denote the column vectors for the equilibrium OD demands, link flows, and internal path flows, associated with a given toll-charge function ϕ. Value of these vectors can be obtained by solving the PA-SUEED problem addressed in Section 7.3.

Hence, A MPEC model is built to find a toll-charge function maximizing the TSB:

$$\max_{\phi \in \Phi} Z(\phi) \tag{7.13}$$

subject to

$$v_a(\phi) = \sum_{w \in W} \sum_{k \in R_w} \left[D_w\left(S_w\left(\mathbf{c}_w(\mathbf{v}), \boldsymbol{\tau}\right)\right) \times p_{wk}\left(\mathbf{c}_w(\mathbf{v}), \boldsymbol{\tau}\right) \delta_{ak}^w \right], a \in A \tag{7.14}$$

Constraint (7.14) is the fixed-point model proposed for the PA-SUEED problem with continuously distributed VOT.

As aforementioned, the toll-charge function is merely required to be positive and non-decreasing, and it is generic to any linear or nonlinear functional forms. This type of toll-charge function has a sound rationality, thus model (7.13)-(7.14) is termed as exact MPEC model. Yet, it is quite challenging to solve this model, since the toll-charge function has no specific functional form. In view of this, an approximation method is proposed in the following section, where a piecewise-linear function is used to approximate any positive and non-decreasing function. This approximation method then gives a mixed-integer MPEC model that can be simply solved.

7.4.2 A Mixed-integer MPEC Model with a Piecewise-linear Approximation Function

A piecewise-linear approximation function is indicated by Figure 7.3. Suppose the nonlinear function in the left-hand-side of Figure 7.3 is a feasible toll-charge function, it then can be approximated by a piecewise-linear function as shown in the right-hand-side of Figure 7.3. The shape of a piecewise-linear function in each interval is uniquely determined by its two boundary values.

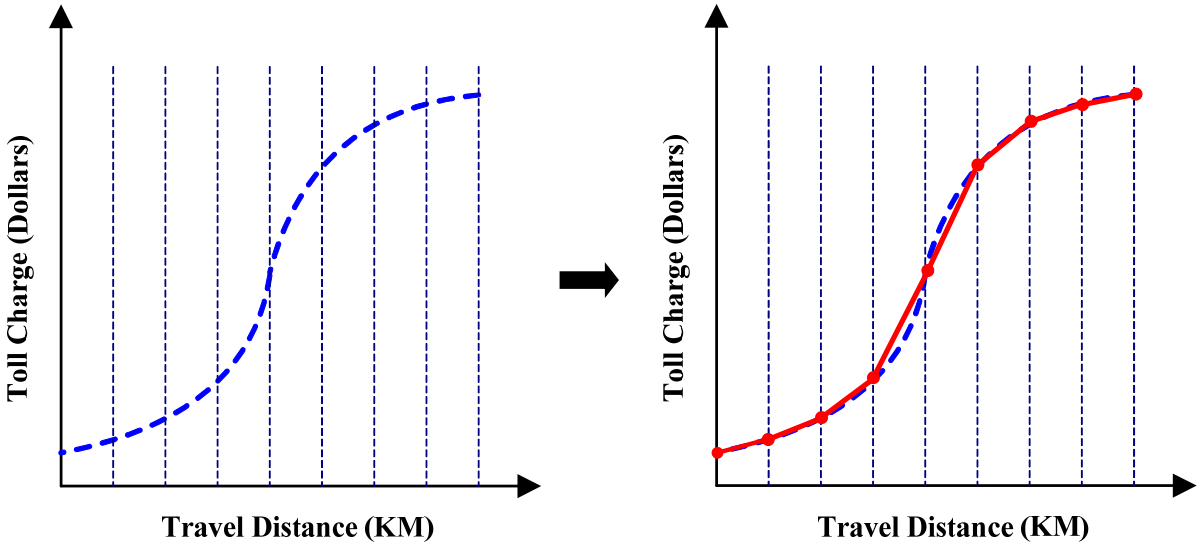

Figure 7.3 An Illustrative Example for the Piecewise-linear Approximation Function

Theoretical definition of the piecewise-linear approximation function is then discussed. Let $d_{\min}$ and $d_{\max}$ be the minimum and maximum lengths of all the internal paths. The range $[d_{\min}, d_{\max}]$ can be uniformly dividend into n intervals, where n is a positive integer number,

$$[d_{\min}, d_2], [d_2, d_3], \ldots, [d_i, d_{i+1}], \ldots [d_n, d_{\max}]. \tag{7.15}$$

The piecewise-linear approximation function contains $n+1$ boundary values over these n intervals, and these boundary values are defined as $y_1 = \phi(d_{\min})$, $y_i = \phi(d_i), i = 2,3\ldots,n$ and $y_{n+1} = \phi(d_{\max})$. The piecewise-linear approximation function is defined accordingly as follows:

$$\bar{\phi}_n(d,\mathbf{y}) = y_i + \frac{y_{i+1} - y_i}{d_{i+1} - d_i}(d - d_i), d_i \le d < d_{i+1}, i = 1,2,\ldots n, \tag{7.16}$$

where $d_1 = d_{\min}$ and $d_{n+1} = d_{\max}$. The column vector $\mathbf{y}$ is defined as $\mathbf{y} = (y_1, y_2, \ldots y_{n+1})^{\mathrm{T}}$.

The number of intervals, n, can influence the fitness of $\bar{\phi}_n(d,\mathbf{y})$ to the original toll-charge function $\phi(d)$. It is reasonable to assume that:

$$n \in [n_{\min}, n_{\max}], \text{and } n \text{ is an integer}. \tag{7.17}$$

Herein, the lower bound $n_{\min}$ is defined, because it will deteriorate the equity of distance-based toll-charge method if the value of n is fairly small. And on the other hand, if the number of intervals is too large, computational cost of solving the optimal piecewise-linear function $\bar{\phi}_n^*$ would geometrically increase, thus an upper bound $n_{\max}$ is also defined for n.

Based on the piecewise-linear function (7.16), the MPEC model (7.13)-(7.14) can be approximated by the following mixed-integer MPEC model with the decision variables $\mathbf{y}$ and n:

$$\max_{\mathbf{y},n} \bar{Z}\left(\bar{\phi}_n(d,\mathbf{y})\right) \tag{7.18}$$

subject to

$$v_a\left(\bar{\phi}_n\left(d,\mathbf{y}\right)\right)=\sum_{w\in W}\sum_{k\in R_w}\left[D_w\left(S_w\left(\mathbf{c}_w\left(\mathbf{v}\right),\boldsymbol{\tau}\right)\right)\times p_{wk}\left(\mathbf{c}_w\left(\mathbf{v}\right),\boldsymbol{\tau}\right)\delta_{ak}^{w}\right],a\in A \tag{7.19}$$

$$y_{\min}\le y_1 \tag{7.20}$$

$$y_{i-1}\le y_i\le y_{i+1},\forall i=2,3,\ldots,n-1 \tag{7.21}$$

$$y_n\le y_{\max} \tag{7.22}$$

$$n_{\min}\le n\le n_{\max}\ \text{and}\ n\ \text{is an integer} \tag{7.23}$$

where $y_{\min}$ ($y_{\max}$) is a predetermined lower (upper) bound of the toll charge. Eqns. (7.20) -(7.22) ensure that the piecewise-linear function is positive and non-decreasing. This mixed-integer MPEC model is then used to solve optimal distance-based toll design problem, solution algorithm of which is provided in the following section.

7.5 Solution Algorithm

As claimed in Section 7.1, due to the complicated properties of the PA-SUEED problem, existing algorithms for MPEC models are not available in this case. However, for any given function type of toll-charge function with limited number of un-known variables, it would be an accurate and un-expensive way to enumerate and assess all the feasible variables with a small gap. For instance, considering the KM-based toll-charge function, shown in eqn. (7.2), only the slope ρ is a variable, and a reasonable range for ρ is usually not quite large, thus it is pleasant to test all the values of ρ in this range with a small increment.

The brute-force enumeration method is unacceptable for the piecewise-linear approximation function $\overline{\phi}_n(d,\mathbf{y})$. While, considering the discrete property of $\overline{\phi}_n(d,\mathbf{y})$, when the value of n is given, it would be more straightforward to adopt Genetic Algorithm (GA) as a heuristic for solving the mixed-integer MPEC model (7.18)-(7.23). Thereby, all the feasible n are first enumerated and for any particular n, the optimal piecewise-linear function that yields maximal TSB can be solved by GA, and this method is named as Hybrid GA. Note that a feasible chromosome for the piecewise-linear toll-charge function $\overline{\phi}_n(d,\mathbf{y})$ is given below:

y_1	y_2	……	y_i	y_{i+1}	……	y_{n-1}	y_n

and y_i here is defined as a gene of one chromosome. Each chromosome will give rise to different toll charges on the network, which therefore leads to different values of total social benefit (TSB). Detailed procedures of the Hybrid GA are summarized as follows:

Step 1: (Initialization). Set initial value for the number of intervals be $n^{(1)} = n_{\min}$. And initialize the optimal TSB value be $\mathrm{TSB}^* = 0$ and the optimal number of intervals be $n^* = n^{(1)}$. Set the number of iteration be $p = 1$.

Step 2: With the number of intervals equal to $n^{(p)}$, calculate the optimal piecewise-linear function $\overline{\phi}^*_{n^{(p)}}$ with maximal TSB value, denoted by $\mathrm{TSB}^{(p)}$, by the following GA:

Step 2.0: (Initial population). Randomly generate initial population of the piecewise-linear function, and each individual carries a feasible chromosome. Then, set the number of generation $k = 1$.

Step 2.1: (Evaluation). Based on the toll-charge function shown by one chromosome, calculate the value of toll charge on each internal path of the

composite network $G' = (N', A')$. Then, calculate the corresponding PA-SUEED problem with continuously distributed VOT, results of which are used to evaluate the TSB for this toll-charge function. Conduct such evaluations for all the new generated chromosomes.

Step 2.2: (Selection). Select the chromosomes with higher TSB values as survivors for current generation and discard the rest.

Step 2.3: (Crossover). Conduct pairing among survivors, where for each two chosen survivors, denoted by $\bar{Y}_1$ and $\bar{Y}_2$, two new chromosomes are generated by the following function:

$$\begin{aligned} \hat{Y}_1 &= \chi \bar{Y}_1 + (1-\chi)\bar{Y}_2 \\ \hat{Y}_2 &= \chi \bar{Y}_2 + (1-\chi)\bar{Y}_1 \end{aligned} \quad \chi \in (0,1). \tag{7.24}$$

Step 2.4: (Mutation). To guarantee the monotonicity of the toll-charge map, a mutation is conducted in this way: first, randomly choose some genes from existing chromosomes; second, for each chosen gene y_i, update its value by sampling a uniformly distributed random variable in $[y_{\min}, y_{\max}]$, then proportionally change the value of other genes of this chromosome in the interval $[y_{\min}, y_i]$ and $[y_i, y_{\max}]$.

Step 2.5: (Stop test). If a stop criterion is achieved, let the value of $\mathrm{TSB}^{(p)}$ equal to the maximal TSB among the survivors in current generation and record the corresponding chromosome, and then go to Step 3; otherwise, set $k = k+1$ and go to step 2.1.

Step 3: If $\mathrm{TSB}^{(p)} > \mathrm{TSB}^{*}$, then let

$$\begin{aligned} \mathrm{TSB}^* &= \mathrm{TSB}^{(p)} \\ n^* &= n^{(p)}, \end{aligned} \tag{7.25}$$

and record the corresponding chromosome as the optimal toll-charge function.

Step 4: If $n^{(p)} \geq n_{\max}$, then stop and output the optimal toll-charge function, TSB^* and n^*; otherwise, let $n^{(p+1)} = n^{(p)} + 1$, $p = p + 1$ and go to Step 2.

7.6 Numerical Experiment

To numerical validate the proposed methodology, a numerical example, named as Network C, is adopted. This example was used by Meng et al. (2004) as well as Yang et al. (2004) for the non-additive entry-exit highway charge schemes. It contains 13 nodes, 46 links and 8 OD pairs. As shown in Figure 7.4, the Network C has one pricing cordon, encircled by the dotted rectangular. Links in the cordon are grouped in the set:

$$\bar{A} = \{33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46\}.$$

Nodes 4, 5, 6 and 7 are entries as well as exits to this pricing cordon.

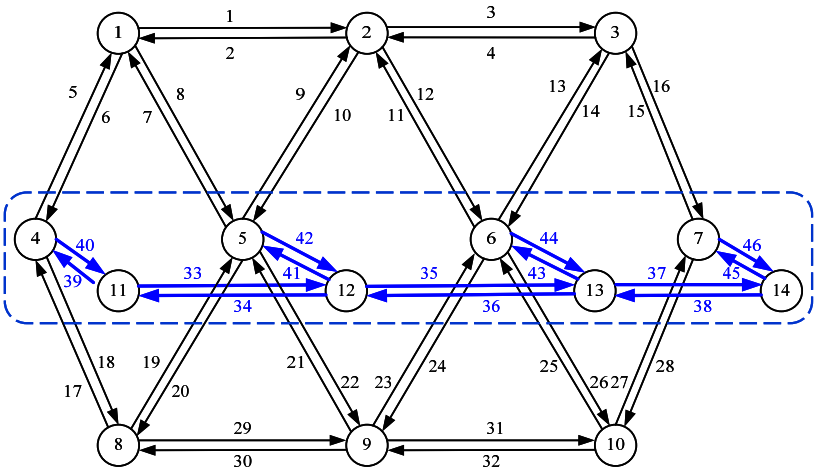

Figure 7.4 Structure of Network C

Demand functions and asymmetric link travel time functions are identical to those in eqns. (3.82) and (3.86). Some parameters for the network attributes, including the upper bound of OD demand $\overline{q}_w$ and values of the t_a^0 and h_a on each link, are identical to those in Yang et al. (2004), which are not included here. The parameter μ in eqn. (3.82) is properly taken as 0.01 for this example. Length of each link is given in Table 7.2, which is used to measure drivers' travel distance in the pricing cordon. Based on the link lengths, it is detected that the maximum and minimum path length, $d_{\min}$ and $d_{\max}$, for this example is 405 and 2453 meters, respectively.

Table 7.2 Length of the Links in the Pricing Cordon of Network C

Link No.	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Length(meters)	790	1050	250	400	850	892	85
Link No.	**40**	**41**	**42**	**43**	**44**	**45**	**46**
Length(meters)	38	35	78	82	66	85	26

The value-of-time (VOT) is assumed to be continuously distributed across the whole population. It is assumed that the VOT is uniformly distributed in the range from 18.0 to 72.0 Singapore-Dollars (S$) per hour. Sample sizes for the two stages of Monte Carlo simulation are taken as 180 and 1000, respectively, to achieve an acceptable accuracy level.

As described in Section 7.5, optimal toll design problem for the KM charge is solved by a brute-force enumeration method and for the general nonlinear distance-based charge is solved by the Hybrid GA. Both of these two methods possess a good parallelism, thus it is quite convenient to use distributed computing to accelerate the computational speed in

each case. For the enumeration method for KM charge, the range of slope ρ can be partitioned into many intervals and then evaluated by different processors in the distributed computing system. For the Hybrid GA, evaluation of each newly generated chromosome is independent and it thus can be conducted concurrently by different processors, which is discussed at Section 6.3.2. With sufficient processors, computational speed of these two methods can be accelerated for tens of times. For conciseness, the sensitivity test for the number of processors is not conducted again for this example, and the number of processors is fixed to be 50.

7.6.1 KM Charge

The optimal KM-based toll-charge function is first coped with, since it can be efficiently solved by the brute-force enumeration method and in addition it can be taken as a benchmark to evaluate the performance of optimal nonlinear distance-based charge. As shown by eqn. (7.2), only the slope ρ is an unknown parameter for the KM toll-charge function. It is assumed that ρ ranges from 0.5 to 10.0 S$ per km. Thus, 951 successive values of ρ in this range with an increment of 0.01 are tested, and the corresponding values of total social benefit (TSB) are shown in Figure 7.5.

Figure 7.5 indicates that the value of TSB achieves its maximum at around $\rho = 0.92$, and the optimal TSB value is 5.204×10^6. While, the smallest TSB value obtained during this enumeration is only 4.094×10^6. As shown in Figure 7.5, when the value of ρ exceeds 2.10, the value of TSB stays at this smallest level. This is because when the toll charge of traveling in a pricing cordon is getting considerably high, no flow (in the Monte Carlo

simulation) would be loaded to any link there. Hence, the pricing cordon is *blocked* from the transportation network. In such a case, the value for TSB is lower, due to the fact that: (a) there is no toll revenue; and (b) the drivers lost some options for route choice, such that their travel disutility increases.

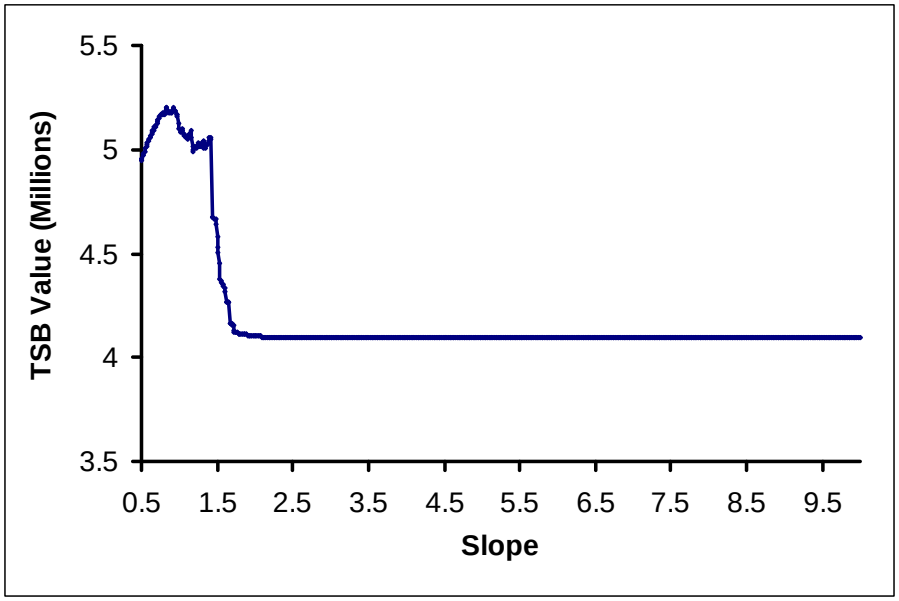

Figure 7.5 Sensitivity Test for the Slope of KM-based Toll-charge function

Another test is conducted for the case when there is no toll charge on the network, which equals to solving a PA-SUEED problem. It shows that the value of TSB in this case is 5.145×10^{6}. However, Figure 7.5 indicates that the values of TSB obtained by most of the KM-based toll-charge functions are even lower than 5.145×10^{6}. This comparison confirms that when an arbitrary toll pattern is initially introduced to the transportation network, it may cause a deduction in total benefit/utility of the network. This

phenomenon is consistent with the previous study of Wilson (1988) on the cordon-based congestion pricing scheme in Singapore.

7.6.2 Nonlinear Distance-based Charge

The Hybrid GA can be used to calculate the optimal piecewise-linear approximation function for general nonlinear distance-based charge. Then, the network transformation is first performed, where links in the pricing cordon are replaced by some dummy links that connect each entry-exit pair. The resultant composite network is shown in Figure 7.6.

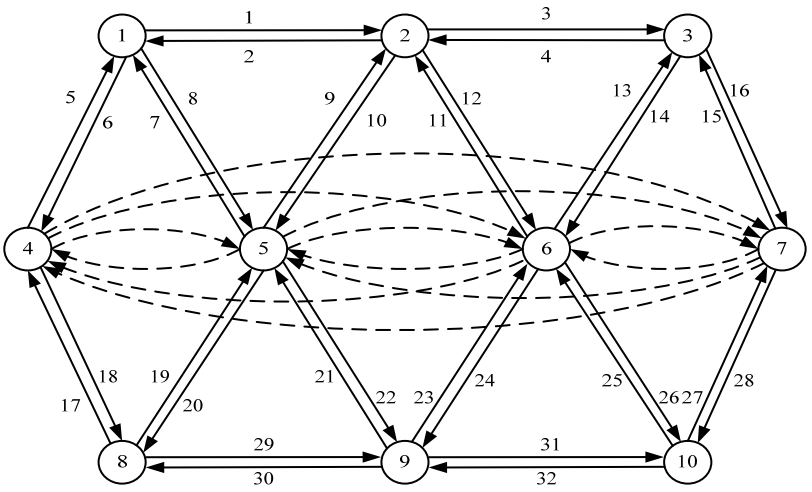

Figure 7.6 Composite Network for Network C

Since $d_{\min}$ and $d_{\max}$ for this example are found to be 405 and 2453 meters, the gap for the length-variety of all the internal paths is 2048 meters. As aforementioned, a range should be decided for the number of intervals, n, which is quite essential for the piecewise-linear approximation function. If n is too small, it would deteriorate the equity of nonlinear distance-based charge, while when n is too large, it is computationally inhibitive to solve the optimal toll design problem. Thus, the upper and lower bound for

n is properly decided to be 10 and 6, respectively. Namely, there are 5 scenarios with the value of integer n changing from 6 to 10. Computational burden is in general increasing as the value of n increases, since larger n implies more genes in each chromosome and it subsequently increases the total number of mutations.

The Hybrid GA is then adopted for solving the optimal $\overline{\phi}_n^*$ of this numerical example. The stop criterion of GA, see Step 2.5 in Section 5, is taken as fixed number of generations, $k = 50$. Moreover, the total number of chromosomes (population) of one generation is chosen to be 200, and lower and upper bound of the toll charges, shown in eqns. (7.20) and (7.22), are set to be $y_{\min} = 0.5$ and $y_{\max} = 10.0$ S$.

Table 7.3 Computational Profile of Different Scenarios to Network C

Scenarios	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
Max TSB	5541560	5789184	5728330	5802743	5877805
T1*	6916080	6949056	6939036	7019860	6964026
T2*	-2700789	-2681664	-2687571	-2637017	-2672666
T3*	1326270	1521791	1476864	1419900	1586445
Increase	6.48%	11.24%	10.07%	11.50%	12.94%
CPU Time (seconds)	826	868	890	906	930
No. of Total Chromosomes	3369	3521	3628	3777	3961

*T1, T2, T3 are terms 1, 2, 3 in the right-hand-side of eqn. (7.12)

Eventually, computational results of each scenario with different n is tabulated in Table 7.3. In table 7.3, the first column of data is the maximal TSB value obtained by each scenario, and the second to fourth column is the value of each term in the expression for

TSB, as in the right-hand-side of eqn. (7.12), namely, $\mathrm{T1} = \sum_{w \in W} \int_0^{q_w(\boldsymbol{\tau})} D_w^{-1}(x)dx$, $\mathrm{T2} = -\sum_{w \in W} q_w(\boldsymbol{\tau}) S_w\left(\mathbf{c}_w\left(\mathbf{v}(\boldsymbol{\tau})\right), \boldsymbol{\tau}_w(\phi)\right)$, and $\mathrm{T3} = \sum_{e \in E} \sum_{l \in \bar{R}_e} E\left(\frac{\bar{f}_{el}(\boldsymbol{\tau})\tau_{el}(\phi)}{\alpha}\right)$. The fifth column entitled "Increase" is the improvement of nonlinear toll charge over KM charge in terms of the TSB obtained, recalling that the maximal TSB obtained by KM charge is 5.204×10^6 . The sixth column shows the execution time for each scenario, yet this execution time may not be strictly proportional to the computational burden of each scenario. This is because the computation is performed on the distributed computing system, and each scenario with different n has different concurrency level affecting their parallelism. Most of the computational efforts are devoted onto evaluation of newly generated chromosome, thus the total number of chromosomes evaluated throughout the entire computation process could better reflect the total computational burden of each scenario, which is provided in the last column of Table 7.3.

As per the data in Table 7.3, the optimum TSB value among all the scenarios is 5.877×10^6 when $n = 10$. Compared with the maximal TSB obtained by optimal KM charge, 5.204×10^6 , the optimal nonlinear distance-based charge has a significant improvement, which is 12.94% as shown in Table 7.3. The optimal chromosome output from this scenario is depicted by Figure 7.7, which has 10 intervals and 11 boundary values. Clearly, the optimal distance-based toll-charge function is a highly nonlinear one.

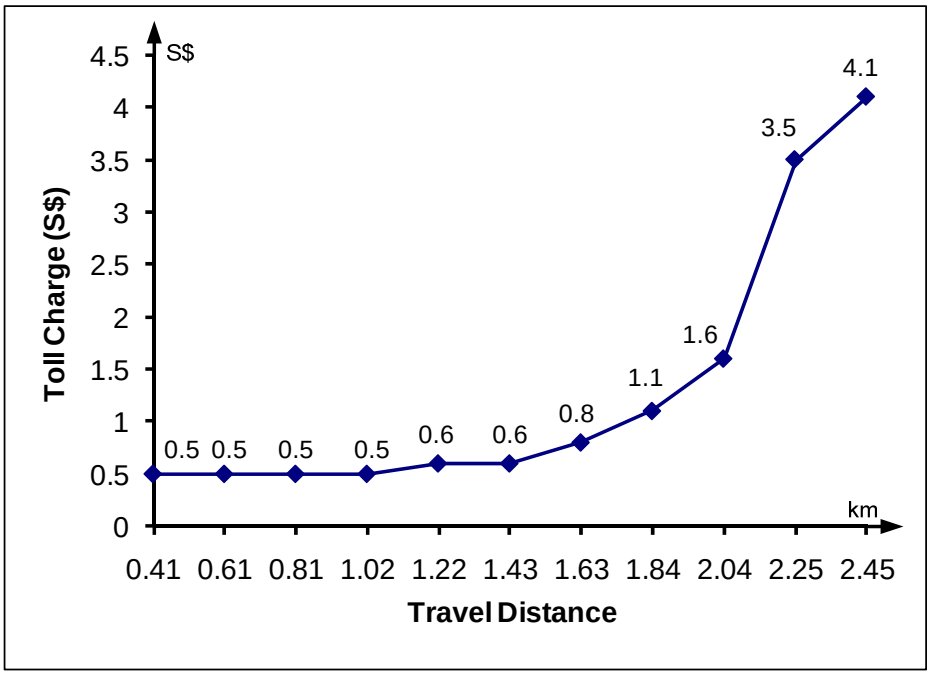

Figure 7.7 Optimal Toll-charge Function for Network C

As shown by Figure 7.7, the optimal nonlinear toll-charge function is a continuous non-decreasing function. Longer journeys in the cordon are charged up to eight times higher than the shorter journeys, which is more equal than the commonly adopted pay-per-entry and licensing basis charge. When implemented in practice, the equity property of distance-based charge makes it prone to be accepted by the public users. However, the nonlinear distance-based toll-charge function is comparatively more complicated in view that it has many different intervals. This complexity may cause some difficulties to the drivers in precisely perceiving the toll charge on each path and then in wisely making their route plan. To cope with this problem, some empirical policies executed in the initialization stage of many congestion pricing schemes can be followed. For instance, as used for the ERP system in Singapore (Foo, 2000), the authorities can (a) establish the nonlinear distance-based charge on partial pricing cordons and/or selected time of day as

a trial and (b) frequently launch some publicity campaign for propagation. Such policies would help the drivers/users to be aware of and familiar with the updated charging rules, which is quite positive for the user acceptance of nonlinear distance-based toll charge.

7.7 Conclusions

This chapter solved the optimal toll design problem for the distance-based toll-charge method of cordon-based congestion pricing scheme. A MPEC model was developed for the toll-charge function with maximal TSB, taking the fixed-point model for PA-SUEED problem with continuously distributed VOT as a constraint. The toll-charge function is assumed to be generic to any positive and non-decreasing functional form. To solve the MPEC model, a piecewise-linear approximation function was first utilized to approximate any feasible toll-charge function, which gives a mixed-integer MPEC model. Then a Hybrid GA was adopted to solve the mixed-integer MPEC model. The proposed methodology was numerically validated by a numerical example. The output optimal piecewise-linear toll-charge function is highly nonlinear and it can considerably improve the TSB compared with the un-tolled case and KM charge.

This study is taken on a *static* transportation network, while the congestion pricing schemes/trials in some cities (e.g. Singapore and Stockholm) are temporally dynamic and varying constantly at different time of day. It is thus necessary to extend the methodology proposed in this chapter for the time-differentiable toll charges using dynamic or multi-period traffic assignment approaches.

CHAPTER 8 CONCLUSIONS

8.1 Outcomes and Research Contributions

The research work in this dissertation firstly focused on theoretical studies of traffic assignment problem in the context of probit-based asymmetric stochastic user equilibrium with elastic demand (PA-SUEED). PA-SUEED was proven to possess better representativeness to the practical conditions, and meanwhile it is also generic to the cases of fixed demand or symmetric link travel time functions. Such a framework for traffic assignment thus has sound fitness to the practical conditions. Nevertheless, due to the complicated properties of PA-SUEED, it is computationally burdensome, in view that the only existing convergent solution algorithm (Cost-Averaging method) only has a sub-linear speed. It thereby prohibits the research as well as practical implementations of PA-SUEED. To improve the computational speed of solving PA-SUEED thus became the first objective of this dissertation.

Chapter 3 first reviewed about the existing formulations and solution algorithms for PA-SUEED problem. Then, two variational inequality (VI) models were proposed, which can be solved by two prediction-correction (PC) algorithms, named as FPC and CPC. These two PC algorithms, in nature, have linear convergent speed, thus can outperform the existing CA method. Yet, numerical experiments detected that when it approaches the optimal solution, the step search technique of the two PC algorithms was undermined. Therefore, two composite computational methods named as hybrid FPC-CA and hybrid CPC-CA were proposed. The numerical experiments showed that these two hybrid PC-

CA algorithms can improve the computational speed for six to ten times in different scenarios, in contrast with CA method.

Chapter 4 addressed the PA-SUEED with link capacity constraints, which is the most mathematical topic in this dissertation. Link capacity constraint is a rational and necessary extension to the conventional traffic assignment problems. Due to the complexities of PA-SUEED, the link capacity constraint problem addressed in this chapter is quite challenging to be solved. Based on a mathematical definition for this problem, a VI model was first proposed. Then, the monotonicity and Lipschitz-continuity of this VI model were rigorously proven. These properties of the VI model ensure the existence of solution to the VI model and also the convergence of a PC algorithm as a solution method. A numerical example was finally employed to test the proposed methodology.

Chapter 5 employed the distributed computing technique to accelerate the computational speed of solution algorithms for PA-SUEED. Solution algorithms for the PA-SUEED require solving the stochastic network loading (SNL) in each iteration, and a link-based two-stage Monte Carlo simulation method was proposed for the SNL. Despite a satisfactory accuracy level, the Monte Carlo simulation method is computational demanding. Yet, this method as well as its subroutines possesses a perfect parallelism. Thus, three distributed computing approaches were proposed for the workload partition of the SNL. A comprehensive numerical experiment indicated that the computational

speed of solution algorithms for PA-SUEED can be further accelerated for more than 50 times, by means of distributed computing.

The theoretical studies in Chapters 3 to 5 have provided a solid foundation for the formulations and algorithms for PA-SUEED problem. The efficient hybrid PC-CA algorithms as well as distributed computing approaches have inherently reduced its computational burden, making it suitable for practical implementations.

Based on the theory of user equilibrium, network flows come from the commuters' un-cooperative travel behavior. It is unlikely to achieve a rational and acceptable traffic condition, especially in the dense urban areas. Congestion pricing is one of the few instruments that can be used by network authorities to facilitate traffic demand management. It can adjust commuters' route choice behavior by changing their travel costs on different routes. Thus, congestion pricing is a good complement for the studies of traffic assignment. Notwithstanding a extensive literature for the studies of congestion pricing, the theoretical achievements have seldom been implemented in practice. In reality, nearly all the practical pricing schemes adopt the cordon-based congestion pricing, due to its convenience in construction, operation and supervision. Cordon-based congestion pricing scheme thus was taken as another target of this dissertation. Two timely topics with practical significance were accordingly addressed respectively in Chapter 6 and Chapter 7, named as speed-based and distance-based toll design. Continuously distributed value-of-time (VOT) was taken as another dimension for the contributions of the studies here.

The speed-based and distance-based toll design problems are both formulated as a mathematical programming with equilibrium constraints (MPEC), with the objective of maximizing total social benefit (TSB). The non-convexity of proposed MPEC model in conjunction with the random VOT makes it not available to apply any existing solution algorithm. Thus, a genetic algorithm-type method was taken to solve the toll design problem, computational speed of which is further accelerated by virtue of distributed computing. The proposed methodology for each toll design problem was then tested by the numerical experiment. It showed that when an arbitrary toll charge pattern is used, it would even reduce the value of TSB, compared with the un-tolled case.

8.2 Recommendations for Future Work

The theoretical achievements in this dissertation for the PA-SUEED problem as well as its application for the two congestion pricing topics are merely an initial step for the studies in this area. Future efforts are necessary and worthwhile to further extend these studies or to implement them in practice. Apart from those future research topics mentioned in the end of each chapter, some recommendations are provided here for a proportional of the most valuable research topics.

The two efficient hybrid PC-CA algorithms in Chapter 3 as well as the PC algorithm in Chapter 4 mostly rely on the achievements of He and Liao (2002) in view of its effectiveness and conciseness. However, some more sophisticated findings in the field of VI studies are detected (e.g., He et al., 2008), which can further improve the computational speed, compared with the method of He and Liao (2002). It is of

considerable interest to check the implementations of these finding for solving PA-SUEED problem, and numerically test the level of improvements.

Based on the achievements in Chapter 4, some other extensions to PA-SUEED with link capacity constraints are also reasonable and well known to the researchers, for instance, the multi-user-class problem, multi-vehicle-type problem (Daganzo, 1983), multi-modal transportation systems (Hamdouch et al., 2007), Park and Ride problem (Lam et al., 2006), etc.

Regarding the distributed computing, all the tests in this dissertation are performed in a local area network. Thus, even if the proposed distributed computing approaches are included in any commercial software, it may be difficult to run it on a personal computer. Although the algorithms can be solved in parallel on personal computers with dual or quad cores, superiority of the distributed computing approaches would be largely weaken, since they requires tens of processors. Thereby, it would be a promising topic to discuss about the implementations of the proposed approaches on some internet-based computing platform, e.g., the cloud computing.

The second half of this dissertation, Chapters 6 and 7, mainly targets at the practical-oriented congestion pricing schemes, which have certain advantages and conveniences for practical implementations. Some policy issues regarding the practical acceptance are also briefly discussed. From this angle, some other topics, helpful to the acceptance of congestion pricing by the public, are also necessary to be further investigated in the

context of PA-SUEED with continuously distributed VOT, including the assessing system for the fairness of pricing schemes, Pareto-improving schemes, etc.

Regarding the continuously distributed VOT, in the numerical tests of this study is assumed to be uniformly distributed. It is necessary to further test these examples in terms of a VOT following normal, exponential or lognormal distributions. More importantly, an in-depth research for the distribution of random VOT resorting to real survey data is considerably valuable.

REFERENCES

Aashtiani, H.Z., 1979. *The multi-modal traffic assignment problem*. Doctoral dissertation, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA.

Aashtiani, H.Z. and Magnanti T.L., 1981. Equilibria on a congested transportation network. SIAM J. *Algebraic and Discrete Mathematics*, 2, 213-226.

Abdulaal, M. and LeBlanc, L. J., 1979. Methods for combining modal split and equilibrium assignment models. *Transportation Science*, 13, 292-314.

Akiyama, T., Okushima, M., 2006. Implementation of cordon pricing on urban network with practical approach. *Journal of Advanced Transportation*, 40, 221-248.

Allsop, R.E., 1974. Some possibilities of using traffic control to influence trip distribution and route choice. *In*: Buckley, D.J. (ed.) *Proceedings of the 6th International Symposium on Transportation and Traffic Theory*, 345-375.

Asmuth, R.L., 1978. *Traffic network equilibria*. Technical Report SOL 78-2 (Stanford, CA: systems optimization laboratory, Stanford University).

Bard, J.F., 1998. *Practical Bilevel optimization: algorithms and applications.* Kluwer Academic Publishers, Dordrecht.

Bar-Gera, H., 2011, Transportation Network Test Problems Website. Accessed May 15th, 2011. http://www.bgu.ac.il/~bargera/tntp/.

Beckmann, M.J. and Golob, T.F., 1974. Traveler decision and traffic flows: a behavioral theory of network equilibrium. *Proceedings of the 5th International Symposium on the Theory of Traffic Flow and Transportation*, Sydney, August 26-28, D. J. Buckley, ed., Elsevier, New York, NY, 453-482.

Beckmann, M.J., McGuire, C.B. and Winsten, C.B., 1956. *Studies in Economics of Transportation*. Yale University Press, New Haven, Conn.

Bekhor, S. and Toledo, T., 2005. Investigating path-based solution algorithms to the stochastic user equilibrium problem. *Transportation Research Part B*, 39(3), 279-295.

Bell, M.G.H., 1991. The estimation of origin-destination matrices by constrained generalized least squares. *Transportation Research Part B*, 25, 13-22.

Bell, M.G.H., 1995a. Stochastic user equilibrium assignment in networks with queues. *Transportation Research Part B*, 29, 125-137.

Bell, M.G.H., 1995b. Alternatives to Dial's logit assignment algorithm. *Transportation Research Part B*, 29, 287-295.

Bell, M.G.H. and Iida, Y., 1977. *Transportation network analysis*. John Wiley & Sons.

Bellei, G., Gentile, G., Papola, N., 2002. Network pricing optimization in multi-user and multimodal context with elastic demand. *Transportation Research Part B*, 36, 779-798.

Ben-Ayed, O., Boyce, D.E., and Blair C.E., 1988. A general bilevel linear programming formulation of the network design problem. *Transportation Research Part B*, 22(4), 311-318.

Ben-Akiva, M., and Lerman, S.R., 1985. *Discrete choice analysis: theory and application to travel demand*. The MIT Press, Cambridge, Massachusetts, London, England.

Bertsekas, D.P., 1976. On the Goldstein-Levitin-Polyak gradient projection method. *IEE transactions on automatic control*, AC-21(2), 174-184

Bertsekas D.P. and Gafni E. M., 1982. Projection methods for variational inequalities with application to the traffic assignment problem, *Mathematical Programming Study*, 17, 139-159.

Binder, K. and D.W. Heermann, 1992. *Monte Carlo Simulation in Statistical Physics: An Introduction*. Springer-Verlag, Berlin.

Blum, J.R., 1954. Multidimensional stochastic approximation methods, *Annals of Mathematical Statistics,* 25, 737-744

Boyce, D.E., Janson, B.N. and Eash, R.W., 1981. The effect on equilibrium trip assignment of different link congestion functions. *Transportation Research Part A*, 15, 23-232.

Campbell, M.E., 1950. *Route selection and traffic assignment, tech. report*. Highway Research Board Correlation Service.

Cantarella, G.E., 1997. A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. *Transportation Science*, 31(2), 107-128.

Cantarella G.E., and Binetti M.G., 1998. Stochastic equilibrium traffic assignment with value-of-time distributed among user. *International Transactions of Operational Research*, 5(6), 541-553.

Carroll, J.D., 1959. A method of traffic assignment to an urban network, *Highway Research Board Bulletin*, 224, 64-71.

Cascetta, E., and Postorino, M.N., 2001. Fixed-point approaches to the estimation of O/D matrices using traffic counts on congested networks. *Transportation Science*, 35(2), 134-147.

Chen, M. and Bernstein, D. H., 2004. Solving the toll design problem with multiple user groups. *Transportation Research Part B*, 38, 61-79.

Chen, M., Alfa, A.S., 1991. Algorithms for solving Fisk‘s stochastic traffic assignment model. *Transportation Research Part B*, 25, 405 – 412.

Chen, A., Lo, H.K., Yang, H., 2001. A self-adaptive projection and contraction algorithm for the traffic assignment problem with path-specific costs. *European Journal of Operational Research,* 135, 27-41.

Chiou, S.W., 2005. Bilevel programming for the continuous transport network design problem. *Transportation Research Part B,* 39(4), 361-383.

Cipriani, E., Gori, S., and Petrelli, 2010. Transit network design: a procedure and an application to a large urban area. *Transportation Research Part C*, doi:10.1016/j.trc.2010.09.003.

Cochran, W.G., 1977. *Sampling techniques*. John Wiley & Sons, New York.

Connors, R.D., Sumalee, A. Watling, D.P., 2007. Sensitivity analysis of the variable demand probit stochastic user equilibrium with multiple user-classes. *Transportation Research Part B*, 41, 593-615.

Dafermos, S.C., 1971. An extended traffic assignment model with applications to two-way traffic. *Transportation Science,* 5, 366-389.

Dafermos, S.C., 1973. Toll patterns for multiclass-user transportation networks. *Transportation Science,* 7, 211-223.

Dafermos, S.C., 1980. Traffic equilibrium and variational inequalities. *Transportation Science,* 14, 42-54.

Dafermos, S.C., 1982. Relaxation algorithms for the general asymmetric traffic equilibrium problem. *Transportation Science,* 16, 231-240.

Dafermos, S.C., 1983. An iterative scheme for variational inequalities. *Mathematical Programming,* 26, 40-47.

Daganzo, C.F., 1979. *Multinomial Probit: the theory and its application to demand forecasting.* Academic Press, New York.

Daganzo, C.F., 1982. Unconstrained extremal formulation of some transportation equilibrium problems. *Transportation Scienc*e, 16, 332-360.

Daganzo, C.F., 1983. Stochastic network equilibrium with multiple vehicle types and asymmetric, indefinite link cost Jacobians, *Transportation Science,* 17, 282-300.

Daganzo, C.F. and Sheffi, Y., 1977. On stochastic models of traffic assignment. *Transportation Science*, 11(3), 253-274.

Demeulemeester, E. Dodin, B. and Herroelen, W., 1993. A random activity network generator. *Operations Research,* 41(5), 972-980.

Dial, R.B., 1971. A probabilistic multipath traffic assignment algorithm which obviates path enumeration. *Transportation Research*, 5, 83-111.

Dial, R.B., 1996. Bicriterion traffic assignment: basic theory and elementary algorithms. *Transportation Science*, 30(2), 93-111.

Dijkstra, E.W., 1959. A note on two problems in connection with graphs (spanning tree, shortest path). *Numerical Mathematics,* 1, 269-271.

Eliasson J., 2009. A cost-benefit analysis of the Stockholm congestion charging system. *Transportation Research Part A*, 43(4), 468-480.

Eliasson, J., Hultkrantz, L. Nerhagen, L. and Rosqvist, L.S., 2009. The Stockholm congestion - charging trial 2006: Overview of effects. *Transportation Research Part A*, 43(3), 240-250.

Esselink, K., Loyens, L.D.J.C., and Smit, B., 1995. Parallel Monte Carlo simulations. *Physical Review E*, 51(2), 1560-1568.

Facchinei, F. and Pang, J.-S., 2003. *Finite-dimensional variational inequalities and complementarity problems: volumes 1 &2*. Springer-Verlag.

Ferrari, P., 1995. Road pricing and network equilibrium. *Transportation Research Part B*, 29, 357-372.

Ferrari, P., 1997. Capacity constraints in urban transport networks. *Transportation Research Part B*, 31, 291-301.

Fishman, G.S. and Shaw, T.D., 1989. Evaluating reliability of stochastic flow networks. *Probability in the Engineering and Informational Sciences,* 3, 493-509.

Fisk, C., 1980. Some developments in equilibrium traffic assignment. *Transportation Research Part B*, 14, 243-255.

Fisk, C. and Nguyen, S. 1982. Solution algorithms for network equilibrium models with asymmetric user costs, *Transportation Science*, 16(3), 361-381.

Florian, M., and Gendreau, M., 2001. Applications of parallel computing in transportation. *Parallel Computing*, 27, 1521-1522.

Florian, M. and Spiess, H., 1982. the convergence of diagonalization algorithms for asymmetric network equilibrium problems. *Transportation Research Part B*, 16, 447-483.

Foo, T.S., 2000. An advanced demand management instrument in urban transport: electronic road pricing in Singapore. *Cities*, 17, 33-45.

Frank M. and Wolfe, P., 1956. An algorithm for quadratic programming, *Naval Research Logistics Quarterly*, 3, 53-65.

Friesz, T.L., Tobin, R.L., Cho, H.J., Mehta, N.J., 1990. Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, 48(2), 265-284.

Gen, T., and Cheng, R., 1997. *Genetic algorithms and engineering design*. John Wiley & Sons, Inc.

Gentile, J.E., 1998. *Random Number Generation and Monte Carlo Methods*. Springer Verlag.

Goldberg, D., 1989. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading, MA.

Gropp, W., Lusk, E., and Skjellum, A., 1999. *Using MPI: Portable parallel programming with the Message-Passing Interface*. The MIT Press.

Gunarsson, S.O., 1972. An algorithm for multipath traffic assignment, *PTRC Seminar Proceedings, Urban Traffic Model Research*.

Habbal, M., Koutsopoulos, H., Lerman, S., 1994. A decomposition algorithm for the all-pairs shortest path problem on massively parallel computer architectures. *Transportation Science*, 28(4), 292-308.

Hamdouch, Y., Florian, M., Hearn, D.W., and Lawphonganich, S., 2007. Congestion pricing for multi-modal transportation systems. *Transportation Research Part B*, 41(3), 275-291.

Han, D., Yang, H., 2009. Congestion Pricing in the Absence of Demand Functions. *Transportation Research Part E*. doi:10.1016/j.tre.2008.03.002.

He, B.S., 1992. A projection and contraction method for a class of linear complementarity problems and its application in convex quadratic programming, *Applied Mathematics and Optimization*, 25, 247-262.

He, B.S., 1994. A new method for a class of Linear Variational Inequalities, *Mathematical Programming*, 66, 137-144.

He, B.S., Li, M., Liao, L.-Z., 2008. An improved contraction method for structured monotone variational inequalities. *Optimization*, 57(5), 643-653.

He, B.S. and Liao L.Z., 2002. Improvements of some projection methods for monotone nonlinear variational inequalities. *Journal of Optimization Theory and Applications*, 112(1), 111-128.

Hearn, D. W. and Lawphongpanich, S., 1984. Convex programming formulations of the asymmetric traffic assignment problem. *Transportation Research Part B*, 18, 357-365.

Hearn, D.W. and Lawphongpanich, S., 1990. A dual ascent algorithm for traffic assignment problems. *Transportation Research Part B*, 24, 423-430.

Hearn, D.W. and Ribera, J., 1980. Bounded flow equilibrium problems by penalty methods. *Proceedings of IEEE International Conference on Circuits and Computers*, 1, IEEE, New York, 162-166.

Hensher, D.A. and Puckett, S., 2007. Assessing the influence of distance-based charges on freight transporters. *Transport Review*, 28(1), 1-19.

Hribar, M.R., Taylor, V.E., and Boyce, D.E., 2001. Implementing parallel shortest path for parallel transportation applications. *Parallel Computing*, 27, 1537-1568.

Huang, H.J. and Yang, H., 1996. Optimal variable road-use pricing on a congested network of parallel routes with elastic demand. In: *Transportation and Traffic Theory* (ed., Lessort J.B.). Elsevier Science, 479-500.

Inouye, H., 1987. Traffic equilibria and its solution in congested road networks. In: Genser, R. (Ed.), *Proceedings of IFAC Conference on Control in Transportations*, 267-272.

Johnson, R. A. and Wichern, D. W. 2002. *Applied Multivariate Statistical Analysis*, Fifth Edition. Prentice Hall, Upper Saddle River, New Jersey 07458.

Kennedy, A.D., 1999. The hybrid Monte Carlo algorithm on parallel computers. *Parallel Computing*, 25, 1311-1339.

Koh, A., Shepherd, S. Sumalee, A., 2009. Second best toll and capacity optimization in networks: solution algorithm and policy implications. *Transportation*, 36, 147-165.

Korpelevich, G.M., 1976. The extragradient method for finding saddle points and other problems, *Ekonomika Matematicheskie Metody*, 12, 747-756.

Kumar, V., Singh, V., 1991. Scalability of parallel algorithms for the all-pairs shortest-path problem. *Journal of parallel and distributed computing*, 13, 124-138.

Lam, W.H.K., Li, Z., Huang, H.J. and Wong, S.C., 2006. Modeling time dependent travel choice problems in road networks with multiple user classes and multiple parking facilities. *Transportation Research Part B*, 40(5), 368-395.

Langdon, M. G., 1984a. Improved algorithms for estimating choice probabilities in the multinomial probit model. *Transportation Science,* 18(3), 267-299.

Langdon, M. G., 1984b. Methods of determining choice probability in utility maximizing multiple alternative models. *Transportation Research Part B,* 18(3), 209-234.

Langmyhr, T., 2001. Learning from road pricing experience: introducing a second-generation road pricing system. *Planning Theory and Practice*, 2(1), 67-80.

Larsson, T. and Patriksson, M., 1995. An augmented Lagrangean dual algorithm for link capacity side constrained traffic assignment problems. *Transportation Research Part B*, 29(6), 433-455.

Larsson, T. and Patriksson, M., 1999. Side constrained traffic equilibrium models - analysis, computation and applications. *Transportation Research Part B*, 33(4), 233-264.

Lawphongpanich, S., Hearn, D.W., 1984. Simplicial decomposition of the asymmetric traffic assignment problem. *Transportation Research Part B*, 18, 123-133.

Lawphongpanich, S., Hearn, D.W., 2004. An MPEC approach to second best toll pricing. *Mathematical Programming*, 101B(1), 33-55.

Lawphongpanich S., Hearn, D.W., Smith, M. J., 2006. *Mathematical and Computational Models for Congestion Charging*, Springer.

LeBlanc, L.J., Morlok, E.K., Pierskalla, W.P., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research,* 9, 309-318.

Lerman, S.R., and Manski, C.F., 1978. An estimator for the generalized multi-normial probit choice model. *56th Annual TRB Meeting*, Washingto, D.C.

Leurent, F., 1993. Cost versus time equilibrium over a network. *European Journal of Operational Research*, 71, 205-221.

Leurent, F.M., 1995. Contributions to the logit assignment model, *Transportation Research Record,* 1493, 207-212 (Washington, DC: TRB, National Research Council).

Lewis, N.C., 1993. *Road pricing: theory and practice*. Thomas Telford, London.

Li, M.Z.F., 2002. The role of speed-flow relationship in congestion pricing implementation with an application to Singapore. *Transportation Research Part B*, 36, 731-754.

Link, H., 2008. Acceptability of the German charging scheme for heavy goods vehicles: empirical evidence from a freight company survey, *Transport Reviews*, 28(2), 141-158.

Liu, L.N. and McDonald, J.F., 1998. Efficient congest tolls in the presence of unpriced congestion: A peak and off-peak simulation model. *Journal of Urban Economics* 44, 352-366.

Liu, L.N. and McDonald, J.F., 1999. Economic efficiency of second-best congestion pricing schemes in urban highway systems. *Transportation Research Part B,* 33, 157-188.

Liu, Z., and Meng, Q., 2011. Distributed computing approaches for large-scale probit-based Stochastic User Equilibrium problems. *Journal of Advanced Transportation*, in press.

Liu, H.X., He., X and He, B., 2009. Method of successive weighted averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and spatial economics*, 9, 485-503.

Lo, H.K., and Chen, A., 2000. Traffic equilibrium problem with route-specific costs: formulation and algorithms. *Transportation Research Part B*, 34, 493-513.

Lo, H.K., Yip, C. W. and Wan, K. H., 2003. Modeling transfer and non-linear fare structure in multimodal network. *Transportation Research Part B*, 37, 149-170.

London Congestion Charging Website. www.cclondon.com.

Luo, Z.Q., Pang, J.S. and Ralph, D., 1996. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, New York.

Ma, C.-M., 1994. Implementation of a Monte Carlo code on a parallel computer system. *Parallel Computing*, 20, 991-1005.

Maher, M.J., 1983. Inferences on trip matrices from observations on link volumes: A Bayesian statistical approach. *Transportation Research Part B*, 17, 435-447.

Maher, M., 2001. SUEED – stochastic user equilibrium assignment with elastic demand. *Traffic Engineering & Control*, 42(5), 163-167

Maher M.J. and Hughes P.C., 1997a. An algorithm for sueed-stochastic user equilibrium with elastic demand. *8th IFAC Symposium on Transportation Systems*, Chania, Crete, June, 1997.

Maher M.J. and Hughes P.C., 1997b. A probit-based stochastic user equilibrium assignment model. *Transportation Research Part B*, 31(4), 341-355.

Maher M.J. and Hughes P.C., 1998. The stochastic user equilibrium assignment problem with elastic demand – a comparison of new approaches. *6th Meeting of the EURO Working Group on Transportation*, 9-11 September 1998, Gothenberg, Sweden.

Maher M.J., Hughes P.C. and Kim K-S., 1999. New algorithms for the solution of the stochastic user equilibrium assignment problem with elastic demand. *14th International Symposium on Transportation and Traffic Theory,* Jerusalem, July 1999.

Maher, M., Stewart, K. and Rosa, A., 2005. Stochastic social optimum traffic assignment. *Transportation Research Part B*, 39, 753-767.

Maher M.J. and Zhang X., 2000. Formulation and algorithms for the problem of stochastic user equilibrium assignment with elastic demand. *8th EURO Working Group meeting on Transportation*, Rome, September 2000.

Marchand, M., 1968. A note on optimal tolls in an imperfect environment. *Econometrica*, 36, 575-581.

Maruyama, T. and Sumalee, A., 2007. Efficiency and equity comparison of cordon- and area-based road pricing schemes using a trip-chain equilibrium model. *Transportation Research Part A*, 41, 655-671.

May, A.D., Shepherd, S.P., Sumalee, A. and Koh, A, 2008. Chapter 7 of the Book: *Road congestion pricing in Europe – implications for the United States*, edited by Richardson, H.W., and Bae, C-H. C., 138-155.

May, A.D., Liu, R., Shepherd, S.P., and Sumalee, A., 2002. The impact of cordon design on the performance of road pricing schemes. *Transport Policy,* 9, 209-220.

May, A.D., and Milne, D.S., 2000. Effects of alternative road pricing systems on network performance. *Transportation Research Part A*, 34(6), 407-436.

Mayet, J. and Hansen, M., 2000. Congestion pricing with continuously distributed values of time. *Journal of Transport Economics and Policy*, 34, 359-370.

Mendell, N.R. and Elston R.C., 1974. Multifactorial qualitative traits: genetic analysis andprediction of recurrence risks. *Biometrics* 30, 41-57.

Meng, Q., Lam, W. H. K., and Yang, L., 2008. General stochastic user equilibrium traffic assignment problem with link capacity constraints. *Journal of Advanced Transportation*, 42, 429-465.

Meng, Q., Lee, D.-H., Cheu, R.L., Yang, H., 2004. Logit-based stochastic user equilibrium problem for entry-exit toll schemes. *Journal of Transportion Engineering-ASCE*, 130(6), 805-813.

Meng, Q., and Liu, Z., 2011. Mathematical models and computational algorithms for probit-based asymmetric stochastic user equilibrium problem with elastic demand. *Transportmetrica,* DOI:10.1080/18128601003736026.

Meng, Q., Yang, H. and Bell, M.G.H., 2001. An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research Part B,* 35, 83-105.

Miller, S.D., Payne, H.J. and Thompson, W.A., 1975. An algorithm for traffic assignment on capacity constrained transportation networks with queues. Paper presented at the Johns Hopkins Conference on Information Science and System, The Johns Hopkins University, Baltimore, MD, April 2-4, 1975.

Mitchell, G., Namdeo, A. and Milne, D., 2005. The air quality impact of cordon and distance based road user charging: an empirical study of Leeds, UK. *Atmospheric Environment*, 39, 6231-6242.

Moatti, A., Goldberg, J. and Memmi, G., 1987. Parallel Monte Carlo calculations with many microcomputers. *Computer Physics Communication,* 45, 355-359.

Moore, E.F., 1957. The shortest path through a maze, in: *Proceedings of an International Symposium on the Theory of Switching*, Harvard University, 285-292.

Nagel, K., and Rickert, M., 2001. Parallel implementation of the TRANSIMS micro-simulation. *Parallel Computing*, 27, 1161-1639.

Nagurney, A., 1993. *Network economics: a variational inequality approach*. Kluwer Academic publishers, Dordrecht, Boston, London.

Nagurney, A., and Zhang, D., 1996. *Projected Dynamical Systems and Variational Inequality with Applications*, kluwer Academic Publishers, Dordrecht, Boston, MA.

Namdeo, A. and Mitchell, G., 2008. An empirical study of estimating vehicle emissions under cordon and distance based road user charging in Leeds, UK, *Environ Monit Assess*, 136, 45-51.

Nguyen, S., and Pallotino, S., 1988. Equilibrium traffic assignment in large scale transit networks. *European Journal of Operational Research*, 37 (2), 176 – 186.

Nie. Y., Zhang, H.M. and Lee, D.H., 2004. Models and algorithms for the traffic assignment problem with link capacity constraints. *Transportation Research Part B*, 38(4), 285-312.

Nie. Y., and Liu, Y., 2010. Existence of self-financing and Pareto-improving congestion pricing: impact of value of time distribution. *Transportation Research Part B*, 44, 39-51.

O'Cearbhaill, E.A. and O'Mahony, M., 2005. Parallel implementation of a transportation network model. *Journal of Parallel and Distributed Computing*, 65, 1-14.

Ohno, H., Suzuki, T., Yamaguchi, Y., Okamoto, S. and Iizuka, K., 2007. Development of the next generation road pricing system with GPS technology. *Technical Review*, 44(2), 1-5.

Olszewski, P., Fan, H.S.L. and Tan, Y.-W., 1995. Area-wide traffic speed-flow model for the Singapore CBD. *Transportation Research Part A*, 29(4), 273-281.

Olszewski, P. and Xie, L., 2005. Modelling the effects of road pricing on traffic in Singapore. *Transportation Research Part A*, 39, 755-772.

O'Mahony, M., Geraghty, D., and Humphreys, I., 2000. Distance and time based road pricing trial in Dublin. *Transportation*, 27, 269-283.

Ortega, J.M. and Rheinboldt, W.C., 1970. *Iterative solution of nonlinear equations in several variables*. Academic Press, New York and London.

Ortuzar, J.D, and Willumsen, L.G., 1995. *Modelling transport*. John Wiley & Sons Ltd.

Outrata, J.V., Kocvara, M. and Zowe, J., 1998. *Nonsmooth approach to optimization problems with equilibrium constraints.* Kluwer Academic Publishers, Dordrecht.

Panicucci, B., Pappalardo, M., Passacantando, M., 2007. A path-based double projection method for solving the asymmetric traffic network equilibrium problem. *Optimization Letters*, 1, 171-185.

Pape, U., 1974. Implementation and efficiency of Moore-algorithms for the shortest route problem. *Mathematical Programming,* 7, 212-222.

Patriksson, M., 1994a. *The traffic assignment problems: models and methods*. VSP, Utrecht, the Netherlands.

Patriksson, M., 1994b. On the convergence of descent methods for monotone variational inequalities*. Operations Research Letters*, 16(5), 265-269.

Patriksson, M., 2004. Sensitivity analysis of traffic equilibria. *Transportation Science*, 38, 258-281.

Phang, S.-Y., and Toh, R.S., 1997. From manual to electronic road congesting pricing: the Singapore experience and experiment. *Transportation Research Part E*, 33 (2), 97-106.

Pigou, A.C., 1920. *The economics of welfare*. MacMillan, London.

Pollack, M. and Wiebenson, W., 1960. Solutions for the shortest route problem. *Operations Research*, 8, 224-230.

Powell, W. B. and Sheffi, Y., 1982. The convergence of equilibrium algorithms with predetermined step sizes. *Transportation Science*, 16, 45-55.

Prashker, J.N. and Bekhor, S., 2004. Route choice models used in the stochastic user equilibrium problem: a review. *Transport Reviews,* 24, 437-436

Richards, M., Gilliam, C., Larkinson, J., 1996. The London congestion charging research programme: 6. The findings. *Traffic Engineering and Control* 37 (7/8), 436–440.

Rosa, A. and Maher, M.J., 2002. Algorithms for solving the probit path-based SUE traffic assignment problem with one or more user classes. *Transportation and Traffic Theory in the 21st Century. Proceedings of the 15th ISTTT,* 371-392

Rubinstein, R.Y., 1981. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York.

Santos, G., 2008. The London congestion charging scheme. Chapter 8 of the Book: *Road congestion pricing in Europe – implications for the United States*, edited by Richardson, H.W., and Bae, C-H. C., 159-175.

Sheffi, Y. 1985. *Urban transportation networks: equilibrium analysis with mathematical programming models*. Prentice-Hall, INC, Englewood Cliffs, New Jersey.

Sheffi, Y. and Powell, W.B., 1982. An algorithm for the equilibrium assignment problem with random link times. *Networks,* 12(2), 191-207.

Shepherd, S. and Sumalee, A., 2004. A genetic algorithm based approach to optimal toll level and location problems. *Network & Spatial Economics,* 4, 161-179.

Shiftan, Y., Button, K., and Nijkamp, P., 2007. *Transportation planning*. An Elgar Reference Collection.

Shimizu, K., Ishizuka,Y. and Bard, J.F., 1997. *Nondifferentiable and two-level mathematical programming.* Kluwer Academic Publishers, Boston.

Singleton, G.L, Wu, C.-H. and Tsai, J.-H., 1991. A parallel Monte Carlo simulation for gaseous electronics on a dynamically reconfigurable multiprocessor system. *Computer Physics Communication*, 66, 181-193.

Singapore Land Transport Authority's Official Website (Accessed February 11, 2011): www.lta.gov.sg/motoring_matters/index_motoring_erp.htm.

Small, K.A., 1992. *Urban transportation economics*. Harwood Academic.

Small, K.A., Winston, C. and Yan, J., 2005. Uncovering the distribution of motorists' preferences for travel time and reliability. *Econometrica*, 73(4), 1367-1382.

Smith, M.J., 1979. The existence, uniqueness and stability of traffic equilibrium. *Transportation Research Part B*, 13, 295-304.

Smith, M. J., 1987. Traffic control and traffic assignment in a signal-controlled network with queueing. *Proceedings of the 10th Symposium on Transportation and Traffic Theory*, MIT, 61-77.

Sumalee, A., 2004. *Optimal Road Pricing Scheme Design*. Ph.D. Dissertation, Institute for Transportation Studies, the University of Leeds.

Sumalee, A., 2007. Multi-concentric optimal charging cordon design. *Transportmetrica*, 3, 41-71.

Sumalee A., May A.D., and Shepherd, S.P., 2005. Comparison of judgmental and optimal road pricing cordons, *Journal of Transport Policy*, 12(5), 384-390.

Sumalee, A., and Xu, W., 2011. First-best marginal cost toll for a traffic network with stochastic demand. *Transportation Research Part B*, 45, 41-59.

Suwansirikul, C. Friesz, T.L., Tobin, R.L., 1987. Equilibrium decomposed optimization : a heuristic for the continuous network equilibrium design problem. *Transportation Science*, 21(4), 254-263.

Tang, L.K. and Melchers, R.E., 1987. Improved approximation for multinormal integral. *Structural Safety,* 4, 81-93.

Thompson, W.A., 1976. *Traffic assignment for capacitated transportation networks including queuing with application to freeway corridor control*. PhD thesis, Department of Electrical Engineering, University of Southern California, Los Angeles, CA.

Tobin, R.L., 1977. An extension of Dial's algorithm utilizing a model of tripmakers' perceptions, *Transportation Research*, 11, 337–342.

Traff, J.L., 1995. An experimental comparison of two distributed single-source shortest path algorithms. *Parallel Computing*, 21, 1505-1532.

Traynor, C.A. and Anderson, J.B., 1988. Parallel Monte Carlo calculations to determine energy differences among similar molecular structures. *Chemical Physics Letters,* 147(4), 389-394.

Tremblay, N. and Florian, M., 2001. Temporal shortest paths: parallel computing implementations. *Parallel Computing*, 27, 1569-1609.

Ubbels, B., Rietveld, P. and Peeters, P., 2002. Environmental effects of a kilometer charge in road transport: an investigation for the Netherlands. *Transportation Research Part D*, 7, 255-264.

Van den Berg, V., Verhoef, E.T., 2011. Congestion tolling in the bottleneck model with heterogeneous values of time. *Transportation Research Part B*, 45, 60-78.

Van Vliet, D., 1978. Improved shortest path algorithms for transport networks. *Transportation Research*, 12, 7-20.

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,* 36, 707-729.

Verhoef, E.T., Bliemer, M., Steg, L., and Wee, B., 2008. *Pricing in road transport: a multi-disciplinary perspective*. Edward Elgar, Cheltenham, UK, Northampton, MA, USA.

Verhoef, E.T., Nijkamp, P. and Rietveld, P., 1996. Second-best congestion pricing: The case of an untolled alternative. *Journal of Urban Economics,* 40, 279-302.

Verhoef, E.T., Rowendal, J., 2004. Pricing, capacity choice and financing in transportation networks. *Journal of Regional Science*, 44(3), 405-435.

Verhoef, E.T., Small, K.A., 2004. Product differentiation on roads: constrained congestion pricing with heterogeneous users. *Journal of Tranport Economics Policy*, 38(1), 127-156.

Walters, A.A., 1961. The theory and measurement of private and social cost of highway congestion. *Econometrica* 29, 676-699.

Wardrop, J.G., 1952. Some theoretical aspects of road traffic research. *Proceedings, Institution of Civil Engineers* II (1), 325-378.

Watling, D., 1998. Perturbation stability of the asymmetric stochastic equilibrium assignment model. *Transportation Research Part B*, 32, 155-171.

Watling, D., 2006. User equilibrium traffic network assignment with stochastic travel times and late arrival penalty. *European Journal of Operational Research*, 175, 1539-1556.

Williams, H.C.L., 1977. On the formation of travel demand models and economic evaluation measures of user benefits, *Environment and Planning*, A9, 285-344.

Wilson, P.W., 1988. Welfare effects of congestion pricing in Singapore. *Transportation*, 15, 191-210.

Wong, S.C. and Yang, H., 1997. Reserve capacity of a signal controlled network. *Transportation Research Part B*, 31, 303-314.

Wong, S.C., Wong, C.K. and Tong, C.O., 2001. A parallelized genetic algorithm for the calibration of Lowry model. *Parallel Computing*, 27, 1523-1536.

Wood, J., Al-Bahadili, H. and Khaddaj, S.A., 1991. A comparison of Monte Carlo photon transport on two types of parallel computer. *Annuals of Nuclear Energy,* 18(3), 155-166.

Xiao, F., and Yang, H., 2008. Efficiency loss of private road with continuously distributed value-of-time. *Transportmetrica*, 4, 19-32.

Xu, W., 2006. *Development of practical implementation methods for road pricing*. Ph.D Dissertation, Department of Civil Engineering, The Hong Kong University of Science and Technology.

Yang, H., 1995. Heuristic algorithms for the bilevel origin-destination matrix estimation problem. *Transportation Research Part B*, 29, 231-244.

Yang, H., 1997. Sensitivity analysis for the elastic-demand network equilibrium problem with applications. *Transportation Research Part B*, 31(1), 55-70.

Yang, H. and Bell, M.G.H., 1997. Traffic restraint, road pricing and network equilibrium. *Transportation Research Part B,* 31(4), 303-314.

Yang, H. and Bell, M.G.H., 1998. Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, 18(3), 257-278.

Yang, H. and Huang, H.J., 1998. Principle of marginal-cost pricing: How does it work in a general network? *Transportation Research Part A,* 32, 45-54.

Yang, H. and Huang, H.J., 2005. *Mathematical and economic theory of road pricing*. Elsevier Ltd.

Yang, H. and Lam, W.H.K., 1996. Optimal road tolls under conditions of queuing and congestion. *Transportation Research Part A*, 30(5), 319-332.

Yang, H., Meng, Q. and Lee, D.-H., 2004. Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions. *Transportation Research Part B*, 38, 477-493.

Yang, H., Sasaki, T., Iida Y. and Asakura, Y., 1992. Estimation of origin-destination matrices from link traffic counts on congested networks, *Transportation Research Part B*, 26, 417-434.

Yang, H., and Yagar, S., 1995. Traffic assignment and signal control in saturated road networks. *Transportation Research Part A*, 29, 125-139.

Yang, H., Zhang, X.N., and Meng, Q., 2004. Modeling private highways in networks with entry-exit based toll charges. *Transportation Research Part B*, 38, 191-213.

Zhang X.N. and Yang, H., 2004. The optimal cordon-based network congestion pricing problem. *Transportation Research Part B*, 38, 517-537.

Zhao, Y. and Kockelman, K.M., 2006. On-line marginal-cost pricing across networks: incorporating heterogeneous users and stochastic equilibria. *Transportation Research Part B*, 40, 424-435.

Zhao, C. and Wood, J., 1989. The Monte Carlo method on a parallel computer. *Annals of Nuclear Energy,* 16(2), 649-657.

LIST OF PUBLICATIONS

Journal Papers

1. Meng, Q. and **Liu, Z.**, 2011. Mathematical models and computational algorithms for probit-based asymmetric stochastic user equilibrium problem with elastic demand, *Transportmetrica*, DOI:10.1080/18128601003736026.
2. **Liu, Z.** and Meng, Q., 2011. Distributed computing approaches for large-scale probit-based stochastic user equilibrium problem, *Journal of Advanced Transportation,* DOI: 10.1002/atr.177.
3. Meng, Q. and **Liu, Z.**, 2011. Trial-and-Error Method for Congestion Pricing Scheme under Side-Constrained Probit-Based Stochastic User Equilibrium Conditions, *Transportation*, 38, 819-843.
4. Meng, Q. and **Liu, Z.**, 2012. Impact Analysis of Cordon-based Congestion Pricing Scheme on Mode-Split of Bimodal Transportation Network, *Transportation Research Part C*, 21, 134-147.
5. Meng, Q., Wang, S. and **Liu, Z.**, 2011. Large-Scale Intermodal Liner Shipping Service Network Design, *Transportation Research Record*, accepted for publication.
6. Meng, Q. and **Liu, Z.**, 2011. Optimal distance-based toll design for cordon-based congestion pricing scheme with continuously distributed value-of-time, under revision with *Transportation Research Part B.*
7. **Liu, Z.** and Meng, Q., 2010. Modeling transit-based park-and-ride services on a multimodal network with congestion pricing schemes, under review with *Transportation Research Part C.*

8. Meng, Q. and **Liu, Z.**, 2010. Asymmetric stochastic user equilibrium problem with link capacity constraints and elastic demand, under review with *European Journal of Operational Research*.
9. Meng, Q. and **Liu, Z.**, 2011. Speed-based Toll Design for Cordon-Based Congestion Pricing Scheme, under review with *Transportation Research Part E*.

Conference Papers

1. Meng, Q. and **Liu, Z.**, 2010. Trial-and-error method for cordon-based congestion pricing scheme with probit-based stochastic user equilibrium constraints. *Proceeding of the 12th World Conference on Transport Research*, Lisbon, Portugal, July 11-15.
2. **Liu, Z.** and Meng, Q., 2010. A reformulation of truck and trailer routing problem. *Proceeding of the 3rd International Conference on Transportation and Logistics*, Fukuoka, Japan, September 6-8.
3. **Liu, Z.** and Meng, Q., 2010. Probit-based stochastic user equilibrium problem: is it computationally acceptable? *Proceeding of the 15th HKSTS International Conference*, Hong Kong, China, December 11-14.
4. Liu, Z. and Meng, Q., 2011. Nonlinear Congestion Pricing: Model Development and Distributed Algorithm Design. At *AIT-ITB-NUS-KU symposium*, Bangkok, Thailand, November, 17-18.
5. Liu, Z. and Meng, Q., 2011. Toll Adjustment for Cordon-Based Congestion Pricing Scheme Using Traffic Counts. *Proceeding of the 16th HKSTS International Conference*, Hong Kong, China, December 17-20.