Dynamic Congestion Pricing in Urban Networks with the Network Fundamental Diagram and Simulation-Based Dynamic Traffic Assignment

Ziyuan Gu

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

UNSW SYDNEY

School of Civil and Environmental Engineering

Faculty of Engineering

April 2019

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname/Family Name	:	Gu
Given Name/s	:	Ziyuan
Abbreviation for degree as given in the University calendar	:	CVEN1630
Faculty	:	Faculty of Engineering
School	:	School of Civil and Environmental Engineering
Thesis Title	:	Dynamic Congestion Pricing in Urban Networks with the Network Fundamental Diagram and Simulation-Based Dynamic Traffic Assignment

Abstract

This thesis focuses on modeling and optimization of two-region urban pricing systems and analyzing and understanding the effects of pricing on the network traffic flow. The motivation of this work is the fact that traffic congestion is growing fast in cities around the world especially in city centers, and hence the need for an effective and efficient travel demand management (TDM) policy. With the aim of advancing the current congestion pricing theory, this thesis proposes and integrates different advanced pricing regimes with the concept of the Network Fundamental Diagram and a simulation-based dynamic traffic assignment (DTA) model, studies and compares different computationally efficient simulation optimization (SO) methods, and analyzes and understands the effects of different pricing regimes on the network traffic flow.

This thesis demonstrates through computer simulations the effectiveness of a well-designed pricing system on improving the network performance. The major finding is that the distance only toll, which represents the state of the practice, naturally drives travelers into the shortest paths within the pricing zone (PZ) resulting in a more uneven distribution of congestion and hence, a larger hysteresis loop in the NFD and lower network flows especially during network recovery. This limitation is overcome by two more advanced pricing regimes, namely the joint distance and time toll (JDTT) and the joint distance and delay toll (JDDT), through the introduction of either a time or a delay toll component. Moreover, this thesis explicitly models and minimizes the heterogeneity of congestion distribution as part of the toll level problem (TLP). The toll area problem (TAP) is also investigated by means of network partitioning.

To optimize different pricing regimes through computer simulations, this thesis develops two computationally efficient SO frameworks. The first framework employs a proportional-integral (PI) controller from control theory to solve a simple TLP featuring a low-dimensional decision vector, a set-point objective and only bound constraints. The second framework employs regressing kriging (RK) from machine learning to solve a complex TLP that has either a high-dimensional decision vector, a complex objective, or a set of complex constraints. A comprehensive comparison between the two methods and two other widely used methods, namely simultaneous perturbation stochastic approximation (SPSA) and DIviding RECTangles (DIRECT), are performed.

Overall, this thesis provides valuable insights into the study, design, and implementation of urban pricing systems and the effects of pricing on the network traffic flow. Results of this work not only help in developing effective pricing systems to mitigate urban traffic congestion, but also provide competitive solutions to other types of network design problems (NDPs).

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed …….....

Date…….....

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The student contributed greater than 50% of the content in the publication and is the "primary author", i.e. the student was responsible primarily for the planning, execution and preparation of the work for publication
- The student has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

Please indicate whether this thesis contains published material or not.

☐ *This thesis contains no publications, either published or submitted for publication.*

☒ *Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement.*

☐ *This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below.*

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Name	**Signature**	**Date (dd/mm/yy)**
Ziyuan Gu		

ABSTRACT

This thesis focuses on modeling and optimization of two-region urban pricing systems and analyzing and understanding the effects of pricing on the network traffic flow. The motivation of this work is the fact that traffic congestion is growing fast in cities around the world especially in city centers, and hence the need for an effective and efficient travel demand management (TDM) policy. With the aim of advancing the current congestion pricing theory, this thesis proposes and integrates different advanced pricing regimes with the Network Fundamental Diagram (NFD) and simulation-based dynamic traffic assignment (DTA), studies and compares different computationally efficient simulation-based optimization (SO or SBO) methods, and analyzes and understands the effects of different pricing regimes on the network traffic flow.

This thesis demonstrates through computer simulations the effectiveness of a well-designed pricing system on improving the network performance. The major finding is that the distance only toll, which represents the state of the practice, naturally drives travelers into the shortest paths within the pricing zone (PZ) resulting in a more uneven distribution of congestion and hence, a larger hysteresis loop in the NFD and lower network flows especially during network recovery. This limitation is overcome by two more advanced pricing regimes, namely the joint distance and time toll (JDTT) and the joint distance and delay toll (JDDT), through the introduction of a time and a delay toll component, respectively. Moreover, this thesis explicitly models and minimizes the heterogeneity of congestion distribution as part of the toll level problem (TLP). The toll area problem (TAP) is also investigated by means of network partitioning.

To optimize different pricing regimes through computer simulations, this thesis develops two computationally efficient SO frameworks. The first framework employs a

proportional-integral (PI) controller from control theory to solve a simple TLP featuring a low-dimensional decision vector, a set-point objective and only bound constraints. The second framework employs regressing kriging (RK) from machine learning to solve a complex TLP that has either a high-dimensional decision vector, a complex objective, or a set of complex constraints. A comprehensive comparison between the two methods and two other widely used methods, namely simultaneous perturbation stochastic approximation (SPSA) and DIviding RECTangles (DIRECT), are performed.

Overall, this thesis provides valuable insights into the study, design, and implementation of urban pricing systems and the effects of pricing on the network traffic flow. Results of this work not only help in developing effective pricing systems to mitigate urban traffic congestion, but also provide competitive solutions to other types of network design problems (NDPs).

LIST OF PUBLICATIONS

Gu, Z., Waller, S.T., Saberi, M., 2018. Surrogate-based toll optimization in a large-scale heterogeneously congested network. *Comput.-Aided Civ. Inf. Eng.*, 1-16.

Gu, Z., Shafiei, S., Liu, Z., Saberi, M., 2018. Optimal distance- and time-dependent area-based pricing with the Network Fundamental Diagram. *Transp. Res. Part C* 95, 1-28.

Gu, Z., Saberi, M., Sarvi, M., Liu, Z., 2018. A big data approach for clustering and calibration of link fundamental diagrams for large-scale network simulation applications. *Transp. Res. Part C* 94, 151-171.

Gu, Z., Liu, Z., Cheng, Q., Saberi, M., 2018. Congestion pricing practices and public acceptance: A review of evidence. *Case Stud. Transp. Policy* 6(1), 94-101.

Shafiei, S., **Gu, Z.**, Saberi, M., 2018. Calibration and Validation of a Simulation-based Dynamic Traffic Assignment Model for a Large-Scale Congested Network. *Simul. Modell. Pract. Theory* 86, 169-186.

Gu, Z., Saberi, M., 2019. Continuous simulation-based optimization of expensive black-box traffic systems: A comparative review of algorithms and application to toll pricing. *Transp. Res. Part B*, under review.

Gu, Z., Saberi, M., 2019. A bi-partitioning approach to congestion pattern recognition and toll area identification. *Transp. Res. Part C*, under revision.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank myself for being persistent and hard-working all the way to this point of my life. I also want to thank my parents, Zhenggui and Li, for their ever-present love and support. I'm especially grateful for having my beautiful wife, Qinying, with me during all these years.

I would like to express my sincere gratitude to my mentor Meead Saberi for all the advice, knowledge, support, and encouragement he has offered to make this happen. I'm also thankful to Zhiyuan Liu, Travis Waller, and Majid Sarvi for serving in my supervisory team and providing valuable comments and support. I want to acknowledge Mohsen Ramezani, Nan Zheng, Nicholas Geroliminis, Geoffrey Rose, and Inhi Kim for helping me in various ways and their inspiring knowledge.

I want to thank the supporting staff from the Research Center for Integrated Transport Innovation (rCITI), UNSW and the Institute of Transport Studies (ITS), Monash University for always being there for me. Thank you, Maria, Pattie, Jenny, and Min.

Finally, I feel grateful to have all my friends from both universities who have made my life in the past few years much more colorful and meaningful. Thank you, Sajjad, Richard, Wentao, Reza, Xinyuan, Xiaoying, Xiang, Chenyang, Chris, Long, Milad, Chence, Tanapon, and Mudabber. I'm also thankful to Yang, Yicong, Jiawei, Qingfeng, and Xiangnan for our everlasting friendships.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

CHAPTER 1. INTRODUCTION

With rapid population and employment growth, traffic congestion in major cities around the world is expected to worsen causing significant economic and productivity loss. For example, the congestion cost in Melbourne, Australia is projected to reach $10.2 billion in 2030, an increase from $4.6 billion in 2015 (Bureau of Infrastructure, Transport and Regional Economics (BITRE), 2015). While reducing travelers' comfort during their trips, congestion also undermines the livability of the city as the city space is largely occupied by cars producing substantial vehicle emissions that affect the environmental quality and public health.

The congestion problem is not new, and has always been an active area of research. Building and expanding road infrastructure, as a traditional approach to reducing congestion, is financially and environmentally unsustainable and, more importantly, both theory and practice have shown that it may result in a return to congestion due to the induced demand (Sheffi, 1985). Therefore, demand-oriented strategies or travel demand management (TDM) policies have been widely advocated and implemented as promising solutions to the congestion problem, one of which is congestion pricing. Unlike gating or perimeter control, congestion pricing originates from the economic theory and hence serves as an economic lever to influence traffic. The rationale behind the concept is to internalize road users' travelling impacts on others which they are either unaware of or unwilling to consider, commonly known as externalities of trips (Yang and Huang, 2005).

To date, congestion pricing has been successfully implemented in different parts of the world including Singapore, London, Stockholm, and Milan, and is being considered as an initiative in a few other metropolises as well. See Gu et al. (2018) for a recent review. However, along with the increasing interest in implementing congestion pricing comes

the inadequacy of theoretically sound optimization methods particularly for a large-scale network with sophisticated pricing regimes, a fact that perhaps explains in part why one could hardly find any documentation elaborating on how the existing pricing systems were determined in the first place as well as how they were and will be updated. Singapore's Electronic Road Pricing (ERP) system is an exception where the toll price is adjusted quarterly to achieve a speed target (Olszewski and Xie, 2005)[1]. While the classical theory on congestion pricing is well established, e.g. see Yang and Huang (2005) for a comprehensive overview of first- and second-best pricing, its application entails great details of the network in question including traffic data of each individual link and knowledge of the origin-destination (OD) demand. This demanding requirement prevents the so-called microscopic approach from being applied efficiently to a large-scale network, thereby necessitating a macroscopic approach.

Macroscopic traffic flow relations for an urban network were initially proposed by Godfrey (1969) followed by Daganzo (2007); Mahmassani et al. (1987); Olszewski et al. (1995). More recently, with the re-theorization of the Network Fundamental Diagram (NFD) or Macroscopic Fundamental Diagram (MFD) using field data from Yokohama, Japan (Geroliminis and Daganzo, 2008), a new branch of congestion pricing theory has been enabled that largely facilitates the design and implementation of a large-scale pricing system due to its macroscopic nature.

This thesis focuses on advancing the newly established macroscopic approach by proposing and integrating more efficient and equitable pricing regimes with computationally efficient simulation optimization (SO) methods. Through computer simulations, we have demonstrated the capabilities of the proposed pricing optimization frameworks in

[1] More information can be found at the Land Transport Authority website: https://www.lta.gov.sg/content/ltaweb/en/roads-and-motoring/managing-traffic-and-congestion/electronic-road-pricing-erp.html.

driving the network to its optimal state as well as reducing the heterogeneity of congestion distribution. Results of this work not only help in developing effective pricing systems to mitigate urban traffic congestion, but also provide competitive solutions to other types of network design problems (NDPs).

The rest of this chapter is organized as follows. Section 1.1 states the problem to be solved in this thesis. Section 1.2 describes the research objectives. Section 1.3 summarizes the thesis contributions. Section 1.4 presents the thesis organization.

1.1. Problem Statement

Although congestion pricing theory has been established since the 1920s, the pricing mechanisms researched are mainly limited to link- and cordon-based regimes. Recent technological advances have motivated further investigation into distance-based and joint regimes representing a more efficient and equitable means of congestion pricing. More importantly, congestion pricing theory to date has been largely constrained within the classical microscopic approach. To better understand the effects of pricing on traffic dynamics in a large-scale network, further research effort is needed to extend the recently established macroscopic approach. A detailed discussion of the research gaps in the literature is provided in CHAPTER 2.

1.2. Research Objectives

The main objective of this thesis is twofold:

- To propose a methodological framework integrating different computationally efficient SO methods with the NFD for solving an expensive toll level problem (TLP) in a large-scale dynamic traffic network.

- To extend the existing congestion pricing theory to better understand the effects of pricing on various network traffic phenomena such as hysteresis, gridlock, and capacity drop.

1.3. Thesis Contributions

The main contributions of this thesis are summarized as follows:

- A distance-based pricing regime is investigated to highlight its methodological limitation.
- Two joint pricing regimes, namely the joint distance and time toll (JDTT) and the joint distance and delay toll (JDDT), are proposed to extend the distance-based alternative.
- A comparison among different pricing regimes is made to characterize their respective performance.
- A unified pricing optimization framework is proposed to solve the TLP in which different SO methods can fit.
- Computationally efficient feedback control and response surface method (RSM) are examined in comparison with other SO methods including simultaneous perturbation stochastic approximation (SPSA) and DIviding RECTangles (DIRECT).
- The effects of pricing on network traffic phenomena such as hysteresis, gridlock, and capacity drop are characterized.
- Heterogeneity of congestion distribution is confirmed to be a key factor of hysteresis through computer simulations and hence considered in the optimization.

- A network partitioning approach is proposed to solve the toll area problem (TAP).

1.4. Thesis Organization

The organization of this thesis is shown in Figure 1.1.

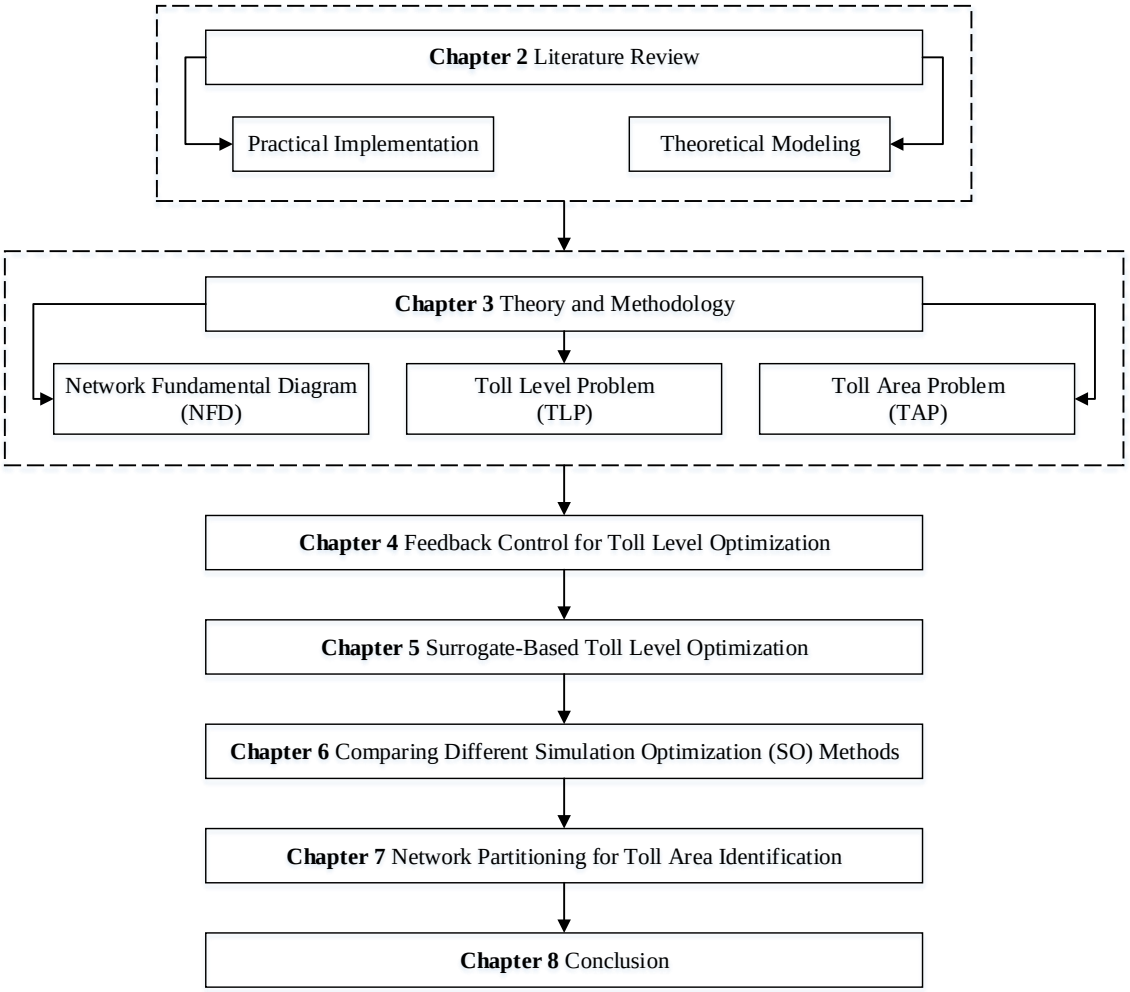


Figure 1.1 Organization of this thesis

CHAPTER 2. LITERATURE REVIEW

This chapter provides a comprehensive overview of the literature on congestion pricing practice and theory. Section 2.1 discusses the state of the practice as well as its implications. Section 2.2 reviews previous studies on congestion pricing theory with the objective of providing a solid methodological background of the existing models and solution algorithms. Section 2.3 concludes the chapter by summarizing the research gaps in the literature. The work of this chapter has been published:

- Gu, Z., Liu, Z., Cheng, Q., Saberi, M., 2018. Congestion pricing practices and public acceptance: A review of evidence. *Case Stud. Transp. Policy* 6(1), 94-101.
- Saberi, M., Gu, Z., 2018. Transport Strategy refresh background paper: Transport Pricing. City of Melbourne (CoM), Melbourne, Australia.

To facilitate the presentation, the variables used in this chapter are first summarized in Table 2.1.

Table 2.1 Variables used in CHAPTER 2

Notation	Interpretation
$\tau_h(i)$	Toll rate for the h-th tolling interval during iteration i
$\bar{K}_h(i)$	Average network density within the h-th tolling interval during iteration i
$P_{\mathrm{P}}/P_{\mathrm{I}}$	Proportional/integral gain parameter
K_{cr}	Critical network density

2.1. Congestion Pricing Practice

Road infrastructure expansion is a traditional way of alleviating traffic congestion. Due to limited space in dense urban areas as well as the well-known Braess's paradox (Sheffi, 1985), this is not a sustainable solution. Various TDM policies instead have been embraced by transportation scientists and practitioners among which congestion pricing seems to attract the most attention. Rather than a compulsory rule for travelers, e.g. traffic signal control, it is an economic lever used to influence traffic. In Sub-section 2.1.1, we discuss the definition, categorization, and implementation of congestion pricing. Implications of practice are elaborated in Sub-section 2.1.2.

2.1.1. Definition, Categorization, and Implementation

Congestion pricing is one form of road pricing with the main objective of managing demand and reducing congestion. It is therefore functionally different from other road pricing that is aimed at collecting revenue for infrastructure investment or improving environmental quality (May, 1992). With recent technological advances, e.g., see de Palma and Lindsey (2011) for a comprehensive overview, we classify congestion pricing regimes into five basic categories. See Figure 2.1 for a graphical representation.

- Link- or facility-based regime imposes a charge on specific roads or road segments. It is particularly suited for addressing isolated bottleneck or corridor congestion but not regional congestion, e.g. the usual congestion spreading across the central business district (CBD).
- Zonal regime imposes a charge on vehicles entering, exiting, or traveling entirely within a bounded area, typically referred to as the pricing zone (PZ). The charge does not distinguish between a trip that reaches the

destination immediately upon entering the bounded area and a trip that traverses the whole area.

- Cordon-based regime highly resembles the zonal regime with the only difference that vehicles traveling entirely within the bounded area are not charged. The regime can adopt a single cordon only or multiple concentric cordons with or without radial screen lines for controlling orbital movements (Sumalee, 2007).
- Distance-based regime is perhaps the state of the practice that determines the charge linearly or nonlinearly (Meng et al., 2012) based on vehicle kilometers traveled (VKT) read from the odometer or a telematics device. It is an explicit charge based on the amount of road usage as opposed to the implicit one-off zonal or cordon-based regime.
- Time- or delay-based regime is another explicit charge based on the vehicle's total travel time spent in the network. The charge by itself might result in safety and environmental concerns by encouraging vehicles to drive more aggressively and use minor roads (May and Milne, 2000).

Zonal and cordon-based regimes are recipes for regional congestion and hence can be jointly referred to as area-based pricing. During the past few decades, there has been an increasing interest from government authorities across the globe in implementing area-based pricing in the CBD where congestion tends to cause greater economic and productivity losses (Liu et al., 2013). Therefore, this is taken as the primary target in this thesis. Distance- and time-based regimes represent a more efficient and equitable means of congestion pricing that can be integrated with either link- or area-based regime. This is considered and highlighted in this thesis.

Area-based congestion pricing was first introduced in Singapore in 1975 under the name area licensing scheme (ALS). It operated until 1998 when the ERP system came as an upgrade. Following Singapore's success, several other attempts have been made in different parts of the world. Table 2.2 categorizes a few typical examples including both adopted and not adopted cases. A comprehensive discussion of each case is provided in Gu et al. (2018) and Saberi and Gu (2018).

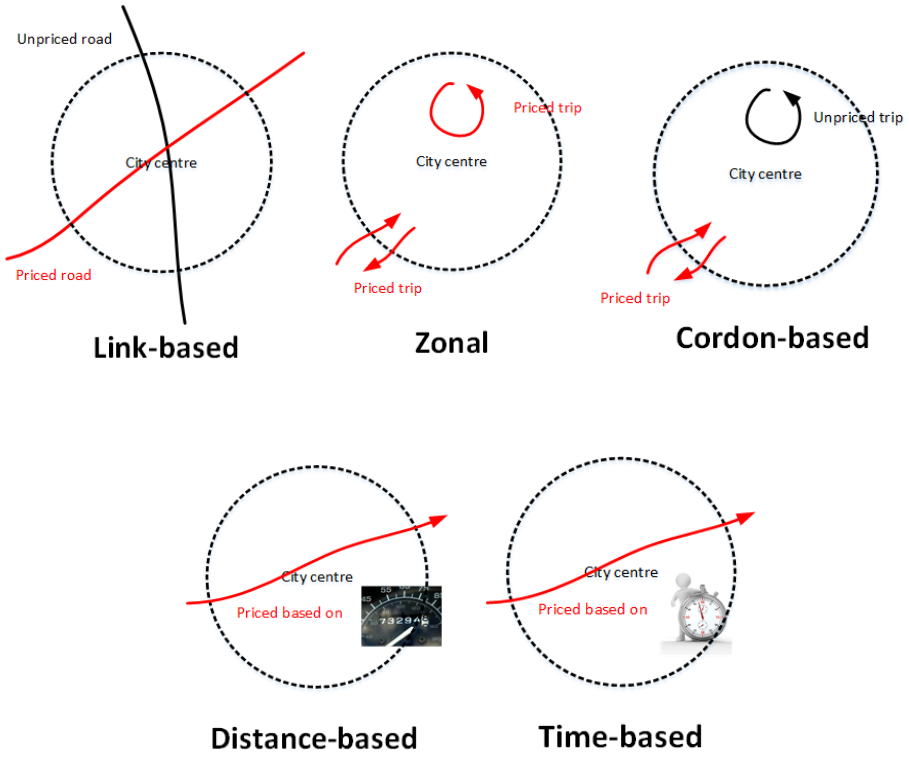


Figure 2.1 Classification of congestion pricing regimes (Saberi and Gu, 2018)

Table 2.2 Categorization of area-based congestion pricing practice around the world (Saberi and Gu, 2018)

Result	Zonal	Cordon-based	Distance-based
Adopted	Singapore (ALS)	Singapore (ERP)	Oregon, USA
	London, UK	Stockholm, Sweden	
		Milan, Italy (Ecopass and Area C)	
Not adopted	New York, USA	Hong Kong, China	
		Edinburgh, UK	
		Greater Manchester, UK	

2.1.2. Implications

Congestion pricing has been studied extensively since the seminal work by Pigou (1920) and Knight (1924). Concerns about practical implementation prevail in contrast with theory that is in favor of it. Evidence suggests that technical and financial problems no longer remain the biggest obstacles but public acceptance. A few decent discussions on this topic include Hensher and Li (2013); Noordegraaf et al. (2014); Sørensen et al. (2014); Zheng et al. (2014). A dig into the literature reveals four influencing factors on public acceptance towards congestion pricing and hence on the transition from theory to practice, namely privacy, complexity, equity, and uncertainty (Gu et al., 2018) which are further summarized in Table 2.3 for each case included in Table 2.2.

Table 2.3 Summary of the four influencing factors for each case included in Table 2.2, modified based on Gu et al. (2018)[1]

	Privacy	Equity	Complexity	Uncertainty	
				Effectiveness[2]	Revenue allocation
Singapore	√	√	√		√
London	√	√	×	√	√
New York		×	√		√
Stockholm		√	√	√	√
Milan		√	√		√
Hong Kong	×	×		×	√
Edinburgh		×	×	×	√
Greater Manchester			×		√
Oregon	√	√	√	√	√

Note: [1] √ = addressed, × = improperly handled, blank = inconclusive; [2] Here, effectiveness refers to how effective congestion pricing will be in achieving its objectives prior to permanent implementation, which can be rendered by a practical trial or theoretical modeling

The Privacy concern results from the communications technologies associated with congestion pricing that might record personal information. It is one of the major reasons why the proposed charge was not adopted on a permanent basis in Hong Kong (Hau, 1990). Singapore's ERP system, the London Congestion Charge, and the opt-in user-pays system in Oregon are all designed with considerations to address the privacy concern (Santos, 2005; Whitty, 2007). Addressing the privacy concern is relatively simple. For example, a telematics device can be configured not to transmit data when a vehicle is only a few kilometers away from its origin or destination, a practice that was previously adopted in the congestion pricing trial in Melbourne, Australia (Transurban, 2016).

The equity concern usually refers to the distributional effect of congestion pricing in that it might impose greater travel burden on low-income families and people with mobility impairments, thereby limiting their travel options. See Gu et al. (2018) for further elaboration. The emphasis here, however, is that the equity concern also arises from

a modeling perspective. Specifically, zonal and cordon-based regimes are far less equitable than distance- and time-based regimes because the charge is one-off and independent of the actual amount of road usage. The London Congestion Charge, for example, is a once-a-day charge that allows an unlimited number of passages through the PZ. According to Francke and Kaniok (2013), the distance-based regime coupled with a fixed kilometer rate was in general most preferred over the other conceivable alternatives. However, while a fixed rate is certainly easy to understand, it lacks both efficiency and equity as compared with a time-of-day rate currently adopted in Singapore and Stockholm.

Congestion pricing can be relatively simple or highly complex. Previous international experience reveals that a simple charge particularly at the initial stage is of great importance, and that the ease of understanding turns out to be critical for gaining public acceptance (Hensher and Li, 2013). In Edinburgh and Greater Manchester, part of the argument for the failure of congestion pricing was the complexity of the two pricing cordons. A lesson learned is that a gradually evolving charge is preferred over a "big bang" type of pricing reform.

The uncertainty concern surrounds the effectiveness of congestion pricing and its revenue allocation (De Borger and Proost, 2012). How the generated revenue will be allocated is perhaps relatively easy to address. Evidence suggests that the public is more supportive if revenue is meant to improve public transport (De Borger and Proost, 2012; Farrell and Saleh, 2005). The emphasis here is on the effectiveness of congestion pricing as people with inadequate information would be 2.14 times more negative than those well-informed, holding all other factors constant (Odeck and Kjerkreit, 2010). As a means of reducing risk-averse behavior (Christin et al., 2002), prior knowledge of how effective congestion pricing will be can be offered through a practical trial or theoretical modeling. However, to the best of our knowledge, the London Congestion Charge is perhaps the

only case that involves rigorous modeling while all the other cases are not well-documented and hence inconclusive in this regard. It highlights a research need from a practical point of view to develop large-scale modeling techniques for congestion pricing.

2.2. Congestion Pricing Theory

The overall toll design problem (TDP) of area-based pricing consists of the TLP and the TAP (Ekström et al., 2012). Assuming the PZ is exogenously given, i.e. without explicitly solving the TAP, a dominant research effort to date has been made to address the TLP only considering pricing regimes including but not limited to (i) zonal (Simoni et al., 2015; Ye et al., 2015), cordon-based (Liu et al., 2013; Zheng et al., 2016; Zheng et al., 2012), and distance-based (Daganzo and Lehe, 2015; Liu et al., 2017; Meng et al., 2012). A variant entry-exit based pricing regime can be found in Meng and Wang (2008); Yang et al. (2004)._ENREF_110 Yang et al. (2002) is one of the very few studies on solving the TDP, i.e. on solving the TLP and the TAP simultaneously. A deep dig into both subjects are provided in the following Sub-sections 2.2.1 and 2.2.2, respectively.

2.2.1. Toll Level Problem (TLP)

Theory on addressing the TLP originates from economic theory dating back to the 1920s. Following the acknowledged seminal work by Pigou (1920) and Knight (1924) on MCP, Walters (1961) and Li (2002) further applied MCP to the highways in the USA and the ERP system in Singapore, respectively. In essence, MCP equates to the difference between the marginal social cost and the marginal private cost so as to internalize the congestion externality. This is illustrated in Figure 2.2 as the vertical line connecting points g and h. It turns out that MCP is analytically consistent with the well-known Pigouvian tax.

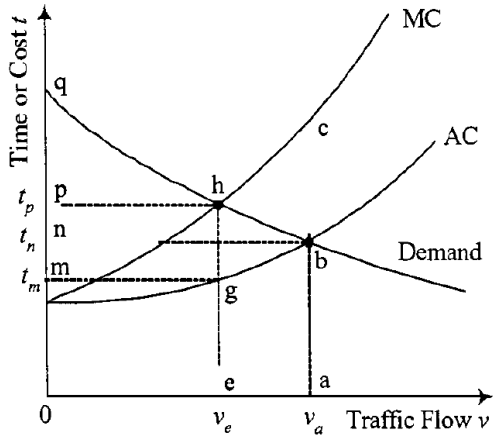


Figure 2.2 Graphical representation of MCP (Yang and Huang, 2005)

Mathematically speaking, the total travel cost associated with a flow q, $TC(q)$, is $qAC(q)$. The marginal social cost, $MC(q)$, is expressed as

$$MC(q) = \frac{\mathrm{d}TC(q)}{\mathrm{d}q} = \frac{\mathrm{d}(qAC(q))}{\mathrm{d}q} = AC(q) + q\frac{\mathrm{d}AC(q)}{\mathrm{d}q} \tag{2.1}$$

where the term $q\frac{\mathrm{d}AC(q)}{\mathrm{d}q}$ represents the amount of toll that should be imposed to make an efficient use of the facility. Yang et al. (2004) extended and applied MCP to a general traffic network and found that if every link in the network is tolled based on MCP, the network is driven from user equilibrium (UE) to system optimum. Here system optimum refers to the minimization of the total travel cost in the case of fixed demand or the maximization of the social cost in the case of elastic demand. This is the theory of first-best pricing. Analytically speaking, first-best pricing problems can be formulated as nonlinear optimization problems and solved using one of the available methods, e.g. the Frank-Wolfe method.

Despite apparently perfect theoretical basis, practical implementation of first-best pricing has hardly achieved any progress simply because tolling every link in the network

results in a high operating cost and poor public acceptance. Therefore, various second-best pricing problems have been proposed focusing on part of the network only. Mathematically speaking, all the second-best pricing problems can be treated as mathematical programming with equilibrium constraints (MPEC), a particular case of bilevel optimization. The upper level forms the objective function to be minimized or maximized based on the problem at hand while the lower level models travelers' collective route choice behavior in the network.

In summary, all the second-best pricing problems differ in three aspects:

- What is the objective?
- What is the pricing regime?
- What is the assumption on the route choice behavior?

The objective of second-best pricing problems can be multiple including but not limited to (i) total travel time minimization (Chen et al., 2014), (ii) network speed regularization (Liu et al., 2013), (iii) total revenue maximization (Chen et al., 2016), (iv) network travel time reliability maximization (Chen et al., 2018), and (v) network flow maximization (Zheng et al., 2016). Each of these objectives corresponds to a unique way by which the network is evaluated and hence, different researchers and practitioners may have different preferences. The assumption on the route choice behavior is typically UE including its stochastic and dynamic counterparts or non-equilibrium stochastic flow. Alternatively, we can categorize the assumption into static traffic assignment (STA) and dynamic traffic assignment (DTA). While a few studies adopted STA (Liu and McDonald, 1999; Liu et al., 2013; Liu et al., 2014; Meng et al., 2012; Verhoef, 2002; Yang and Zhang, 2003; Zhang and Yang, 2004), there is a growing interest in implementing DTA whereby traffic conditions, especially congestion propagation, are allowed to vary over different time intervals (Chen et al., 2016; Chung et al., 2012; de Palma et al., 2005; Lawphongpanich and

Yin, 2012; Liu et al., 2017; Tan et al., 2015). This certainly facilitates investigation into the more appealing dynamic congestion pricing.

All the aforementioned studies belong to the "traditional" second-best pricing paradigm. With recent advances on the theory and applications of the NFD, promising research efforts have been made to extend and modernize this tradition by integrating the NFD with MPEC, thereby providing valuable insights into the effects of pricing on network traffic dynamics and phenomena. While Geroliminis and Levinson (2009) and Daganzo and Lehe (2015) incorporated the NFD into a bottleneck model for modeling the capacity drop phenomenon, respectively, the first fundamental study on combining the NFD with second-best pricing in a large-scale dynamic traffic network was only recently conducted by Zheng et al. (2012) where a discrete integral (I-type) feedback controller was applied within an agent-based simulation environment:

$$\tau_h(i) = \begin{cases} \tau_h(i-1) + P_{\mathrm{I}}(\overline{K}_h(i) - K_{\mathrm{cr}}), & i > 1 \\ P_{\mathrm{I}}(\overline{K}_h(i) - K_{\mathrm{cr}}), & i = 1 \end{cases} \tag{2.2}$$

where $\tau_h(i)$ is the adjusted toll rate for the h-th tolling interval during iteration i, $\overline{K}_h(i)$ is the average network density within the h-th tolling interval during iteration i, $P_{\mathrm{I}} > 0$ is an integral gain parameter to be estimated, and K_{cr} is the critical network density identified from the NFD. When $i = 1$, the simulation is run without pricing as the base scenario. In short, the NFD was used to describe congestion at the network level based on which the toll rate was iteratively adjusted such that the NFD of the PZ did not enter the congested regime. See Figure 2.3 for a graphical interpretation.

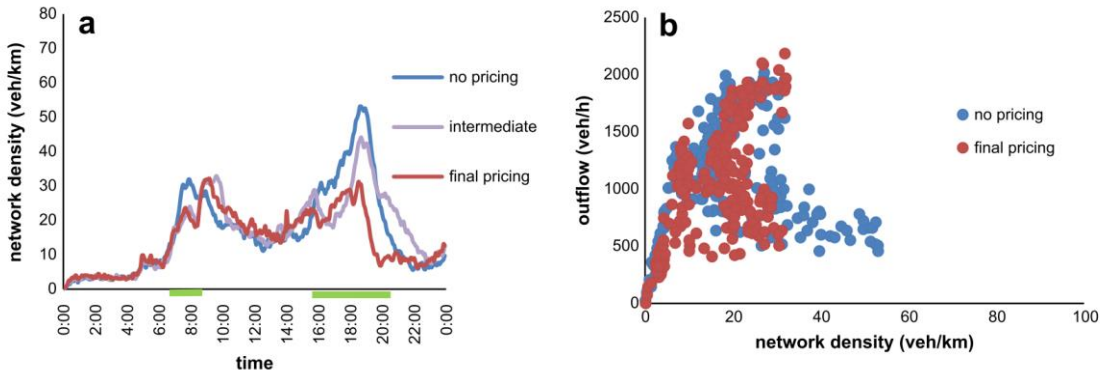


Figure 2.3 NFD-based pricing control logic (Zheng et al., 2012)

This NFD-based pricing control logic turns out to be axiomatic forming the methodological basis for a few subsequent studies (Gu et al., 2018; Zheng et al., 2016). The I-type controller was later extended to a proportional-integral (PI) counterpart in Zheng et al. (2016) whereby travelers' adaptation to pricing was considered and modeled:

$$\tau_h(i) = \begin{cases} \tau_h(i-1) + P_{\mathrm{P}}\big(\overline{K}_h(i) - \overline{K}_h(i-1)\big) + P_{\mathrm{I}}(\overline{K}_h(i) - K_{\mathrm{cr}}), & i > 1 \\ P_{\mathrm{I}}(\overline{K}_h(i) - K_{\mathrm{cr}}), & i = 1 \end{cases} \tag{2.3}$$

where $P_{\mathrm{P}} > 0$ is an additional proportional gain parameter to be estimated. A comparison between the two feedback controllers was made and the simulation results confirmed that the latter outperformed the former. A similar feedback structure for NFD-based pricing was also proposed in Simoni et al. (2015), although without using a typical feedback controller. Instead, the NFD as well as the generalized 3D-NFD was integrated with MCP for deriving the toll adjustment rule that leads to the optimum. Note that, in the presence of a black-box simulation without explicit mathematical modeling, integrating the NFD with a feedback controller requires trial-and-error to estimate the controller gain parameters. In contrast, when the system can be described fully mathematically as a set of

equations (e.g., a multi-reservoir NFD-based system), a control engineering method can be employed to obtain presumably better controller gain parameters (Keyvan-Ekbatani et al., 2016).

To solve a second-best pricing problem, one can indeed devise an exact solution algorithm (Liu et al., 2014) or even use the somewhat "stupid" brute force method. However, given the usual non-convexity, non-linearity, and non-closed form characteristics of the objective function rendered by DTA, a heuristic algorithm is more employed (Verhoef, 2002). The biggest obstacle, unfortunately, is its computational complexity particularly in the presence of a large-scale dynamic traffic network, i.e. its scalability. Therefore, to solve such a problem featuring a computationally expensive objective function, a high-dimensional decision vector comprising multiple decision variables, and simulation (if used) noise, a "smart" enough method is needed to guide the search for the optimum in a computationally efficient manner.

A stochastic traffic simulator, if used, introduces a source of numerical noise through different random seed numbers. Meanwhile, many deterministic computer experiments involve another type of numerical noise that refers to the random deviations from the expected smooth response (Forrester et al., 2006). The key message is that stochastic traffic simulation tends to further increase the computation complexity of the problem at hand and limit our options for candidate solutions. Indeed, one could hardly devise an analytical method in the presence of a black-box simulation due to the lack of an explicit mathematical model of the system under consideration. For the same reason, exact gradient methods are no longer applicable but stochastic approximation methods might be considered as an alternative, e.g. SPSA (Spall, 1992).

Given the above demanding requirements for solving an expensive second-best pricing problem, a family of algorithms termed simulation-based optimization (SO or

SBO) have recently been investigated and advocated as a computationally efficient method (Amaran et al., 2016; Osorio and Bierlaire, 2013; Osorio and Chong, 2015). While SO has already been applied to a variety of fields, its application to toll level optimization is quite recent. Specifically, existing SO methods for solving the TLP can be classified into two broad categories:

- Feedback control (Simoni et al., 2015; Zheng et al., 2016; Zheng et al., 2012)
- Surrogate-based optimization (Chen et al., 2016; Chen et al., 2014; Chen et al., 2018; Chow and Regan, 2014; Ekström et al., 2016; He et al., 2017)

We have already discussed a few studies on integrating the NFD with feedback control which, to some extent, resembles the trial-and-error method (Yang et al., 2004). While enjoying desirable properties such as fast and global convergence and robustness (Zheng et al., 2012), feedback control has its own methodological constraint. This is elaborated in CHAPTER 4 and CHAPTER 6, respectively. Surrogate-based optimization, also known as RSM or metamodeling, is a totally different alternative method aiming to construct a mathematical model of a simulation model. It focuses on learning and approximating the simulation input-output mapping using a limited number of function evaluations (Amaran et al., 2016). See Figure 2.4 for a graphical interpretation.

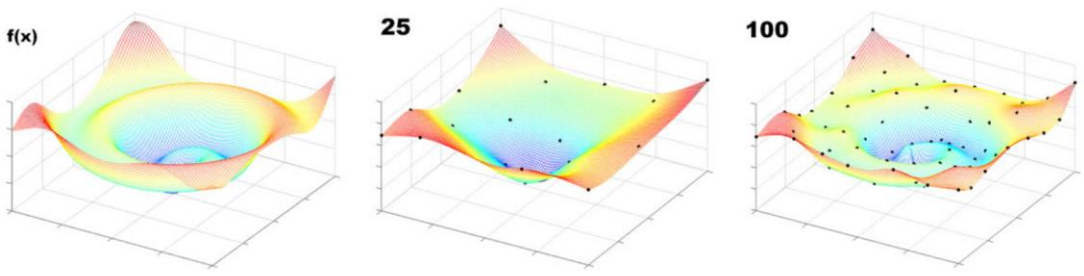


Figure 2.4 A costly function and the associated surrogate models using 25 and 100 sample points, respectively (Ekström et al., 2016)

While surrogate models can be built in local regions to sequentially guide the search for the optimum, global surrogate models from space-filling design of experiments (DOE) are more appealing given their capability to find the global optimum (Forrester et al., 2008; Jones et al., 1998). Several successful attempts have been made to date to apply surrogate-based optimization for solving expensive second-best pricing problems considering different objectives and functional forms of the response surface. For example, Chow and Regan (2014) chose the radial basis function to construct the surrogate model and solved a constrained multi-objective toll optimization problem. However, a comprehensive comparison between different surrogate models revealed that (regressing) kriging with expected improvement (EI) sampling is the best performing surrogate model (Chen et al., 2014; Ekström et al., 2016), which has therefore been further investigated in Chen et al. (2016); Chen et al. (2018); He et al. (2017). Kriging, also known as Gaussian process regression or Bayesian optimization, originates from geostatistics and has become popular in designing and analyzing computer experiments (Sacks et al., 1989). Coupled with EI sampling, it first constructs using an initial set of sample points a somewhat coarse response surface assuming a Gaussian process, and then refine the response surface in an iterative manner by adding additional infill sample points. The method is therefore capable of balancing between global exploration and local exploitation (Forrester et al., 2008).

2.2.2. Toll Area Problem (TAP)

Unlike the TLP which has been researched quite extensively, studies on the TAP are relatively limited. As we have previously touched upon, most studies on the TDP assumed an exogenously given PZ, thereby reducing the original problem to the TLP only. As part of the overall TDP, further investigation into the TAP is desirable.

To explicitly solve the TAP, a few studies employed a somewhat engineering-oriented judgmental approach. A review of various judgmental criteria can be found in

May et al. (2002). Since the judgmental approach heavily relies on the topology of the city under consideration, the resultant PZ is largely experience-based rather than being a product out of explicit mathematical modeling. In this context, it might even be true that the solution to the TLP is sub-optimal (May et al., 2002; Sumalee, 2004). To address this concern, a location index based method was proposed for simultaneously determining the optimal toll locations and levels (May et al., 2002). The judgmental approach was considered only as a supplementary tool to obtain the candidate set of toll locations. Despite having some methodological limitations, e.g. being unable to deselect links, the "LOCATE" method was shown to outperform the judgmental approach in terms of the social welfare achieved, and was later integrated with the genetic algorithm (GA) to create an improved "GALOCATE" heuristic method (May et al., 2002; Shepherd and Sumalee, 2004)_ENREF_36. An exact solution algorithm for simultaneously determining the optimal toll locations and levels was only recently proposed by Ekström et al. (2014); Ekström et al. (2012)_ENREF_2. The biggest problem, however, is that both the heuristic and exact methods failed to explicitly consider the closed format of a PZ. That is, the connectivity and compactness of the optimal toll locations were not necessarily guaranteed.

The very first study that explicitly took into account the closed format of a PZ belongs to Yang et al. (2002). While the traffic network was viewed as a directed graph, the concept of cutset in graph theory was applied to mathematically represent a closed PZ. This mathematical formulation was later considered as an additional constraint in the TDP solved by the GA. In a similar fashion, Sumalee (2004) proposed a graph theory based branch-tree framework to mathematically define a closed PZ. The framework was integrated with the GA resulting in a GA-AS method for solving the TAP. Subsequent studies on comparing the GA-AS method with the judgmental approach revealed that a

mathematically derived PZ produced much greater welfare benefits than the judgmental counterpart (Shepherd et al., 2007; Sumalee, 2007; Sumalee et al., 2005).

The above GA-based methods require a predefined initial PZ to generate a set of extended or contracted candidate PZs among which the optimum is to be identified. Otherwise there can be an enormous number of candidate PZs particularly in a large-scale network which renders the problem intractable. Inspired by recent studies on segmenting a heterogeneous network into multiple spatially connected and homogeneous reservoirs using link density, speed, or travel time data (Ji and Geroliminis, 2012; Lopez et al., 2017; Saeedmanesh and Geroliminis, 2016, 2017), a possible solution to the TAP is to apply network partitioning, also known as contiguity-constrained clustering. Network partitioning has the capability of capturing and categorizing the spatial distribution of congestion in the network which is therefore promising for solving the TAP, because a sensible PZ should encapsulate as many as possible the main pockets of congestion. However, the applicability of the existing network partitioning methods is quite limited in that the resultant partitioned network is typically a multi-reservoir system only suited for defining multiple disjoint PZs. In a big city where several congested sub-networks co-exist, a coordinated multi-area pricing system is conceivable but still needs further investigation. For many of the existing real-world pricing implementations (see Section 2.1), the network is modeled as a single-cordon two-region system. In such a two-region network, network partitioning has unfortunately not been investigated in depth to realize its full potential. Unlike the existing GA-based methods which are completely optimization-driven, network partitioning incorporates optimization into a data-driven perspective. The method also provides an interface with the existing GA-based methods into which the identified PZ can be fed as the initial PZ for further refinement. It is therefore a promising

method for solving the TAP particularly in a large-scale network, being both theoretically contributing and of practical significance.

2.3. Chapter Remarks

This chapter provides a comprehensive review of the literature on congestion pricing practice and theory. The overview on practice reveals a theoretical imperative to investigate more efficient and equitable pricing regimes and to develop a set of methods for large-scale pricing modeling and optimization. Among all the discussed practical implementations in Section 2.1, the London Congestion Charge seems to be the only case involving rigorous modeling to determine the initial toll rate. The ERP system in Singapore is perhaps the only case that currently has a clear logic for toll rate adjustment.

The overview on theory covers the state of the art on both the TLP and the TAP. While research on the TLP is well established, the question of large-scale pricing coupling advanced pricing regimes (see Section 3.2) with a macroscopic perspective (see Section 3.1) remains wide open. This thesis tries to answer this question through different computationally efficient SO methods (see Sections 3.3 and 3.4). Studies on the TAP, however, are relatively limited. The existing GA-based methods provide a solution but is dependent on the initialization of the PZ. This is addressed in this thesis through network partitioning (see Section 3.5).

CHAPTER 3. THEORY AND METHODOLOGY

This chapter elaborates on the theory and methodology behind the proposed work. Section 3.1 describes the theory of the NFD as well as how it can be used for toll optimization. Section 3.2 formulates what we call joint tolls as a more efficient and equitable pricing regime. Sections 3.3 and 3.4 present two computationally efficient SO methods, respectively, for solving the TLP. Section 3.5 proposes a network partitioning method for solving the TAP. Section 3.6 concludes the chapter.

3.1. Network Fundamental Diagram (NFD)

To facilitate the presentation, the variables used in this section are first summarized in Table 3.1.

Table 3.1 Variables used in Section 3.1

Notation	Interpretation
K	Average network density
Q	Average network flow
k_i	Average density of link i
q_i	Average flow of link i
l_i/l_a	Length of link i/ a
n_i	Number of lanes of link i
γ	Spatial spread of density
δ	Deviation from spread

The NFD is a macroscopic traffic flow relation for an urban area linking space-mean flow, density, and speed. Specifically, when the somewhat scattered speed-density relationships of individual links in the network are aggregated, the scatter nearly disappear with points lying neatly along a smooth inverse U-shaped curve (Geroliminis and

Daganzo, 2008). While the NFD can be accurately estimated based on Edie's definitions of traffic flow variables (Edie, 1963), e.g. using vehicle trajectory data (Saberi et al., 2014), it can also be approximated perhaps a bit more easily as distance-weighted averages using fixed detector data (Mahmassani et al., 1984):

$$K = \frac{\sum_i k_i l_i n_i}{\sum_i l_i n_i} \tag{3.1}$$

$$Q = \frac{\sum_i q_i l_i n_i}{\sum_i l_i n_i} \tag{3.2}$$

where K and Q are the average network density and flow, respectively, k_i and q_i are the average density and flow of link i, respectively, and l_i and n_i are the length and the number of lanes of link i, respectively. See Figure 3.1 for a graphical representation. Note that in a simulation environment, we have exact knowledge of each wanted traffic flow variable for each link in the network. Hence the resultant NFD is ideal as opposed to an operational NFD that is estimated based on imperfect measurements (Keyvan-Ekbatani et al., 2012).

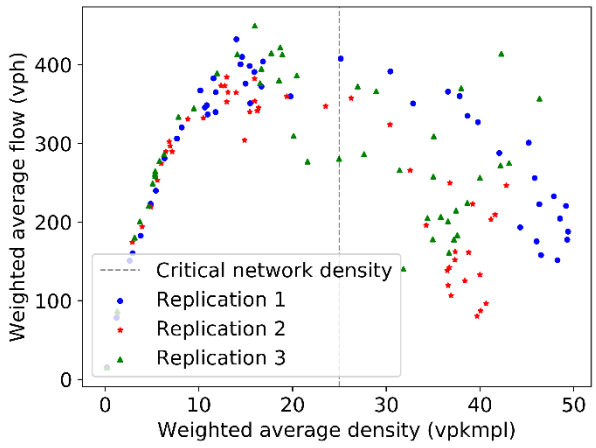


Figure 3.1 Estimated NFDs of the Melbourne CBD from simulation data

The NFD has two desirable properties (Geroliminis and Daganzo, 2008):

- There is a robust linear relation between the network flow and the trip completion rate.
- The shape of the NFD is a property of the network itself including infrastructure and control, and is not very sensitive to different demand patterns.

The first property implies that a detailed knowledge of how the OD demand varies is not necessarily required for NFD-based modeling and optimization. The second property implies that, since the trip completion rate can hardly be measured in reality, the network flow, which is more observable through different types of sensors, can be used instead to measure accessibility, a very important network characteristic.

Since the heterogeneity of congestion distribution is a key determinant of the shape and scatter of the NFD (Buisson and Ladier, 2009; Geroliminis and Sun, 2011; Mahmassani et al., 2013; Mazloumian et al., 2010; Saberi and Mahmassani, 2012, 2013), we introduce the spatial spread of density, γ, representing how congestion is distributed within an area. By definition (Knoop and Hoogendoorn, 2013), it is estimated as the square root of the distance-weighted variance of all link densities:

$$\gamma = \sqrt{\frac{\sum_i l_i n_i (k_i - K)^2}{\sum_i l_i n_i}} \tag{3.3}$$

The spatial spread of density naturally increases with a growing accumulation. That is, an increase in vehicles entering the area inevitably generates a higher spatial spread of density later in time as these vehicles continue their trips within the area. Given the correlation between the spatial spread of density and the accumulation, the level of heterogeneity of congestion distribution is better interpreted as positive deviations from the natural

increment represented by the lower envelope in the spread-accumulation relationship (Simoni et al., 2015). By fitting a polynomial function, $\gamma(K)$, to the lower envelope, the deviation from spread, δ, is obtained:

$$\delta = \gamma - \gamma(K) \tag{3.4}$$

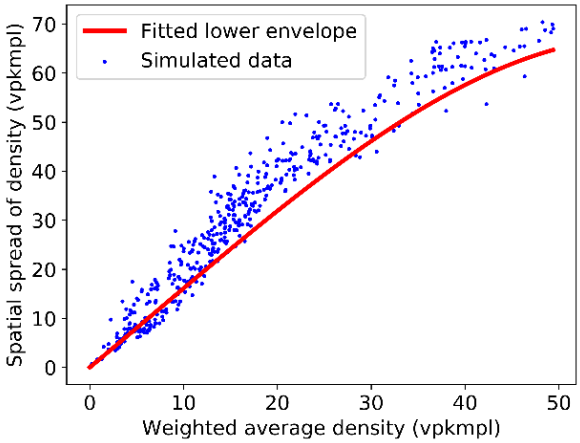


Figure 3.2 Spread-accumulation relationship of the Melbourne CBD from simulation data as well as the fitted lower envelope to represent the deviation from spread

When using the NFD for network control and management, the objective is typically to keep the network operating around the critical network density so as to maximize the rate at which trips are served (Daganzo, 2007). When the network becomes congested or gridlocked, i.e., the network density increases beyond the critical threshold, the network flow or the trip completion rate significantly drops causing undesirable network unproductivity. Meanwhile, given that a more heterogeneous distribution of congestion equates to an increased hysteresis loop in the NFD causing network unproductivity as well, another objective worthy of consideration is to reduce the heterogeneity of congestion distribution (Ramezani et al., 2015). When implementing different network control and management strategies, some studies assumed that the NFD does not change significantly. However, it should be noted that this assumption does not necessarily hold,

especially in the presence of adaptive traffic signals that are found to increase the maximum network flow as well as the critical network density (Keyvan-Ekbatani et al., 2016; Zhang et al., 2013). However, since this thesis deals with congestion pricing problems without considering adaptive traffic signals, we still assume an unchanged or slightly changed NFD after pricing. Further post-check will be performed to ensure the validity of this assumption.

3.2. Joint Tolls

To facilitate the presentation, the variables used in this section are first summarized in Table 3.2.

Table 3.2 Variables used in Section 3.2

Notation	Interpretation
N /N_{p}/N_{np}	Set of nodes in the entire sub-network/PZ/peripheral sub-network
A/A_{p}/A_{np}	Set of directed links in the entire sub-network/PZ/peripheral sub-network
W	Set of OD pairs where $O \subset N$ is the set of origins and $D \subset N$ is the set of destinations, i.e. $W = \{(o,d) \mid o \in O, d \in D\}$
R^{od}	Set of paths between an OD pair $(o,d) \in W$
m	Total number of tolling intervals
$t_a(h)$	Average travel time on link $a \in A$ during the h-th tolling interval
$\delta_{a,r}^{od}$	$\delta_{a,r}^{od} = 1$ if path $r \in R^{od}$ includes link a, otherwise $\delta_{a,r}^{od} = 0$
υ_h/η_h/ξ_h	Distance/time/delay toll rate
t_a^{f}	Free-flow travel time on link a

Zonal and cordon-based pricing regimes are inefficient and inequitable. Unlike a usage-based charge, this type of pay-per-entry fee does not consider the amount of road usage in the PZ and hence is not linked to one's actual contribution to congestion. It is unreasonable to apply the same amount of toll to a trip that reaches the destination

immediately upon entering the PZ and to a trip that traverses the whole area. In this context, distance-based pricing is a much better option and indeed the state of the practice. Apart from the latest opt-in distance-based pricing system, OReGo, in Oregon, USA[2], Singapore's ERP system is planned to upgrade from 2020 onward to be distance-based[3].

A limitation of the distance only toll, however, is that it naturally drives travelers into the shortest paths within the PZ (Liu et al., 2014) resulting in a heterogeneous distribution of congestion and hence a large hysteresis loop in the NFD. This is investigated and highlighted in CHAPTER 4. But, we can do better with the help of various emerging pricing technologies (de Palma and Lindsey, 2011). Note that a few advanced pricing concepts and schemes have been proposed recently such as New York City's Move NY Plan which aims to charge taxis based on both distance and time[4], and the joint distance- and cordon-based pricing trial in Melbourne, Australia (Transurban, 2016).

To overcome the limitation of the distance only toll, we propose and study two joint tolls, namely the JDTT and the JDDT. The JDTT is assumed linearly proportional to both the distance traveled and the time spent within the PZ. As such, travelers would no longer accumulate themselves into the shortest paths within the PZ provided that the travel times on these paths increase substantially. The JDDT works in a similar fashion but slightly improves the equity of the JDTT. Specifically, since the time toll component as part of the JDTT tends to overcharge travelers on a longer link that typically requires more travel time despite being uncongested, the JDDT considers instead a delay toll component that charges travelers in proportion to their experienced travel delays.

[2] See http://www.myorego.org/.
[3] See https://www.lta.gov.sg/apps/news/default.aspx?scr=yes&keyword=ERP2.
[4] See https://nyc.streetsblog.org/2015/02/17/the-complete-guide-to-the-final-move-ny-plan/.

Let $G = (N, A)$ denote a network where N is the set of nodes and A is the set of directed links. With a predefined pricing cordon, network G is partitioned into a PZ denoted by $G_{\mathrm{p}} = (N_{\mathrm{p}}, A_{\mathrm{p}})$ and a peripheral sub-network denoted by $G_{\mathrm{np}} = (N_{\mathrm{np}}, A_{\mathrm{np}})$. Let $l_r^{od}(h)$ and $t_r^{od}(h)$ denote respectively the distance traveled and the time spent within the PZ for path $r \in R^{od}$ during the h-th tolling interval:

$$l_r^{od}(h) = \sum_{a \in A_{\mathrm{p}}} \delta_{a,r}^{od} \, l_a \tag{3.5}$$

$$t_r^{od}(h) = \sum_{a \in A_{\mathrm{p}}} \delta_{a,r}^{od} \, t_a(h) \tag{3.6}$$

where $r \in R^{od}, (o, d) \in W, h \in (1,2, \dots, m)$. Given the linearity assumption, the distance, time, and delay toll components for path $r \in R^{od}$ during the h-th tolling interval are calculated as follows:

$$DI_r^{od}(h) = \upsilon_h \sum_{a \in A_{\mathrm{p}}} \delta_{a,r}^{od} \, l_a \tag{3.7}$$

$$TI_r^{od}(h) = \eta_h \sum_{a \in A_{\mathrm{p}}} \delta_{a,r}^{od} t_a(h) \tag{3.8}$$

$$DE_r^{od}(h) = \xi_h \sum_{a \in A_{\mathrm{p}}} \delta_{a,r}^{od} \left(t_a(h) - t_a^{\mathrm{f}} \right) \tag{3.9}$$

where $\upsilon_h, \eta_h, \xi_h \geq 0$ are the distance, time, and delay toll rates, respectively, and t_a^{f} is the free-flow travel time on link $a \in A_{\mathrm{p}}$. The generalized travel cost function for path $r \in R^{od}$ during the h-th tolling interval is therefore expressed as

$$C_r^{od}(h) = \sum_{a \in A} \delta_{a,r}^{od} t_a(h) + \frac{DI_r^{od}(h) + TI_r^{od}(h) + DE_r^{od}(h)}{VTT} \tag{3.10}$$

where VTT is travelers' average value of travel time. Clearly different choices of $\upsilon_h, \eta_h, \xi_h$ lead to different pricing regimes. For example, when $\upsilon_h > 0, \eta_h = \xi_h = 0$, we have the distance toll only; when $\upsilon_h, \eta_h > 0, \xi_h = 0$, we have the JDTT; when $\upsilon_h, \xi_h > 0, \eta_h = 0$, we have the JDDT. In our simulation model (see APPENDIX A) used in the subsequent chapters, Equation (3.10) is integrated with the C-logit stochastic route choice model (Cascetta et al., 1996) for path assignment.

3.3. Feedback Control

To facilitate the presentation, the variables used in this section are first summarized in Table 3.3.

Table 3.3 Variables used in Section 3.3

Notation	Interpretation
$P_{\mathrm{P}}^{\upsilon}/P_{\mathrm{p}}^{\eta}/P_{\mathrm{P}}^{\xi}$	Proportional gain parameter for $\upsilon_h/\eta_h/\xi_h$
$P_{\mathrm{I}}^{\upsilon}/P_{\mathrm{I}}^{\eta}/P_{\mathrm{I}}^{\xi}$	Integral gain parameters for $\upsilon_h/\eta_h/\xi_h$
$K_h^{\max}(i)$	Maximum network density within the h-th tolling interval during iteration i
$\rho_{\upsilon}/\rho_{\eta}$	Scaling parameter for υ_h/η_h
$P_{\mathrm{P}}, P_{\mathrm{I}}$	Nominal proportional and integral gain parameters
i^*	Iteration in which the steady-state error comes close to zero
ω_1/ω_2	Weight coefficient
$v_a(h)$	Average speed on link a

The PI controller as a classical feedback control strategy has been widely used for traffic control and management purposes. A well-known example is the ALINEA ramp

metering (Papageorgiou et al., 1991). When applied to congestion pricing, the PI controller can work both in real time (Yin and Lou, 2009) and in a day-to-day fashion (Zheng et al., 2016). The difference between the two is twofold:

- Using a small time step, the real-time PI controller produces a frequently changing toll rate over time whereas the day-to-day counterpart produces different static toll rates for different tolling intervals.
- The real-time PI controller does not require an iterative solution framework since the input to the controller for the current time interval always comes from the previous interval. Differently, the day-to-day PI controller needs iterations. For a time interval of interest during the current iteration, the input to the day-to-day PI controller always comes from the same interval but in the previous iteration.

Since we use the NFD to describe congestion at the network level, the PI controller is a handy entry point to solving the TLP because the critical network density identified from the NFD naturally becomes the set point in the PI controller. Given its fast and global convergence and robustness properties (Zheng et al., 2012), the PI controller can effectively achieve our control objective. We emphasize that the convergence property is only valid for the day-to-day PI controller. The output of the real-time PI controller always varies because of the changing real-time input measurements. Note that when the objective changes to, for example, minimizing the total travel time, the PI controller can hardly be applied due to the lack of a set point. Another limitation is its inability to consider complex constraints. In Sub-sections 3.3.1 and 3.3.2, we propose a simultaneous and a sequential feedback control approach, respectively.

3.3.1. Simultaneous Feedback Control Approach

Under the distance only toll scenario, the distance toll rate, υ_h, is the only control input to the network. As such, we can readily apply Equation (2.3) to iteratively adjust υ_h until the control objective is met. Differently, under the JDTT scenario, there is an additional control input to the network other than υ_h, namely the time toll rate, η_h. Therefore, we propose the following simultaneous approach based on the PI controller for iteratively adjusting the JDTT. When $i > 1$, the discrete PI controller for the JDTT is expressed in the following matrix form:

$$\begin{bmatrix}\upsilon_h(i)\\ \eta_h(i)\end{bmatrix} = \begin{bmatrix}\upsilon_h(i-1)\\ \eta_h(i-1)\end{bmatrix} + \begin{bmatrix}P_{\mathrm{P}}^{\upsilon} & P_{\mathrm{I}}^{\upsilon}\\ P_{\mathrm{p}}^{\eta} & P_{\mathrm{I}}^{\eta}\end{bmatrix}\begin{bmatrix}K_h^{\max}(i) - K_h^{\max}(i-1)\\ K_h^{\max}(i) - K_{\mathrm{cr}}\end{bmatrix} \tag{3.11}$$

where $P_{\mathrm{P}}^{\upsilon}, P_{\mathrm{I}}^{\upsilon} > 0$ ($P_{\mathrm{p}}^{\eta}, P_{\mathrm{I}}^{\eta} > 0$) are the proportional and integral gain parameters for υ_h (η_h), respectively. Here we use $K_h^{\max}(i)$, the maximum network density within the h-th tolling interval during iteration i, in place of $\overline{K}_h(i)$ so that the PI controller is more aggressive. Since there are four parameters to be estimated, directly applying trial-and-error can lead to different combinations of parameters and hence different optimal steady-state toll rates. These toll rates are all considered optimal given that the control objective is achieved. To ensure a unique optimum, we need to balance between υ_h and η_h.

Knowing that the PI controller is robust to moderate parameter changes, we assume that the rank of $\begin{bmatrix}P_{\mathrm{P}}^{\upsilon} & P_{\mathrm{I}}^{\upsilon}\\ P_{\mathrm{p}}^{\eta} & P_{\mathrm{I}}^{\eta}\end{bmatrix}$ is one and hence $\begin{bmatrix}P_{\mathrm{P}}^{\upsilon} & P_{\mathrm{I}}^{\upsilon}\\ P_{\mathrm{p}}^{\eta} & P_{\mathrm{I}}^{\eta}\end{bmatrix} = \begin{bmatrix}\rho_{\upsilon} & 0\\ 0 & \rho_{\eta}\end{bmatrix}\begin{bmatrix}P_{\mathrm{P}} & P_{\mathrm{I}}\\ P_{\mathrm{P}} & P_{\mathrm{I}}\end{bmatrix}$ where $\rho_{\upsilon}, \rho_{\eta} > 0$ are the scaling parameters to be determined, respectively, and $P_{\mathrm{P}}, P_{\mathrm{I}} > 0$ are the nominal proportional and integral gain parameters, respectively. By denoting $\begin{bmatrix}\upsilon_h(i)\\ \eta_h(i)\end{bmatrix}$

as $\boldsymbol{\tau}_h(i)$, $\begin{bmatrix}\rho_v & 0\\ 0 & \rho_\eta\end{bmatrix}$ as $\boldsymbol{\rho}$, $\begin{bmatrix}P_{\mathrm{P}} & P_{\mathrm{I}}\\ P_{\mathrm{P}} & P_{\mathrm{I}}\end{bmatrix}$ as $\boldsymbol{P}$, and $\begin{bmatrix}K_h^{\max}(i)-K_h^{\max}(i-1)\\ K_h^{\max}(i)-K_{\mathrm{cr}}\end{bmatrix}$ as $\begin{bmatrix}E_h^{\mathrm{P}}(i)\\ E_h^{\mathrm{I}}(i)\end{bmatrix}$ or simply $\boldsymbol{E}_h(i)$, Equation (3.11) can be rewritten in a compact form:

$$\boldsymbol{\tau}_h(i)=\boldsymbol{\tau}_h(i-1)+\boldsymbol{\rho}\boldsymbol{P}\boldsymbol{E}_h(i) \tag{3.12}$$

When $i=1$, $\boldsymbol{\tau}_h(1)=\boldsymbol{\rho}\boldsymbol{P}\boldsymbol{E}_h(1)=\begin{bmatrix}\rho_v & 0\\ 0 & \rho_\eta\end{bmatrix}\begin{bmatrix}P_{\mathrm{P}} & P_{\mathrm{I}}\\ P_{\mathrm{P}} & P_{\mathrm{I}}\end{bmatrix}\begin{bmatrix}0\\ E_h^{\mathrm{I}}(1)\end{bmatrix}=\begin{bmatrix}\rho_v P_{\mathrm{I}} E_h^{\mathrm{I}}(1)\\ \rho_\eta P_{\mathrm{I}} E_h^{\mathrm{I}}(1)\end{bmatrix}$.

Proposition 3.1 The ratio between the optimal steady-state toll rates is equal to the ratio between the scaling parameters.

Proof. Let i^* denote the iteration in which the steady-state error comes close to zero. We expand Equation (3.12) recursively as follows:

$$\begin{aligned}\boldsymbol{\tau}_h(i^*)&=\boldsymbol{\tau}_h(i^*-1)+\boldsymbol{\rho}\boldsymbol{P}\boldsymbol{E}_h(i^*)=\boldsymbol{\tau}_h(i^*-2)+\sum_{i=i^*-1}^{i^*}\boldsymbol{\rho}\boldsymbol{P}\boldsymbol{E}_h(i)\\ &=\cdots=\boldsymbol{\tau}_h(1)+\sum_{i=2}^{i^*}\boldsymbol{\rho}\boldsymbol{P}\boldsymbol{E}_h(i)\end{aligned} \tag{3.13}$$

Given that

$$\sum_{i=2}^{i^*} \boldsymbol{\rho PE}_h(i) = \sum_{i=2}^{i^*} \begin{bmatrix} \rho_v & 0 \\ 0 & \rho_\eta \end{bmatrix} \begin{bmatrix} P_\mathrm{P} & P_\mathrm{I} \\ P_\mathrm{P} & P_\mathrm{I} \end{bmatrix} \begin{bmatrix} E_h^\mathrm{P}(i) \\ E_h^\mathrm{I}(i) \end{bmatrix}$$

$$= \begin{bmatrix} \rho_v \left(\sum_{i=2}^{i^*} P_\mathrm{P} E_h^\mathrm{P}(i) + P_\mathrm{I} E_h^\mathrm{I}(i) \right) \\ \rho_\eta \left(\sum_{i=2}^{i^*} P_\mathrm{P} E_h^\mathrm{P}(i) + P_\mathrm{I} E_h^\mathrm{I}(i) \right) \end{bmatrix} \tag{3.14}$$

Equation (3.13) can be rewritten as follows implying $\frac{\upsilon_h(i^*)}{\eta_h(i^*)} = \frac{\rho_\upsilon}{\rho_\eta}$:

$$\boldsymbol{\tau}_h(i^*) = \begin{bmatrix} \upsilon_h(i^*) \\ \eta_h(i^*) \end{bmatrix} = \begin{bmatrix} \rho_\upsilon \left(P_\mathrm{I} E_h^\mathrm{I}(1) + \sum_{i=2}^{i^*} P_\mathrm{P} E_h^\mathrm{P}(i) + P_\mathrm{I} E_h^\mathrm{I}(i) \right) \\ \rho_\eta \left(P_\mathrm{I} E_h^\mathrm{I}(1) + \sum_{i=2}^{i^*} P_\mathrm{P} E_h^\mathrm{P}(i) + P_\mathrm{I} E_h^\mathrm{I}(i) \right) \end{bmatrix} \tag{3.15}$$

Proposition 3.1 supports our previous argument that different pairs of the scaling parameters can lead to different optimal steady-state toll rates. In this context, we need to specify the relative weight between the two toll components. Given the link-additive property, we rescale our analysis from the path level to the link level. The generalized travel cost function for link $a \in A_\mathrm{p}$ during the h-th tolling interval is expressed as

$$c_a(h) = t_a(h) + di_a(h) + ti_a(h) \tag{3.16}$$

where $di_a(h) = \upsilon_h l_a$ and $ti_a(h) = \eta_h t_a(h)$ are the incurred distance and time tolls. Assuming the relative weight between the optimal steady-state toll components for link $a \in$

A_{p} during the h-th tolling interval is $\omega_1 > 0$, i.e. $\frac{di_a^*(h)}{ti_a^*(h)} = \omega_1$, the following equality should hold:

$$\frac{v_h(i^*)}{\eta_h(i^*)} = \frac{\rho_v}{\rho_\eta} = \frac{\omega_1 t_a(h)}{l_a} \tag{3.17}$$

However, since $t_a(h)$ varies over time, $\frac{\rho_v}{\rho_\eta} \neq \frac{\omega_1 t_a(h)}{l_a}$. Therefore, instead of focusing on an individual link, we take and assume the average of $\frac{di_a^*(h)}{ti_a^*(h)}$ over all links and tolling intervals to be ω_1:

$$\frac{\sum_{h=1}^{m} \sum_{a \in A_{\mathrm{p}}} \frac{di_a^*(h)}{ti_a^*(h)}}{mN_{\mathrm{p}}} = \omega_1 \tag{3.18}$$

This is an intuitive network-level analogue to the invalid link-level assumption. Since $\frac{di_a^*(h)}{ti_a^*(h)} = \frac{v_h(i^*) l_a}{\eta_h(i^*) t_a(h)} = \frac{\rho_v l_a}{\rho_\eta t_a(h)} = \frac{\rho_v v_a(h)}{\rho_\eta}$ where $v_a(h)$ is the average speed on link $a \in A_{\mathrm{p}}$ during the h-th tolling interval, Equation (3.18) can be rewritten as

$$\frac{\rho_v}{\rho_\eta} = \frac{\omega_1 m N_{\mathrm{p}}}{\sum_{h=1}^{m} \sum_{a \in A_{\mathrm{p}}} v_a(h)} = \frac{\omega_1 m}{\sum_{h=1}^{m} \bar{v}(h)} = \frac{\omega_1}{\bar{\bar{v}}} \tag{3.19}$$

where $\bar{v}(h) = \frac{1}{N_{\mathrm{p}}} \sum_{a \in A_{\mathrm{p}}} v_a(h)$ is the average speed in the PZ during the h-th tolling interval and $\bar{\bar{v}} = \frac{1}{m} \sum_{h=1}^{m} \bar{v}(h)$ is the average of $\bar{v}(h)$ over all tolling intervals. Given that P_{P} and P_{I} are adjustable, we set $\rho_v = 1$ and end up with $\rho_\eta = \frac{\bar{\bar{v}}}{\omega_1}$. Although $\bar{\bar{v}}$ is not

computable as prior knowledge of $v_h(i^*)$ and $\eta_h(i^*)$ is required, different tolls may result in similar network speeds because the control logic is always to keep the maximum or the average network density within the tolling period around the critical threshold identified from the NFD. We therefore approximate $\bar{\bar{v}}$ using speeds obtained under the optimal cordon toll scenario, denoted by $\tilde{v}$. Without loss of generality, we set $\omega_1 = 1$. Validity of the approximation and a sensitivity analysis on ω_1 are provided in CHAPTER 4.

3.3.2. Sequential Feedback Control Approach

Same as the JDTT, the JDDT also has an additional control input to the network other than v_h, namely the delay toll rate, ξ_h. However, we cannot readily apply the simultaneous approach because $t_a(h)$ is replaced by $t_a(h) - t_a^{\mathrm{f}}$ implying that $v_a(h)$ in Equation (3.19) no longer exists and the NFD-enabled approximation no longer holds. Therefore, we propose the following sequential approach based on the PI controller for iteratively adjusting the JDDT.

Analogous to the introduction of ω_1 when dealing with the JDTT, we introduce another weight coefficient, $\omega_2 > 0$, to break down the simultaneous TLP to be solved in a sequential manner. Specifically, at the first step, we set $\xi_h = 0$ and apply Equation (2.3) to obtain the optimal steady-state distance toll rate $v_h(i^*)$. This is virtually the optimal solution to the distance only toll problem. At the second step, given $v(i^*)$ and ω_2, we fix $v_h = \omega_2 v_h(i^*)$ and obtain the optimal steady-state delay toll rate $\xi_h(i^*)$ as follow:

$$\xi_h(i) = \begin{cases} \xi_h(i-1) + P_{\mathrm{P}}^{\xi}\left(K_h^{\max}(i) - K_h^{\max}(i-1)\right) + P_{\mathrm{I}}^{\xi}\left(K_h^{\max}(i) - K_{\mathrm{cr}}\right), & i > 1 \\ P_{\mathrm{I}}^{\xi}\left(K_h^{\max}(i) - K_{\mathrm{cr}}\right), & i = 1 \end{cases} \tag{3.20}$$

where $P_{\mathrm{P}}^{\xi}, P_{\mathrm{I}}^{\xi} > 0$ are the proportional and integral gain parameters for ξ_h, respectively. Without loss of generality, we set $\omega_2 = 0.5$. A sensitivity analysis on ω_2 is provided in CHAPTER 4.

3.4. Surrogate-Based Optimization

To facilitate the presentation, the variables used in this section are first summarized in Table 3.4.

Table 3.4 Variables used in Section 3.4

Notation	Interpretation
μ	Unknown constant mean of the response surface
σ^2	Process variance
$\boldsymbol{\theta}$	Vector of scaling coefficients
λ	Regularization constant
$y_{\min}$	Best observed objective function value so far
$c_{\max}$	Constraint threshold

While the PI controller is an intuitive and easy-to-implement method, there are several methodological disadvantages that prevent it from being adopted in a wider range of applications. As one of the demanding requirements for applying the method, the objective function needs to focus on and minimize the error from a set point simply because of the nature of the PI controller – it aims to drive a system towards a user-defined optimal state represented by a set point. The method also requires that the TLP under consideration involve simple bound constraints only but not complex constraints, a prerequisite that can be easily violated. Therefore, to solve an expensive TLP in its general formulation, we resort to the surrogate-based optimization approach. In what follows, we focus, respectively, on the four key components in the surrogate-based SO framework consisting

of DOE in Sub-section 3.4.1, regressing kriging (RK) in Sub-section 3.4.2, EI sampling in Sub-section 3.4.3, and model validation in Sub-section 3.4.4.

3.4.1. Design of Experiments (DOE)

Since DOE aims to provide an initial set of sample points to construct the starting surrogate model, the space-filling property is desirable as the resultant sample points are spread as uniformly as possible over the entire feasible domain. Latin Hypercube Sampling (LHS) is a space-filling DOE whereby each problem dimension is stratified into an equal number of intervals from which points are uniformly sampled. As such, there is no overlap in LHS when mapping the multi-dimensional sample points into each dimension. To achieve the maximum uniformity or space-fillingness of an LHS plan, one can apply maximin LHS to maximize the minimum distance between all the sample points by generating and evaluating a set of candidate plans (Forrester et al., 2008). The size of the initial set of sample points is chosen to be $2(2m+1)$ where $2m$ is the problem dimension, i.e. the size of the complete toll decision vector. According to Ekström et al. (2016), at least $2m+1$ sample points are required to construct the starting surrogate model. A few additional sample points can also be considered as part of the initial plan such as the corner and center points of the design space.

3.4.2. Regressing Kriging (RK)

In a stochastic process approach, the output of a deterministic computer experiment is modeled as a realization of a stochastic process. The ordinary kriging model is a stochastic process model that assumes an unknown constant mean, μ, of the response surface, $y(\mathbf{x})$, and a zero-mean second-order stationary Gaussian process, Z:

$$y(\mathbf{x}) = \mu + Z(\mathbf{x}), \qquad \mathrm{E}[Z(\mathbf{x})] = 0 \tag{3.21}$$

where $\mathbf{x} = [x_1, x_2, \dots, x_k]^{\mathrm{T}}$ is the decision vector. The covariance function of Z between any two points, $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, is defined as

$$\mathrm{Cov}\left[Z\left(\mathbf{x}^{(i)}\right), Z\left(\mathbf{x}^{(j)}\right)\right] = \sigma^2 \psi\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) \tag{3.22}$$

where σ^2 is the process variance and $\psi(\cdot)$ is the Gaussian correlation function depending on the distance between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ only:

$$\psi\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \exp\left(-\sum_{l=1}^{k} \theta_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right), \qquad \theta_l \geq 0 \tag{3.23}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_k]^{\mathrm{T}}$ is a vector of scaling coefficients that allows for varying impacts of each dimension on the correlation function. The correlation matrix, $\boldsymbol{\Psi}$, is constructed with the (i, j)-th element being $\psi\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right)$.

The ordinary kriging model is an interpolation method that constructs the response surface by passing through all the sample points. When computer experiments display numerical noise, i.e., the output tend to have a random scatter about a smooth trend rather than lying on it, the interpolating kriging model may exhibit overfitting without being able to tolerate data fluctuations (Forrester et al., 2006). The solution is to allow the kriging model not to interpolate but to regress the sample points, which is achieved by adding a regularization constant, λ, to the diagonal of the correlation matrix. That is, $\mathbf{R} = \boldsymbol{\Psi} + \lambda \mathbf{I}$ where $\mathbf{R}$ is known as the regressing correlation matrix and $\mathbf{I}$ is an identity matrix of the same dimension. The resultant model is commonly known as RK or kriging regression (Chen et al., 2014; Forrester et al., 2006; He et al., 2017).

Given the assumption of a Gaussian process, the likelihood function of n observations, $\mathbf{y} = [y_1, y_2, \dots, y_n]^{\mathrm{T}}$, is expressed as

$$L(\mathbf{y}|\mu, \sigma^2, \lambda, \boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{n}{2}}(\sigma^2)^{\frac{n}{2}}|\mathbf{R}|^{\frac{1}{2}}} \exp\left(-\frac{(\mathbf{y}-\mathbf{1}\mu)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2}\right) \quad (3.24)$$

where $\mathbf{1}$ is a unit column of size n and $|\cdot|$ is the determinant operator. The unknown parameters, μ, σ^2, λ, and $\boldsymbol{\theta}$, can be estimated by maximizing the logarithm of L:

$$\max_{\mu,\sigma^2,\lambda,\boldsymbol{\theta}} \log(L) = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2}\log(|\mathbf{R}|) - \frac{(\mathbf{y}-\mathbf{1}\mu)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\mu)}{2\sigma^2} - \frac{n}{2}\log(2\pi) \quad (3.25)$$

where the constant term, $\frac{n}{2}\log(2\pi)$, can be ignored. By setting the first-order derivatives to zero with respect to μ and σ^2, respectively, we obtain the maximum likelihood estimates (MLEs):

$$\hat{\mu} = \frac{\mathbf{1}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{1}} \quad (3.26)$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y}-\mathbf{1}\hat{\mu})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathbf{1}\hat{\mu})}{n} \quad (3.27)$$

Substituting the MLEs into $\log(L)$ results in what is called the concentrated log-likelihood function which is to be maximized with respect to λ and $\boldsymbol{\theta}$:

$$\text{clog}(L) = -\frac{n}{2}\log(\hat{\sigma}^2) - \frac{1}{2}\log(|\mathbf{R}|) \tag{3.28}$$

The kriging predictor for a new point, $\mathbf{x}^*$, is determined by calculating and maximizing the augmented log-likelihood function (Forrester et al., 2006):

$$\hat{y}(\mathbf{x}^*) = \hat{\mu} + \boldsymbol{\psi}^{\mathrm{T}}\mathbf{R}^{-\mathbf{1}}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{3.29}$$

where $\boldsymbol{\psi} = \left[\psi\left(\mathbf{x}^*, \mathbf{x}^{(1)}\right), \psi\left(\mathbf{x}^*, \mathbf{x}^{(2)}\right), \ldots, \psi\left(\mathbf{x}^*, \mathbf{x}^{(n)}\right)\right]^{\mathrm{T}}$ is the correlation vector between $\mathbf{x}^*$ and all the sample points. The associated prediction error is

$$\hat{s}^2(\mathbf{x}^*) = \hat{\sigma}^2\left(1 + \hat{\lambda} - \boldsymbol{\psi}^{\mathrm{T}}\mathbf{R}^{-\mathbf{1}}\boldsymbol{\psi}\right) \tag{3.30}$$

3.4.3. Expected Improvement (EI) Sampling

When kriging is used to approximate the simulation input-output mapping, additional infill sample points are required to enhance the constructed response surface. In general, there are two categories of infill strategies (Ekström et al., 2016):

- One-stage infill strategies which search for infill sample points according to a certain merit function, e.g. maximizing the minimum distance between all the sample points, without using information about the constructed response surface
- Two-stage infill strategies which search for infill sample points by utilizing the constructed response surface

We choose a two-stage infill strategy given its self-learning mechanism – the new response surface is iteratively augmented based on its predecessor. Specifically, we apply

a global optimal infill strategy known as EI sampling as opposed to a suboptimal infill strategy that balances poorly between exploring unvisited regions and exploiting visited regions (Chen et al., 2014). While trying to locate infill sample points that lead to low predictor values for a minimization problem, EI sampling also considers uncertainty about the constructed response surface as reflected by the prediction error. In regions with few sample points, although the current prediction may not be promising, the error is likely to be high suggesting a good opportunity to improve the current best solution by adding infill sample points. Therefore, as a global search method, EI sampling can balance well between local exploitation and global exploration (Forrester et al., 2008).

Unconstrained EI Sampling

Unconstrained EI sampling only considers maximizing the EI of the objective when adding infill sample points. Let $y_{\min}$ denote the best observed objective function value so far. The improvement at a new infill sample point, $\mathbf{x}^*$, is defined as $I(\mathbf{x}^*) = \max(y_{\min} - y(\mathbf{x}^*), 0)$. Knowing that $y(\mathbf{x}^*) \sim N\left(\hat{y}(\mathbf{x}^*), \hat{s}^2(\mathbf{x}^*)\right)$, the EI at this point reads $\mathrm{E}[I(\mathbf{x}^*)] = \mathrm{E}[\max(\mathrm{y}_{\min} - \mathrm{y}(\mathbf{x}^*), 0)]$. When $\hat{s}^2(\mathbf{x}^*) = 0$, $\mathrm{E}[I(\mathbf{x}^*)] = 0$; when $\hat{s}^2(\mathbf{x}^*) > 0$,

$$\mathrm{E}[I(\mathbf{x}^*)] = \frac{1}{\sqrt{2\pi}\hat{s}^2(\mathbf{x}^*)} \int_{-\infty}^{y_{\min}} (y_{\min} - u) \exp\left(-\frac{\left(u - \hat{y}(\mathbf{x}^*)\right)^2}{2\hat{s}^2(\mathbf{x}^*)}\right) du \qquad (3.31)$$

When using ordinary kriging, both the prediction error and the EI stay at zero for all the existing sample points. It is therefore impossible to add an infill sample point that has already been sampled. However, when using RK, $\hat{s}^2(\mathbf{x}^*) = 0$ does not hold at an existing sample point resulting in the possibility of maximizing $\mathrm{E}[I(\mathbf{x}^*)]$ at a previously

sampled point. To prevent RK from getting trapped at an existing sample point, Forrester et al. (2006) proposed a reinterpolation MLE of σ^2:

$$\hat{\sigma}_{\mathrm{ri}}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^{\mathrm{T}}\mathbf{R}^{-1}\boldsymbol{\Psi}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{n} \tag{3.32}$$

The associated reinterpolation prediction error hence reads $\hat{s}_{\mathrm{ri}}^2(\mathbf{x}^*) = \hat{\sigma}_{\mathrm{ri}}^2(1 - \boldsymbol{\psi}^{\mathrm{T}}\mathbf{R}^{-1}\boldsymbol{\psi})$. Now, $\hat{s}_{\mathrm{ri}}^2(\mathbf{x}^*) = 0$ holds for all the existing sample points for which $\mathrm{E}[I_{\mathrm{ri}}(\mathbf{x}^*)] = 0$. When $\hat{s}_{\mathrm{ri}}^2(\mathbf{x}^*) > 0$ and assuming $y(\mathbf{x}^*) \sim N\left(\hat{y}(\mathbf{x}^*), \hat{s}_{\mathrm{ri}}^2(\mathbf{x}^*)\right)$,

$$\mathrm{E}[I_{\mathrm{ri}}(\mathbf{x}^*)] = \frac{1}{\sqrt{2\pi}\hat{s}_{\mathrm{ri}}^2(\mathbf{x}^*)}\int_{-\infty}^{y_{\min}} (y_{min} - u)\exp\left(-\frac{\left(u - \hat{y}(\mathbf{x}^*)\right)^2}{2\hat{s}_{\mathrm{ri}}^2(\mathbf{x}^*)}\right)du \tag{3.33}$$

Constrained EI Sampling

While maximizing the EI of the objective, constrained EI sampling further considers the impact of the constraint on adding infill sample points. Let $c(\mathbf{x})$ denote the response surface of the constraint to be lower than a certain threshold, $c_{\max}$. The constrained improvement at a new infill sample point, $\mathbf{x}^*$, is defined as

$$CI(\mathbf{x}^*) = \begin{cases} I(\mathbf{x}^*), & c(\mathbf{x}^*) \le c_{\max} \\ 0, & c(\mathbf{x}^*) > c_{\max} \end{cases} \tag{3.34}$$

If the constraint is violated at $\mathbf{x}^*$, i.e. $c(\mathbf{x}^*) > c_{\max}$, $CI(\mathbf{x}^*)$ is zero even if $y_{\min} - y(\mathbf{x}^*)$ is large. Same as $y(\mathbf{x}^*)$, $c(\mathbf{x}^*) \sim N\left(\hat{c}(\mathbf{x}^*), \hat{s}_c^2(\mathbf{x}^*)\right)$ where $\hat{c}(\mathbf{x}^*)$ and $\hat{s}_c^2(\mathbf{x}^*)$ are the kriging predictor and the prediction error of the constraint at $\mathbf{x}^*$. The constrained EI therefore reads $\mathrm{E}[CI(\mathbf{x}^*)] = \mathrm{E}[I(\mathbf{x}^*)]\mathrm{P}[c(\mathbf{x}^*) \le c_{\max}]$ where $\mathrm{P}[c(\mathbf{x}^*) \le c_{\max}]$ is the

probability of not violating the constraint. The constrained EI is large only if the EI of the objective and the probability of not violating the constraint are both large. With reinterpolation, we end up with $\mathrm{E}[CI_{\mathrm{ri}}(\mathbf{x}^*)] = \mathrm{E}[I_{\mathrm{ri}}(\mathbf{x}^*)]\mathrm{P}_{\mathrm{ri}}[c(\mathbf{x}^*) \leq c_{\max}]$ where

$$\mathrm{P}_{\mathrm{ri}}[c(\mathbf{x}^*) \leq c_{\max}] = \frac{1}{\sqrt{2\pi}\hat{s}^2_{\mathrm{cri}}(\mathbf{x}^*)} \int_{-\infty}^{c_{\max}} \exp\left(-\frac{\left(u - \hat{c}(\mathbf{x}^*)\right)^2}{2\hat{s}^2_{\mathrm{cri}}(\mathbf{x}^*)}\right) du \qquad (3.35)$$

3.4.4. Model Validation

To validate the accuracy of the constructed surrogate model, one option is to select a few additional sample points to form a test set based on which the observed and predicted objective function values are compared. The training set obviously includes the initial sample points and those added as infill sample points. This option, however, is not desirable particularly when concern about the extra computational effort prevails. A better option which has been adopted in a few relevant studies (Chen et al., 2014; Ekström et al., 2016) is to leave out one observation and predict it based on the remaining observations. This procedure is commonly known as the leave-one-out cross validation (CV) which requires no additional sample points to validate the accuracy of the model.

With a total of n observations, the leave-one-out CV is repeated n times. Each time it produces a cross-validated prediction, $\hat{y}_{-i}\left(\mathbf{x}^{(i)}\right)$, for the corresponding observation, $y\left(\mathbf{x}^{(i)}\right)$, where $i \in (1,2,\dots,n)$. While common measures of effectiveness (MOEs) can be calculated to reflect the prediction accuracy, they are inappropriate for evaluating the surrogate model as the prediction at any point is a normally distributed random variable rather than a scalar (Chen et al., 2014). Knowing that, along with the cross-validated prediction, we also obtain a cross-validated standard error, $\hat{s}_{-i}\left(\mathbf{x}^{(i)}\right)$, we can calculate the

99.7% confidence interval for each $y(\mathbf{x}^{(i)})$ using the prediction plus or minus three standard errors (Jones et al., 1998). Alternatively, we can calculate $\frac{y(\mathbf{x}^{(i)})-\hat{y}_{-i}(\mathbf{x}^{(i)})}{\hat{s}_{-i}(\mathbf{x}^{(i)})}$ to obtain a standardized cross-validated residual, the value of which should be roughly lying within $[-3,3]$ for an accurate surrogate model. Unfortunately, there is no clean proof of convergence for the surrogate model. A practical technique is to track the convergence history.

3.5. Network Partitioning

To facilitate the presentation, the variables used in this section are first summarized in Table 3.5.

Table 3.5 Variables used in Section 3.5

Notation	Interpretation
$S_{\mathrm{K}}^{i}/S_{\mathrm{D}}^{i}$	Density/Distance similarity measure of link i
$\bar{S}_{\mathrm{K}}/\bar{S}_{\mathrm{D}}$	Average density/distance similarity measure of the PZ
$p_{\mathrm{K}}/p_{\mathrm{D}}$	Density/distance scaling parameter
$\tilde{S}_{\mathrm{K}}/\tilde{S}_{\mathrm{D}}$	Density/distance threshold
$d_{i,j}$	Shortest path distance between links i and j
$d_{\max,j^*}$	Maximum shortest path distance between link j^* and any other link
θ	Weight coefficient
$\mathbf{H}$	Clustering assignment matrix
$\mathbf{W}$	Composite similarity matrix
$\mathbf{W}_{\mathrm{K}}/\mathbf{W}_{\mathrm{D}}$	Density/distance similarity matrix
ϵ	Small perturbation on θ
$E_{\mathrm{Y}}/E_{\mathrm{N}}$	Set of links with and without density data

Under the assumption that the network has only one congested city center and that the PZ is single-layered, we aim to solve the TAP by partitioning the network into two

regions using link density data. The TAP to be solved is essentially a multi-objective optimization problem with conflicting objectives: (i) homogeneity, (ii) connectivity, and (iii) compactness. When more links with high densities are included in the partitioned PZ, homogeneity naturally increases but connectivity and compactness decrease. We emphasize that connectivity and compactness are two different concepts in graph theory. A connected graph implies that each pair of nodes can be reached through at least one sequence of edges, while a compact graph is one in which nodes and edges are closely arranged in space. Nevertheless, we consider connectivity and compactness as a single objective by introducing a distance similarity measure that can fulfil both requirements simultaneously. The trade-off between this composite objective and homogeneity is modeled and solved by introducing and optimizing a weight coefficient.

The proposed network partitioning framework consists of four major steps. Sub-section 3.5.1 defines similarity measures and the similarity matrix. Sub-section 3.5.2 introduces symmetric nonnegative matrix factorization (SymNMF) for network partitioning. Sub-section 3.5.3 proposes a heuristic hierarchical search algorithm (HSA) for identifying the most significant solutions from the Pareto front. Sub-section 3.5.4 extends the methodology to consider missing data.

3.5.1. Similarity Measures and the Similarity Matrix

Network partitioning requires defining a similarity measure between each pair of links in the network. Since the objective is to obtain a cluster of links covering the congested city center considering homogeneity as well as connectivity and compactness, we define two similarity measures for each link i in the network, i.e. a density similarity measure, S_{K}^{i}, and a distance similarity measure, S_{D}^{i}. Let $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$ denote the average values of S_{K}^{i} and S_{D}^{i} over all links in the partitioned PZ, respectively. A larger $\bar{S}_{\mathrm{K}}$ implies better homogeneity while a larger $\bar{S}_{\mathrm{D}}$ implies better connectivity and compactness.

Assuming a density threshold, $\tilde{S}_{\mathrm{K}}$, beyond which a link is considered congested, S_{K}^{i} is calculated as

$$S_{\mathrm{K}}^{i} = \begin{cases} 1, & k_i \geq \tilde{S}_{\mathrm{K}} \\ \left(\frac{k_i}{\tilde{S}_{\mathrm{K}}}\right)^{p_{\mathrm{K}}}, & 0 \leq k_i < \tilde{S}_{\mathrm{K}} \end{cases} \tag{3.36}$$

where $p_{\mathrm{K}} > 0$ is a density scaling parameter. Equation (3.36) has similar functionality to the Gaussian probability distribution function used in Ji and Geroliminis (2012), but it provides more flexibility through the use of p_{K}. Equation (3.36) shows that (i) $S_{\mathrm{K}}^{i} \in [0,1]$, (ii) S_{K}^{i} is monotonically increasing when $0 \leq k_i < \tilde{S}_{\mathrm{K}}$, and (iii) a larger p_{K} results in a lower increase rate and hence a higher penalty for dissimilarity.

To calculate S_{D}^{i}, we model each link i in the network as a node i' and build its neighboring relations by means of spatial connections. This connection network is viewed as an undirected graph, G', where each node i' corresponds to link i in the original network. A two-way road segment is represented as a single node despite having two parallel links. Let $d_{i,j}$ denote the spatial distance between links i and j in the network, which is defined as the length of the shortest path between nodes i' and j' in graph G' and calculated based on the adjacency matrix and Dijkstra's algorithm. The adjacency matrix, $\mathbf{A}_{i',j'}$, is a symmetric matrix representing the neighboring relations between all pairs of nodes i' and j' in graph G'. $A_{i',j'} = 1$ implies that nodes i' and j' are adjacent and vice versa. The shortest path between nodes i' and j' is the minimum number of edges to traverse from one to the other. To guarantee a spatially connected and compact PZ partitioned from the original network, we randomly choose a congested link, j^*, located at the city center as the source node in graph G' and apply Dijkstra's algorithm to build a shortest path tree

from the source node to all the other nodes. By setting a distance threshold, $\tilde{S}_{\mathrm{D}}$, as an indicator for spatial coverage, S_{D}^{i} is calculated as

$$S_{\mathrm{D}}^{i}=\begin{cases}1, & 0 \le d_{i,j^*} \le \tilde{S}_{\mathrm{D}} \\ \left(\dfrac{d_{\max,j^*}-d_{i,j^*}}{d_{\max,j^*}-\tilde{S}_{\mathrm{D}}}\right)^{p_{\mathrm{D}}}, & d_{i,j^*} > \tilde{S}_{\mathrm{D}}\end{cases} \tag{3.37}$$

where $d_{\max,j^*}$ is the maximum shortest path distance between link j^* and any other link in the network, d_{i,j^*} is the shortest path distance between link j^* and link i, and $p_{\mathrm{D}} > 0$ is a distance scaling parameter. Equation (3.37) shows that (i) $S_{\mathrm{D}}^{i} \in [0,1]$, (ii) S_{D}^{i} is monotonically decreasing when $d_{i,j^*} > \tilde{S}_{\mathrm{D}}$, and (iii) a larger p_{D} results in a higher decrease rate and hence a higher penalty for dissimilarity.

When network partitioning is performed with S_{D}^{i} as the only input similarity measure, the extracted cluster consists of links that are closely located within or around the area specified by $\tilde{S}_{\mathrm{D}}$, suggesting that compactness is explicitly guaranteed.

Proposition 3.2 The introduction of S_{D}^{i} implicitly guarantees the connectivity of links in the extracted cluster.

Proof. Without loss of generality, let us assume that there is an isolated link, l, in the extracted cluster while all the other links are connected. Using Dijkstra's algorithm to build the shortest path tree, the shortest path distance between links j^* and l should always be larger than that between link j^* and any middle link, i, along this shortest path:

$$d_{l,j^*} > d_{i,j^*}, \qquad \forall i \in r(l,j^*), i \neq l, i \neq j^* \tag{3.38}$$

where $r(l, j^*)$ is the shortest path between links j^* and l. According to Equation (3.37), the following inequality holds:

$$S_{\mathrm{D}}^{l} \leq S_{\mathrm{D}}^{i}, \qquad \forall i \in r(l, j^*), i \neq l, i \neq j^* \tag{3.39}$$

Therefore, the assumption of an isolated link does not hold. Since S_{D}^{i} is larger than or equal to that of the isolated link which is already included in the extracted cluster, any middle link should also belong to the extracted cluster. That is, if any link belongs to the extracted cluster, it cannot exist by itself suggesting that links in the extracted cluster are always connected.

The introduction of S_{D}^{i} can fulfil both connectivity and compactness requirements which are therefore considered as a single objective. Here, connectivity is implicitly considered by incorporating spatial information into a similarity measure (Ji and Geroliminis, 2012; Saeedmanesh and Geroliminis, 2016), rather than being explicitly modeled as a set of constraints in an optimization problem (Saeedmanesh and Geroliminis, 2017). Based on Equations (3.36) and (3.37), we define a composite similarity measure, S^i, for each link i in the network as a weighted average of S_{K}^{i} and S_{D}^{i}:

$$S^i = \theta S_{\mathrm{K}}^{i} + (1 - \theta) S_{\mathrm{D}}^{i} \tag{3.40}$$

where $\theta \in [0,1]$ is a weight coefficient. Equation (3.40) shows that (i) $S^i \in [0,1]$, (ii) S^i represents a θ-dependent trade-off between S_{K}^{i} and S_{D}^{i}, and (iii) a link with a larger S^i is more likely to be included in the partitioned PZ.

Let **W** denote the composite similarity matrix where each element, $W_{i,j}$, measures the similarity between links i and j in the network. Based on Equation (3.40), $W_{i,j}$ is calculated as

$$W_{i,j} = 1 - \left|S^i - S^j\right| \tag{3.41}$$

When $S^i - S^j \to 0$, links i and j are considered similar to each other and hence $W_{i,j} \to 1$; when $\left|S^i - S^j\right|$ increases, i.e. $S^i - S^j \to \pm 1$, dissimilarity between links i and j increases and hence $W_{i,j}$ decreases, i.e. $W_{i,j} \to 0$. The similarity matrix is a simple yet powerful representation of the network that can be used for clustering purposes.

3.5.2. Symmetric Nonnegative Matrix Factorization (SymNMF)

To cluster and analyze data, spectral clustering has been proposed in graph theory to group objects into different clusters using a similarity matrix. It focuses on the pairwise similarity measure between each pair of objects rather than looking directly at data and makes no assumption about the form of the cluster (Saeedmanesh and Geroliminis, 2016). However, given that the performance of spectral clustering is highly dependent on the eigenvalues of the Laplacian matrix (Kuang et al., 2015; Ng et al., 2002), we employ another robust clustering method termed symmetric nonnegative matrix factorization (SymNMF) for network partitioning. As an extended formulation for graph clustering based on NMF, SymNMF provides a nonnegative low-rank approximation of a similarity matrix. Studies (Kuang et al., 2012; Kuang et al., 2015) have shown that SymNMF (i) outperforms other methods such as k-means, spectral clustering, and NMF for graph clustering, and (ii) can capture the clustering structure embedded in the graph representation more naturally. SymNMF as a clustering method has been widely applied in a variety of

fields including image and document clustering (He et al., 2011), community detection (Zhang et al., 2013), and transportation network partitioning (Saeedmanesh and Geroliminis, 2016).

The objectives of various graph clustering methods are inherently consistent which can be reduced to a trace maximization form (Kulis et al., 2009):

$$\max \mathrm{Tr}(\mathbf{H}^{\mathrm{T}}\mathbf{W}\mathbf{H}), \qquad \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}, \mathbf{H} \geq 0 \tag{3.42}$$

where $\mathbf{H} \in \mathbb{R}^{n \times k}$ (normally $n \gg k$) is a clustering assignment matrix, $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a similarity matrix, and $\mathrm{Tr}(\cdot)$ is the matrix trace operator. Each column in $\mathbf{H}$ represents a cluster and hence k is the number of clusters. Each row in $\mathbf{H}$ shows the membership of each of the n objects to the k clusters. Equation (3.42) is mathematically equivalent to the following minimization problem (Kuang et al., 2015):

$$\min \|\mathbf{W} - \mathbf{H}\mathbf{H}^{\mathrm{T}}\|_F^2, \qquad \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}, \mathbf{H} \geq 0 \tag{3.43}$$

where $\|\cdot\|_F$ is the Frobenius norm. Since the constraints on $\mathbf{H}$ make the problem NP-hard, both spectral clustering and SymNMF seek to relax one of the constraints to make the problem tractable – spectral clustering retains $\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}$ while SymNMF retains $\mathbf{H} \geq 0$. The physical meaning of $\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}$ in spectral clustering is that each object only belongs to a single cluster. Although each object can belong to multiple clusters through different membership values in SymNMF, Ding et al. (2005) showed that $\mathbf{H} \geq 0$ leads to $\mathbf{H}^{\mathrm{T}}\mathbf{H} \approx \mathbf{I}$.

To obtain high intra-similarity and low inter-similarity, SymNMF aims to find a nonnegative low-rank matrix $\mathbf{H} \in \mathbb{R}_{+}^{n\times k}$ approximating the given nonnegative symmetric similarity matrix $\mathbf{W} \in \mathbb{R}_{+}^{n\times n}$ by minimizing the Frobenius norm:

$$\min_{\mathbf{H}\geq 0} \|\mathbf{W} - \mathbf{H}\mathbf{H}^{\mathrm{T}}\|_F^2 \tag{3.44}$$

Since Problem (3.44) minimizes a fourth-order non-convex objective function with respect to the elements of **H**, multiple local minima may exist depending on the initialization of **H**. Therefore, one can generate different random seed numbers for initializing **H** to help locate the global minimum. Integrating random sampling with a local search algorithm formulates a multi-start global optimization approach (Rinnooy Kan and Timmer, 1989).

3.5.3. Hierarchical Search Algorithm (HSA)

The input to SymNMF is the composite similarity matrix, **W**, calculated from the composite similarity measure, S^i. Since S^i is defined as a weighted average of the density and distance similarity measures, S_{K}^i and S_{D}^i, the weight coefficient, θ, naturally plays a decisive role: a larger θ puts more weight on S_{K}^i resulting in an increased $\bar{S}_{\mathrm{K}}$ and a decreased $\bar{S}_{\mathrm{D}}$, and vice versa. To achieve the most desirable network partitioning result, the optimal θ is required. Here, "optimal" does not necessarily mean that any solution dominates the others. It only refers to a sensible trade-off between the two conflicting objectives.

Proposition 3.3. The optimal clustering assignment matrix, $\mathbf{H}^*$, remains similar for $\forall\theta \in [\theta-\epsilon, \theta+\epsilon]$ where $0 < \epsilon \leq \min(\theta, 1-\theta)$ is a small perturbation on θ.

Proof. Let $\mathbf{W}_{\mathrm{K}}$ and $\mathbf{W}_{\mathrm{D}}$ denote respectively the density and distance similarity matrices. The composite similarity matrix, $\mathbf{W}$, is expressed as

$$\mathbf{W} = \theta\mathbf{W}_{\mathrm{K}} + (1-\theta)\mathbf{W}_{\mathrm{D}} \tag{3.45}$$

With a small perturbation, $0 < \epsilon \ll 1$, on θ, the perturbed composite similarity matrix, $\mathbf{W}_{\epsilon}$, is expressed as

$$\mathbf{W}_{\epsilon} = (\theta \pm \sigma)\mathbf{W}_{\mathrm{K}} + (1-\theta \mp \sigma)\mathbf{W}_{\mathrm{D}} = \mathbf{W} \pm \sigma(\mathbf{W}_{\mathrm{K}} - \mathbf{W}_{\mathrm{D}}) \approx \mathbf{W} \tag{3.46}$$

Given $\mathbf{W}_{\epsilon} \approx \mathbf{W}$, the optimal clustering assignment matrix, $\mathbf{H}^{*}$, remains similar.

Example. Let us assume that there are three objects to be partitioned into two clusters, i.e. $n = 3$ and $k = 2$. The density and distance similarity matrices, $\mathbf{W}_{\mathrm{K}}$ and $\mathbf{W}_{\mathrm{D}}$, are as follows:

$$\mathbf{W}_{\mathrm{K}} = \begin{bmatrix} 1 & 0.8 & 0.2 \\ 0.8 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}, \qquad \mathbf{W}_{\mathrm{D}} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.8 \\ 0.2 & 0.8 & 1 \end{bmatrix} \tag{3.47}$$

The composite similarity matrix, $\mathbf{W}$, is calculated as

$$\mathbf{W} = \theta\mathbf{W}_{\mathrm{K}} + (1-\theta)\mathbf{W}_{\mathrm{D}} = \begin{pmatrix} 1 & 0.6\theta + 0.2 & 0.2 \\ 0.6\theta + 0.2 & 1 & 0.8 - 0.6\theta \\ 0.2 & 0.8 - 0.6\theta & 1 \end{pmatrix} \tag{3.48}$$

When $\theta = 1$, $\mathbf{W}(\theta = 1) = \mathbf{W}_{\mathrm{K}} = \begin{bmatrix} 1 & 0.8 & 0.2 \\ 0.8 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}$. Since the similarity measure between the first two objects, 0.8, is significantly larger than that between the last two objects, 0.2, the optimal clustering assignment is evident: the first two objects belong to a cluster and the third object forms the other cluster by itself. Assuming that θ reduces by 0.1, i.e. $\varepsilon = 0.1$, the perturbed composite similarity matrix $\mathbf{W}(\theta = 0.9) = \begin{bmatrix} 1 & 0.74 & 0.2 \\ 0.74 & 1 & 0.26 \\ 0.2 & 0.26 & 1 \end{bmatrix} \approx \mathbf{W}(\theta = 1)$. Therefore, the optimal clustering assignment matrix, $\mathbf{H}^*$, remains similar and the optimal clustering assignment does not change. In fact, for $\forall\theta \in [0.9,1]$, the optimal clustering assignment remains unchanged. Although the optimal clustering assignment does not vary continuously as θ changes, we do not know a priori how large ε can be as it is endogenously determined by $\mathbf{W}(\theta)$.

Based on Proposition 3.3, we propose a heuristic HSA to identify the significant solutions from the Pareto front which obviously includes an infinite number of Pareto efficient solutions. Here, a Pareto efficient solution is considered significant if it changes the optimal clustering assignment and achieves a large improvement in the overall similarity between links in the partitioned PZ. To this end, we adopt the concept of "knee" (Chaudhari et al., 2010) which, by definition, refers to the solutions from the Pareto front whereby a small improvement (deterioration) in one objective leads to a large deterioration (improvement) in at least one other objective. It may happen that a significant solution is not from the Pareto front, but this can be easily resolved by checking all the identified significant solutions and removing those that are not Pareto efficient.

3.5.4. Extending the Methodology to Consider Missing Data

To apply the proposed solution framework, we assume to have perfect information about traffic conditions in the network, which, however, does not necessarily hold in practice. In a real-world traffic network, it is common that some links do not have density data for some periods of time. This may happen when no sensors are installed or sensors, although installed, malfunction and cannot provide accurate measurements. We therefore extend the methodology to further consider missing data.

Given that congestion is spatially correlated in adjacent links with obvious directionality and transmissibility (Wang et al., 2017), we try to estimate link densities that are unknown based on the available densities of their upstream and downstream adjacent links of the same direction (Saeedmanesh and Geroliminis, 2016). Let $E_{\mathrm{Y}} \neq \emptyset$ and $E_{\mathrm{N}} \neq \emptyset$ denote respectively the sets of links in the network with and without density data. Let E_i denote the set of upstream and downstream adjacent links of link $i \in E_{\mathrm{N}}$ with known densities, i.e. $E_i = \left\{ j \middle| A_{i',j'} = 1, j \in E_{\mathrm{Y}}, j \text{ and } i \text{ are of the same direction} \right\}$. If $E_i = \emptyset$, we do nothing and the density of link i remains unknown; otherwise we estimate the density of link i as the average of all the available densities of its upstream and downstream adjacent links, and move link i from E_{N} to E_{Y}:

$$k_i = \frac{1}{|E_i|} \sum_{j \in E_i} k_j \tag{3.49}$$

where $|\cdot|$ is the set cardinality operator. The estimation continues in an iterative manner until $E_{\mathrm{N}} = \emptyset$. While Equation (3.49) provides an estimate of any missing link density, we emphasize that this does not necessarily represent the state of the art for traffic state estimation. Given that our focus is on network partitioning, we refer to Antoniou et al. (2013); Nantes et al. (2016); Seo et al. (2015); Tyagi et al. (2012) for perhaps more advanced

methods enabled by multi-source traffic data. However, we show in CHAPTER 7 that the extended framework performs well even with a low penetration rate.

3.6. Chapter Remarks

This chapter provides an in-depth description of the theory and methodology offered by this thesis. In Section 3.1, we briefly revisit the NFD and discuss how it can be used for pricing control and optimization. In Section 3.2, we propose two joint tolls, namely the JDTT and the JDDT, to extend the distance only toll that tends to drive travelers into the shortest paths within the PZ despite being congested. To solve the TLP, we propose two computationally efficient SO frameworks consisting of feedback control in Section 3.3 and surrogate-based optimization in Section 3.4. The feedback control approach is particularly suited for solving a simple TLP featuring a set-point objective and bound constraints only, whereas the surrogate-based optimization approach is more general and can be applied to solve any complex TLP, i.e. a TLP with either a complex objective or complex constraints. To solve the TAP, we propose a network partitioning approach in Section 3.5.

CHAPTER 4. FEEDBACK CONTROL FOR TOLL LEVEL OPTIMIZATION

This chapter provides a numerical study on the feedback control approach for solving the TLP corresponding to Section 3.3. To evaluate and compare the performance of different tolls, we use a recently developed large-scale simulation-based DTA model of Melbourne, Australia deployed in AIMSUN with time-dependent demand for the 6-10 AM peak period. Travelers are assumed to have access to real-time information for re-routing and their route choice is calculated and updated every 5 minutes. VTT is assumed to be $15/h (Legaspi and Douglas, 2015). Further details of the model can be found in APPENDIX A.

The rest of this chapter is organized as follows. Section 4.1 presents the feedback-control enabled SO framework as the solution algorithm. Sections 4.2, 4.3, and 4.4 present the numerical results for the distance only toll, the JDTT, and the JDDT, respectively. A comprehensive comparison between the JDTT and the JDDT is performed in Section 4.5. In Sections 4.6, we investigate the effect of simulation stochasticity while in Section 4.7, we elaborate on the applicability of the feedback control approach. Section 4.8 concludes the chapter. The work of this chapter has been published:

- Gu, Z., Shafiei, S., Liu, Z., Saberi, M., 2018. Optimal distance- and time-dependent area-based pricing with the Network Fundamental Diagram. *Transp. Res. Part C* 95, 1-28.

To facilitate the presentation, the variables used in this chapter are first summarized in Table 4.1.

Table 4.1 Variables used in CHAPTER 4

Notation	Interpretation
m	Number of tolling intervals
τ_h	Toll rate for the h-th tolling interval
$\tau_{\min}/\tau_{\max}$	Lower/upper bound on the toll rate
$K_h^{\max}$	Maximum network density during the h-th tolling interval
K_{cr}	Critical network density
$N_{\max}$	Maximum number of iterations allowed
$P_{\mathrm{P}}/P_{\mathrm{I}}$	Proportional/integral gain parameter
$\omega_1/\ \omega_2$	Weight coefficient

4.1. Feedback-Control Enabled Simulation Optimization (SO) Framework

Consider the following simple single-objective TLP:

$$\min_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\ldots,\boldsymbol{\tau}_m} \mathrm{E}\left[\frac{1}{m}\sum_{h=1}^{m}|K_h^{\max}-K_{\mathrm{cr}}|\right] \tag{4.1}$$

s.t.

$$K_h^{\max} = DTA(\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\ldots,\boldsymbol{\tau}_m), \qquad h=1,2,\ldots,m \tag{4.2}$$

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}_h \le \boldsymbol{\tau}_{\max}, \qquad h=1,2,\ldots,m \tag{4.3}$$

where $\mathrm{E}[\cdot]$ is the expectation operator, $DTA(\cdot)$ is the black-box function of the simulation model, and $\boldsymbol{\tau}_{\min}$ and $\boldsymbol{\tau}_{\max}$ are the lower and upper bounds on the toll rates, respectively. Given a stochastic traffic simulator using different random seed numbers, the objective function in Equation (4.1) aims to minimize the expected average of the absolute difference between $K_h^{\max}$ and K_{cr} for the m tolling intervals. As such, the network is pricing-

controlled near the critical network density without entering the congested regime of the NFD. To calculate the expectation, one can readily apply fixed-number sample path optimization, also known as sample average approximation (Amaran et al., 2016). However, in the context of a computationally expensive objective function, the sample size is usually restricted to simply reduce the effect of noise rather than pursuing a complete noise filter. To handle simulation noise in a more computationally efficient manner, one can apply variable-number sample path optimization (He et al., 2017). While Constraint (4.3) specifies the toll feasible region, Constraint (4.2) maps the input toll rates to the simulation density output which is fed back to the objective function.

Figure 4.1 illustrates the feedback-control enabled SO framework for solving Problem (4.1-4.3). The detailed algorithmic steps are as follows:

Step 1. Given a single-cordon two-region network, run the simulation without pricing to get the base scenario NFD of the PZ.

Step 2. Set $i = 1$ and determine from the base scenario NFD K_{cr} and the tolling period when network densities exceed K_{cr}.

Step 3. Calculate the initial toll rates when $i = 1$ using Equation (2.3) for the distance only toll, Equation (3.12) for the JDTT, and Equations (2.3) and (3.20) for the JDDT.

Step 4. Set $i = i + 1$ and run the simulation with the newly calculated toll rates to get the updated NFD of the PZ.

Step 5. If $i \leq N_{\max}$ where $N_{\max}$ is the maximum number of iterations allowed, calculate the updated toll rates when $i > 1$ using Equation (2.3) for the distance only toll, Equation (3.12) for the JDTT, and Equations (2.3) and (3.20) for the JDDT, and go back to Step 4; otherwise terminate the algorithm.

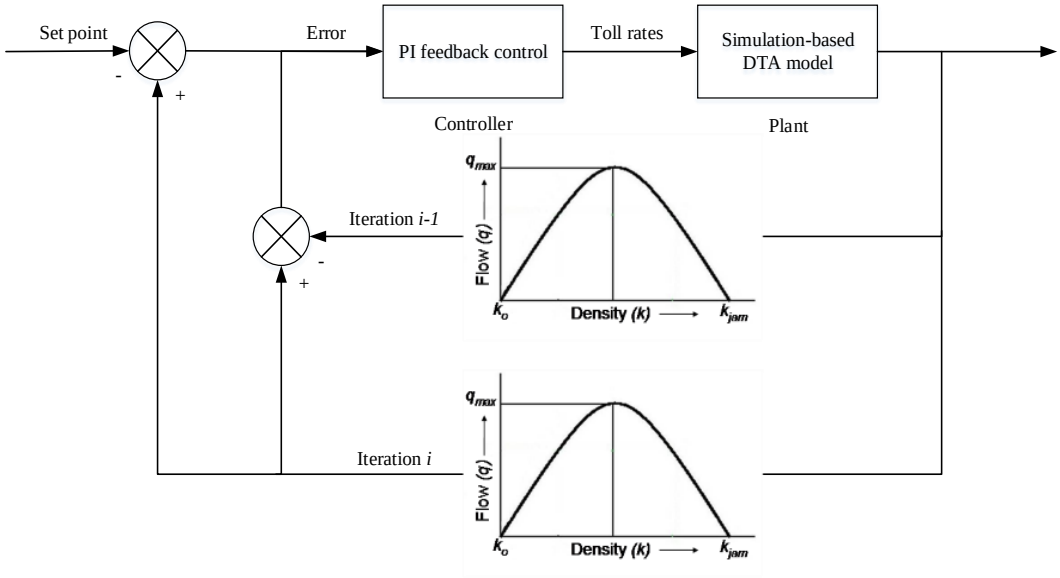


Figure 4.1 Closed-loop block diagram of the feedback-control enabled SO framework

The proposed method can be applied to solve both static and time-dependent TLPs. The biggest advantage of time-dependent pricing is that travelers are not overcharged while the control objective is still met. That is, time-dependent pricing does not over-control the network resulting in unnecessary network unproductivity. When applying the method to solve the time-dependent TLP, an independent PI controller is deployed in each tolling interval. Here, "independent" means that for the h-th tolling interval, the input measurements to the PI controller come from this specific interval only without consideration for the other intervals. The proposed method resembles but extends Zheng et al. (2016); Zheng et al. (2012) in terms of the pricing regimes investigated and how the PI controller is tailored and applied.

If the distribution of congestion in the PZ exhibits strong heterogeneity, further network partitioning (Ji and Geroliminis, 2012; Saeedmanesh and Geroliminis, 2016, 2017) can be considered. The resultant coordinated multi-area pricing scheme is a

challenging research question that remains open. The feedback control approach here targets a single PZ only and hence does not consider coordination among multiple PZs.

4.2. Distance Only Toll

We run the simulation without pricing to obtain the base NFD of the PZ based on which the critical network density, K_{cr}, and the tolling period are determined. As shown in Figure 4.2(a) and (b), the maximum network flow without pricing occurs when the network density varies between 20 and 30 vpkmpl. We therefore set $K_{\mathrm{cr}} = 25$ vpkmpl and the resultant tolling period is 40 minutes between 8:35 and 9:15 AM. To compare with the distance only toll, we apply the feedback control approach to obtain an optimal cordon toll of about \$1.9. The associated network performance is shown in Figure 4.2(a) and (b). The optimal cordon toll successfully keeps the network from entering the congested regime of the NFD and reduces the size of the hysteresis loop. This is because a portion of travelers are priced off the PZ resulting in a lower level of congestion and hence a more homogenous distribution of congestion. Figure 4.2(c) presents a sensitivity analysis on the controller gain parameters to (i) verify the global convergence of the PI controller, and (ii) provide general guidance on applying trial-and-error. Although different pairs of P_{P} and P_{I} are used, the optimal cordon toll is globally convergent. When $P_{\mathrm{P}} = 0.05$ and $P_{\mathrm{I}} = 0.4$, the oscillatory behavior of the PI controller is more significant making it difficult to pinpoint the optimum. Once the parameters are lowered, the oscillation weakens although at the cost of an increased number of iterations until convergence. Given this trade-off, a rule of thumb is to start with a slightly larger pair of P_{P} and P_{I}, and gradually decrease their values until a relatively smooth convergence pattern is achieved.

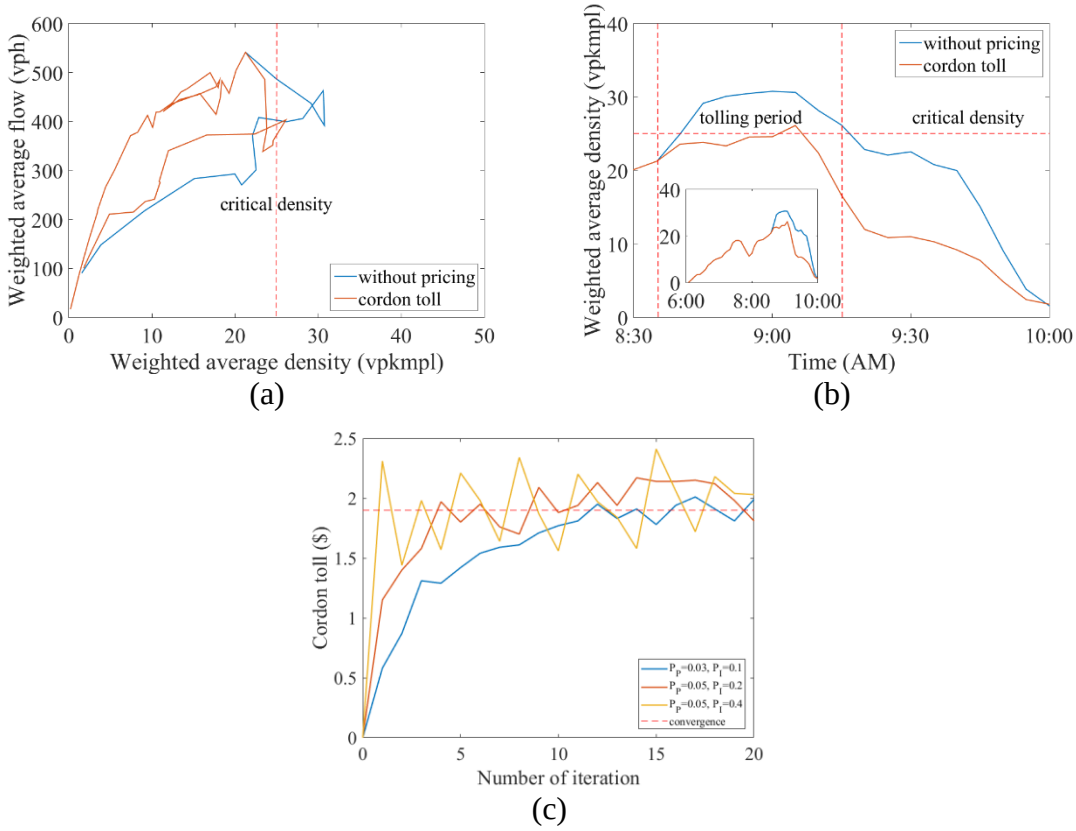


Figure 4.2 Simulation results of the PZ under the non-tolling and the optimal cordon toll scenarios: (a) simulated NFDs, (b) density time series, (c) sensitivity analysis on the controller gain parameters

The cordon toll does not consider the distance traveled within the PZ. Travelers are equally charged regardless of their actual amount of road usage. The consequent inequity may create poor public acceptance. The distance only toll, however, calculates the toll price by the trip length within the PZ rather than being pay-per-entry. It therefore distinguishes between, for example, a traveler reaching the destination immediately upon crossing the cordon and one traversing the whole PZ, thereby creating a more efficient and equitable pricing system. As shown in Figure 4.3(b), the optimal distance only toll rate is about $1/km. Although the control objective is met, the resultant NFD in Figure 4.3(a) exhibits a much larger hysteresis loop than under the non-tolling and the optimal cordon toll scenarios. When the network is unloading, i.e. recovering from congestion, the distribution of congestion tends to be more uneven as the congested areas clear more

slowly and are fragmented. This uneven distribution of congestion inevitably reduces the network flow during recovery resulting in a clockwise hysteresis loop in the NFD (Gayah and Daganzo, 2011). Given that travelers choose their routes with the least generalized travel costs, the distance only toll naturally drives them into the shortest paths within the PZ as a shorter trip length equates to a lower toll price. A dominant portion of travelers therefore travel on the same shortest paths and the distribution of congestion becomes more heterogeneous.

Figure 4.3(d) shows that the distance only toll results in a much less total distance traveled within the PZ, as expected. Figure 4.3(e) and (f) show respectively the spread-accumulation relationships and the time series of the deviation from spread to quantitatively analyze and compare the heterogeneity of congestion distribution. The fitted $\gamma(K) = -0.0003154K^3 + 0.01499K^2 + 1.127K$. Corresponding to the clockwise hysteresis loop in the NFD, an anticlockwise hysteresis loop forms in the spread-accumulation relationship, the size of which also increases under the optimal distance only toll scenario. The spatial spread of density increases sharply after applying the distance only toll and then stays at a much higher level. This finding is consistent with Simoni et al. (2015) who argued that the decrease in the network flow is caused by clusters of congestion rather than by the increase of travelers.

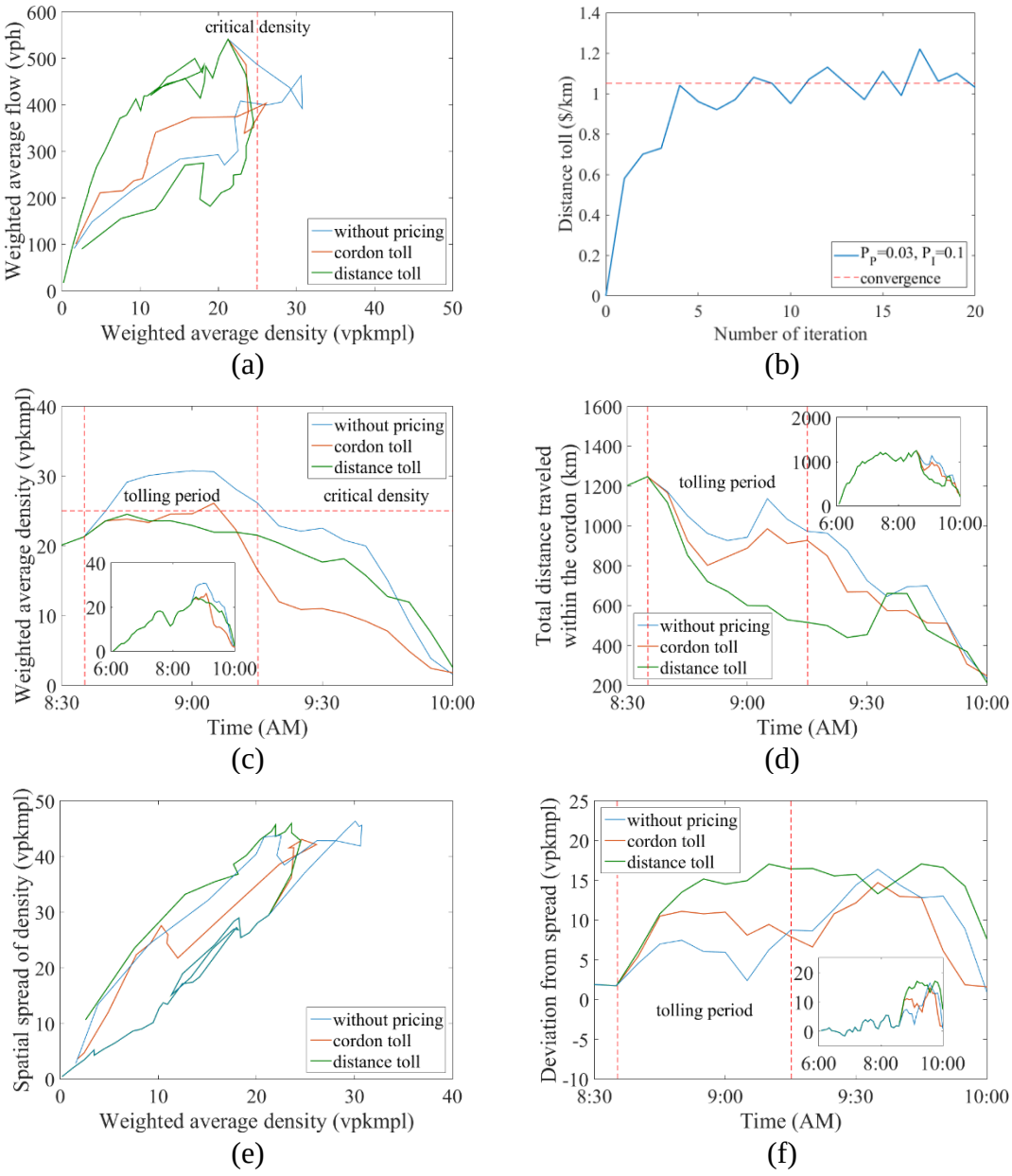


Figure 4.3 Simulation results of the PZ under the optimal distance only toll scenario: (a) simulated NFDs, (b) convergence of the distance toll rate, (c) density time series, (d) time series of the total distance traveled, (e) spread-accumulation relationships, and (f) time series of the deviation from spread

4.3. Joint Distance and Time Toll (JDTT)

The question to be answered is how we can improve the distance only toll such that the network exhibits less heterogeneous distribution of congestion and the resultant NFD has a smaller hysteresis loop. We show that travelers are driven into the shortest

paths within the PZ when charged with the distance only toll. Although the travel times on some shortest paths increase, most travelers do not change their routes as the utility from paying a lower toll price dominates the disutility from the increase in travel time. A straightforward solution is to charge travelers jointly based on the distance traveled and the time spent within the PZ. As such, travelers are more likely to distribute themselves into the second or third shortest path if the travel time on the shortest path rises considerably.

Figure 4.4(b) shows that the optimal distance toll rate is about \$0.35/km and the corresponding optimal time toll rate is \$9/h. Despite having a different order of magnitude, the time toll rate exhibits the same convergence as the distance toll rate given their linear correlation, and hence we only show for the latter. As shown in Figure 4.4(a), (g), and (h), under the optimal JDTT scenario, the size of the hysteresis loop reduces and the deviation from spread stays at a lower level. Compared with the distance only toll, the JDTT increases the total distance traveled within the PZ, as expected, because travelers no longer accumulate themselves into the shortest paths. Figure 4.4(e) and (f) further compare the average network speed and the average link queue length, respectively. There is a sudden jump in the speed profile at the beginning of the simulation as we load the network without a warm-up period. After applying different optimal tolls, the time series show that the JDTT keeps the network speed at a higher level while reducing the link queue length to the greatest extent. The distance only toll, on the other hand, results in a lower and less stable network speed as well as an increased link queue length.

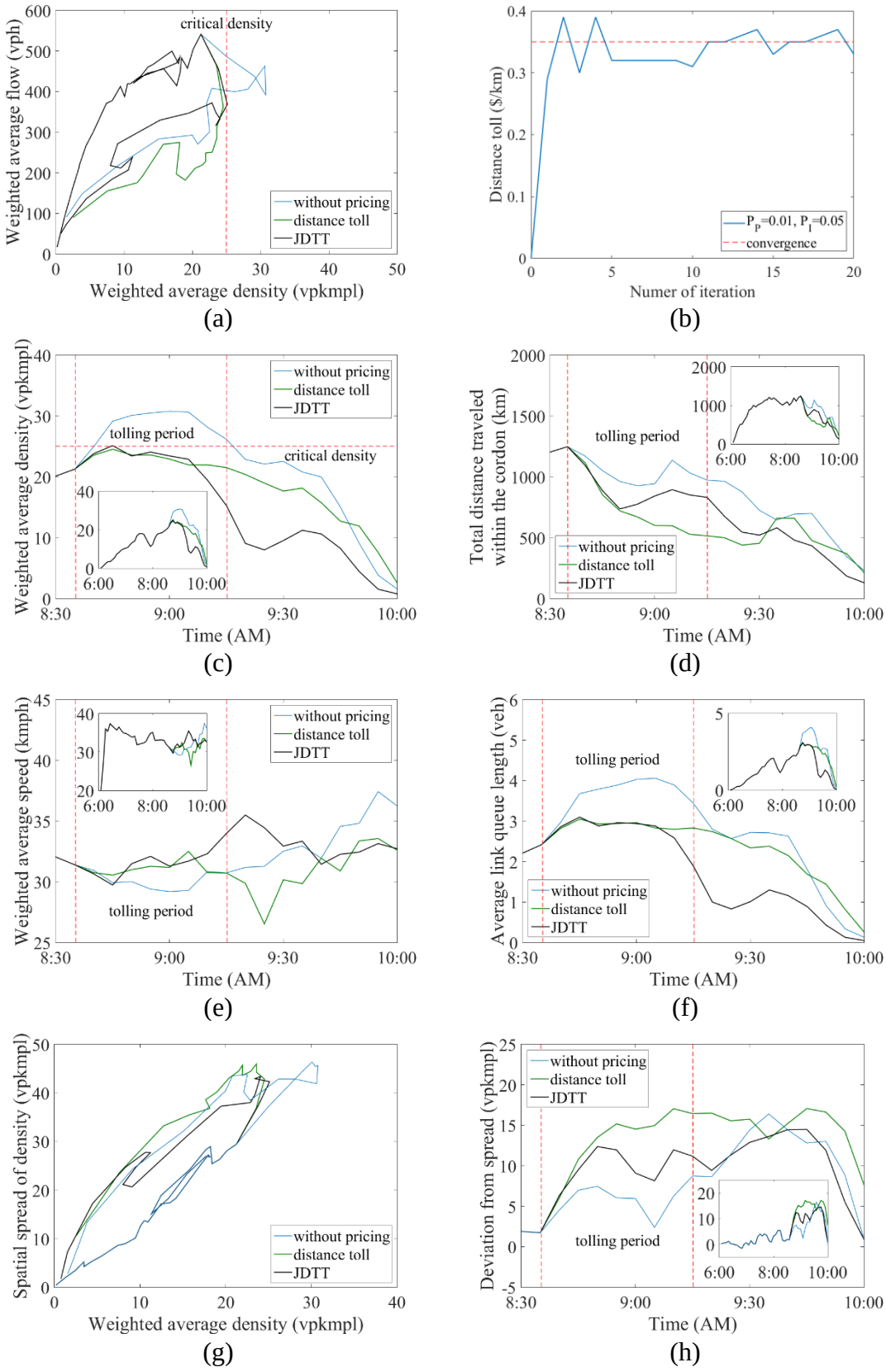


Figure 4.4 Simulation results of the PZ under the optimal JDTT scenario: (a) simulated NFDs, (b) convergence of the distance toll rate, (c) density time series, (d) time series of the total distance traveled, (e) speed time series, (f) queue time series, (g) spread-accumulation relationships, and (h) time series of the deviation from spread

The reason why the distance only toll results in a network performance degradation is because the concentration of travelers into some shortest paths within the PZ generate more and bigger clusters of congested links. As shown in Figure 4.5, the distance only toll results in a more heterogeneous distribution of congestion particularly in the bottom left corner of the PZ and part of the connected peripheral network. The spatial differences are not significant at the beginning of the tolling period, i.e. 8:40 AM, but gradually become prominent later in time towards the end of the tolling period, i.e. 9:10 AM.

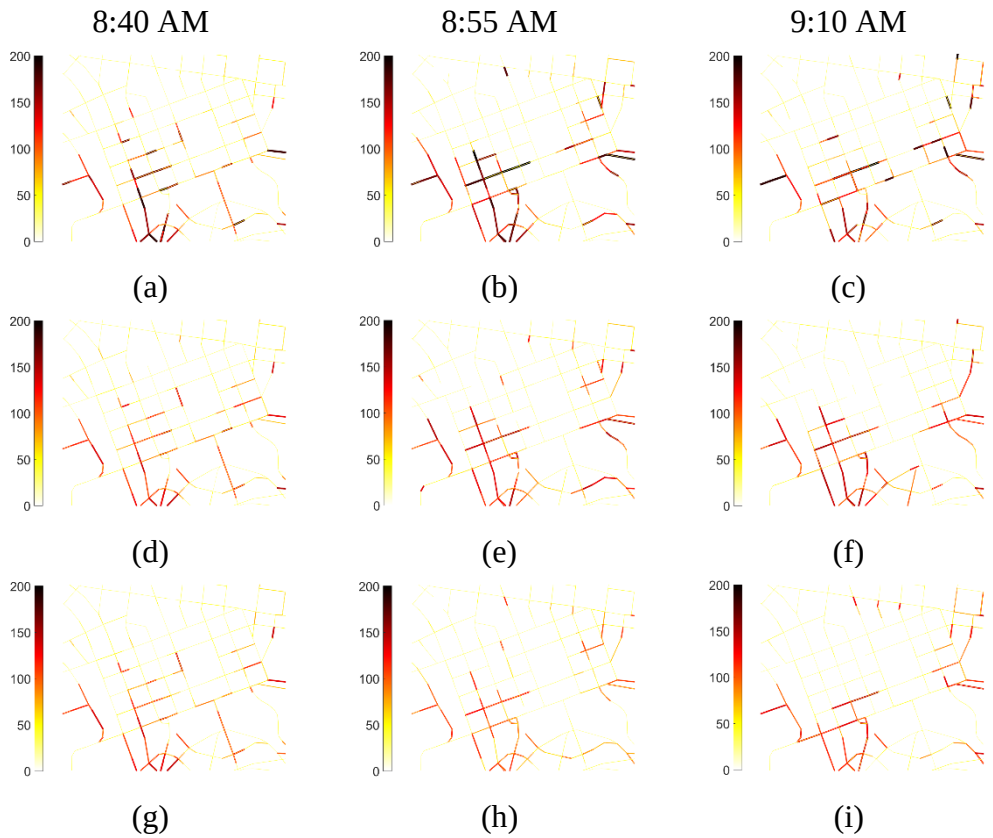


Figure 4.5 Comparing the spatiotemporal evolution of link densities within the PZ during the tolling period under different tolling scenarios: (a-c) non-tolling, (d-f) distance only toll, (g-i) JDTT

Since we approximate $\bar{\bar{v}}$ using speeds obtained under the optimal cordon toll scenario, we further calculate, under both the optimal JDTT and the optimal cordon toll scenarios, the average network speed over the tolling period and end up with $\bar{\bar{v}} = 32.37$ km/h and $\tilde{v} = 32.17$ km/h, thereby justifying the validity of the approximation. Note that if either of the toll rates hits the lower (upper) bound during an iteration, it is fixed at the minimum (maximum) during the subsequent iterations and only the other toll rate is to be adjusted. If the control objective is not met when both toll rates hit their respective upper bounds, it means that the current pricing set-up cannot drive the network to its optimal state, and that we can either simply raise the upper bounds or consider an additional TDM policy to create a mixed network control system.

4.3.1. Sensitivity Analysis on the Weight Coefficient

We perform a sensitivity analysis on ω_1 to examine its effect on the pricing control results. Three different values of ω_1 are tested, i.e. $\omega_1 = 0.33, 1, 3$. A larger value implies a more dominating role played by the distance toll component. Figure 4.6(b) and (c) show that, regardless of the value of ω_1, we consistently achieve the global convergence and the network is well controlled without entering the congested regime of the NFD. Figure 4.6(a) and (d) further show that both the size of the hysteresis loop and the deviation from spread increase when $\omega_1 = 3$. This is because when a large ω_1 is used, the JDTT resembles the distance only toll which cannot well reduce the uneven distribution of congestion. Therefore, a small value of ω_1 is advisable.

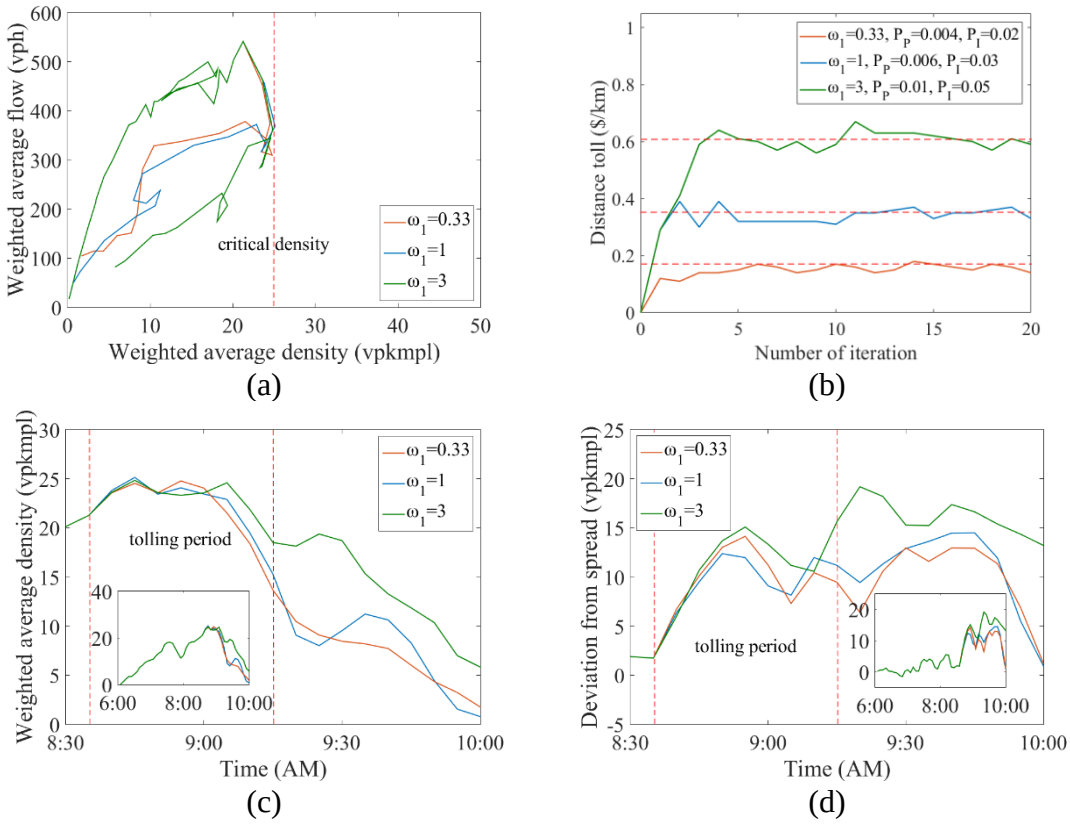


Figure 4.6 Sensitivity analysis on ω_1: (a) simulated NFDs, (b) convergence of the distance toll rate, (c) density time series, and (d) time series of the deviation from spread

4.3.2. Time-Dependency

We further apply the feedback control approach to time-dependent JDTT by dividing the tolling period into two 20-min tolling intervals. Here, the duration is simply chosen based on experience and is by no means an optimization result. Intuitively, it should not be too small considering travelers' adaption – travelers may hardly adapt to the rapidly changing toll rates and the network may become unstable. It should also not be too large considering the effectiveness of congestion management – different levels of congestion may not be well captured and distinguished.

While Figure 4.7(a) and (c) show that the optimal time-dependent JDTT effectively achieves the control objective, Figure 4.7(b) displays an interesting non-smooth convergence of the distance toll rate for the second tolling interval. This is because during

the first few iterations, the toll rates in both tolling intervals naturally increase to reduce the number of travelers entering the PZ. Due to the interplay between the two tolling intervals, an increase in the toll rate during the first interval inevitably leads to a less congested network during the second. Therefore, when the toll rate during the first interval gets close to convergence, the toll rate for the second drops.

The advantage of time-dependence is that travelers are not overcharged as the toll price varies according to the changing level of congestion. If the tolling period is long enough to capture different levels of congestion, this advantage is even more significant. Compared with the fixed JDTT, the time-dependent JDTT is less conservative by allowing more travelers to enter the PZ while still achieving the control objective. This is reflected in Figure 4.7(e) where the time-dependent JDTT results in a larger number of vehicles entering the PZ during the tolling period, and the shaded area certainly represents the difference. Also, as shown in Figure 4.7(f), the time-dependent JDTT allows for a larger total distance traveled in the PZ.

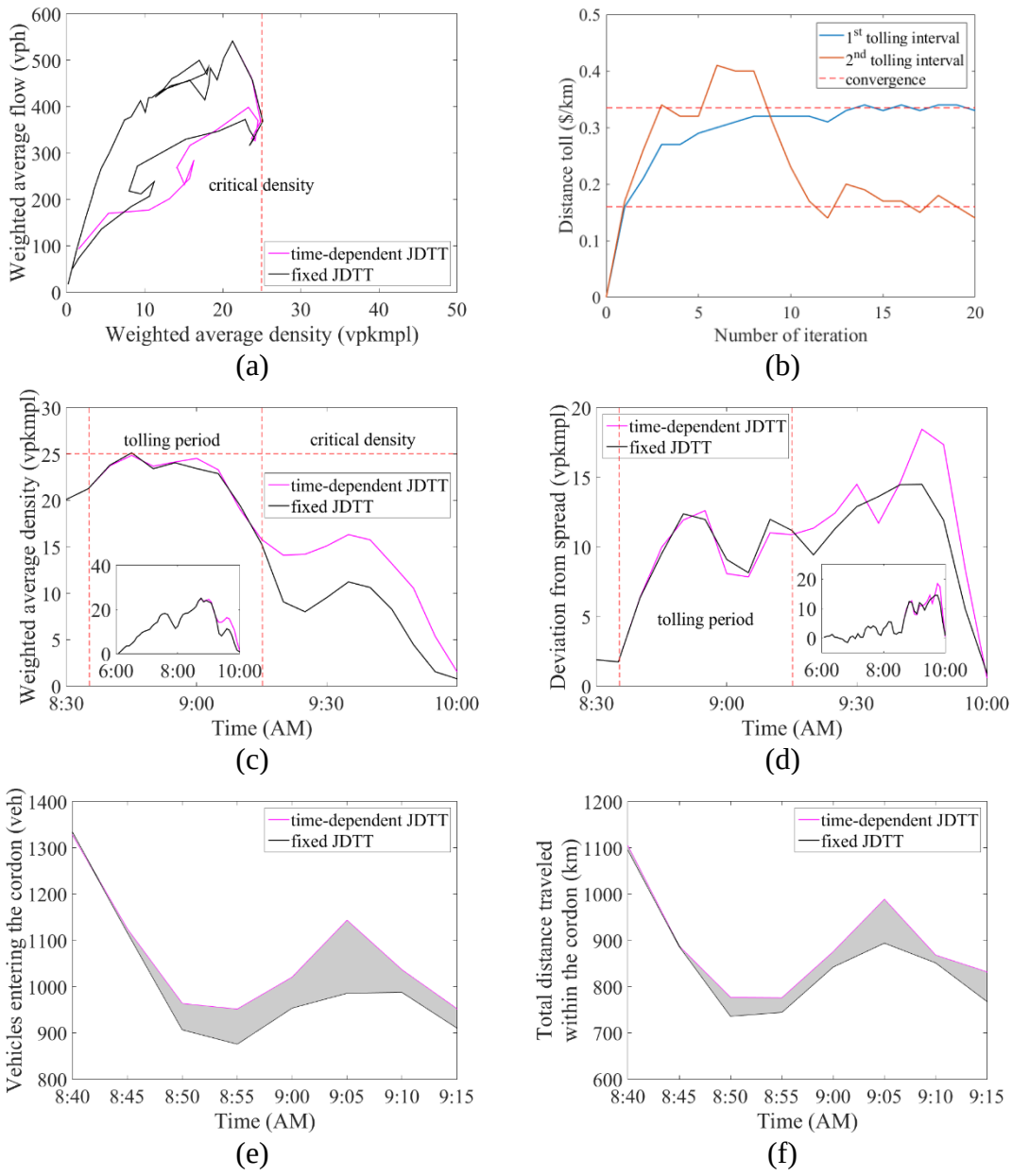


Figure 4.7 Simulation results of the PZ under the optimal time-dependent JDTT: (a) simulated NFDs, (b) convergence of the distance toll rate, (c) density time series, (d) time series of the deviation from spread; (e) time series of the number of vehicles entering the PZ, and (f) time series of the total distance traveled

4.4. Joint Distance and Delay Toll (JDDT)

The time toll component of the JDTT tends to overcharge travelers as a longer link typically requires more travel time despite being uncongested. Hence the JDDT which charges travelers according to their travel delays is more sensible. As shown in

Figure 4.8(b), the optimal delay toll rate is about \$9/h corresponding to the optimal distance toll rate of \$0.5/km. While effectively achieving the control objective, the JDDT reduces the uneven distribution of congestion within the PZ resulting in a less distinct hysteresis loop in the NFD. This is reflected quantitatively in Figure 4.8(g) and (h), and qualitatively in Figure 4.9. Results so far suggest that the JDTT and the JDDT perform equally well in controlling the network.

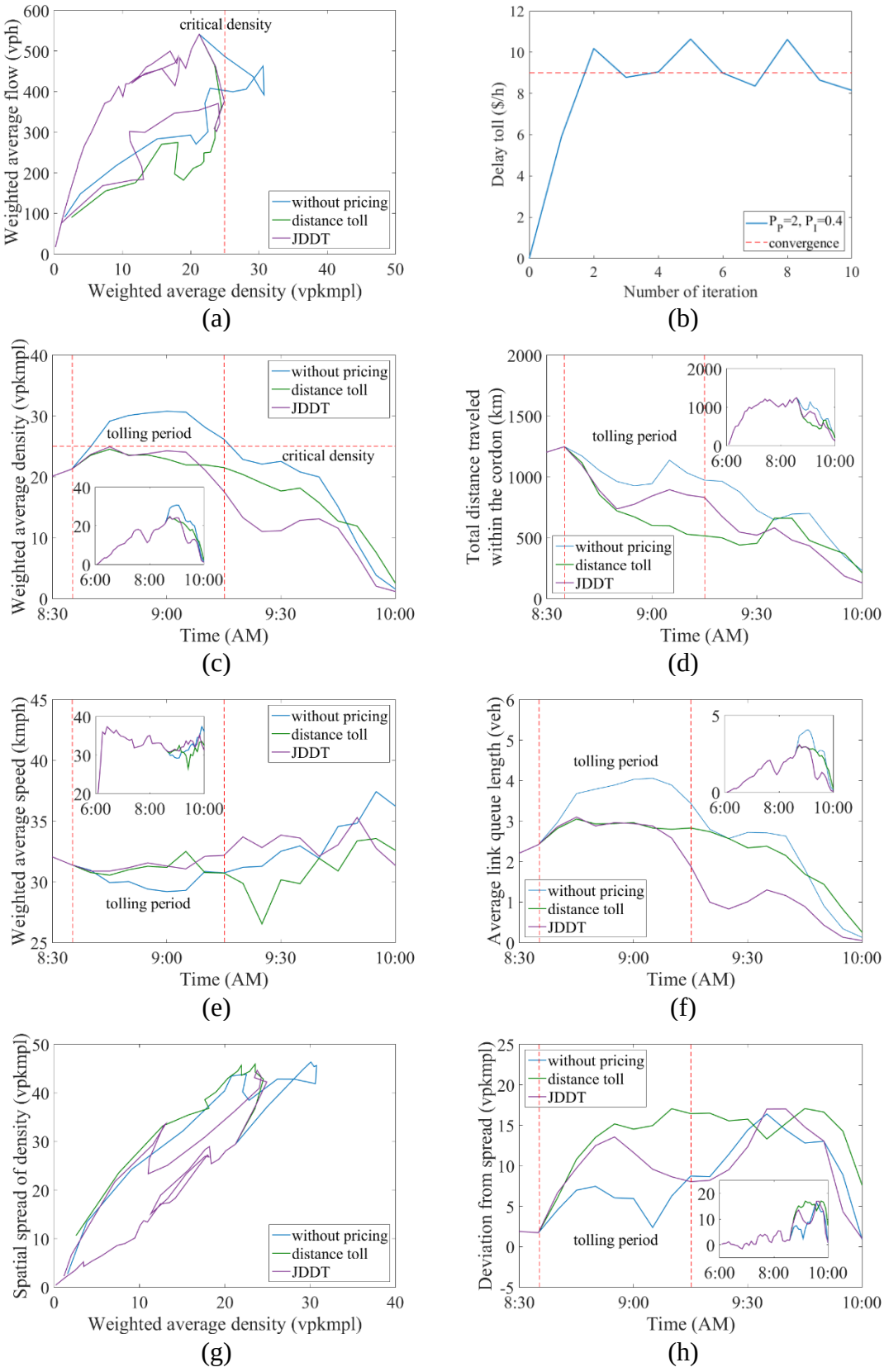


Figure 4.8 Simulation results of the PZ under the optimal JDDT scenario: (a) simulated NFDs, (b) convergence of the delay toll rate, (c) density time series, (d) time series of the total distance traveled, (e) speed time series, (f) queue time series, (g) spread-accumulation relationships, and (h) time series of the deviation from spread

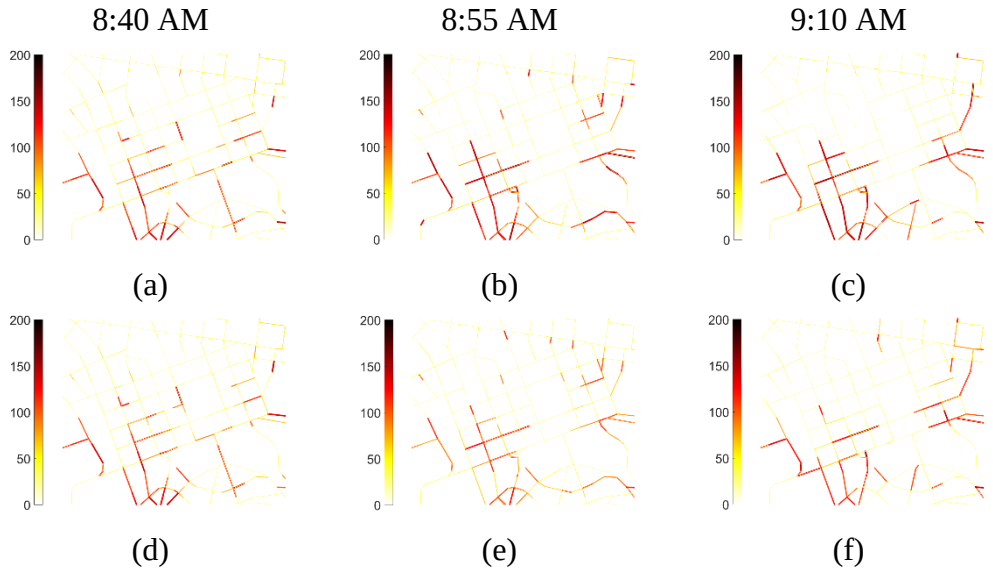


Figure 4.9 Comparing the spatiotemporal evolution of link densities within the PZ during the tolling period between (a-c) the distance only toll and (d-f) the JDDT

4.4.1. Sensitivity Analysis on the Weight Coefficient

We perform a sensitivity analysis on ω_2 to examine its effect on the pricing control results. Three different values of ω_2 are tested, i.e. $\omega_2 = 0.25, 0.5, 0.75$. A larger value implies a more dominating role played by the distance toll component. Since the convergence of the distance toll component remains unchanged regardless of the value of ω_2, we only present the convergence of the delay toll component in Figure 4.10(b). Figure 4.10(a), (c), and (d) show that the network performance does not vary significantly as ω_2 changes, and hence the pricing control results are not sensitive to different values of ω_2.

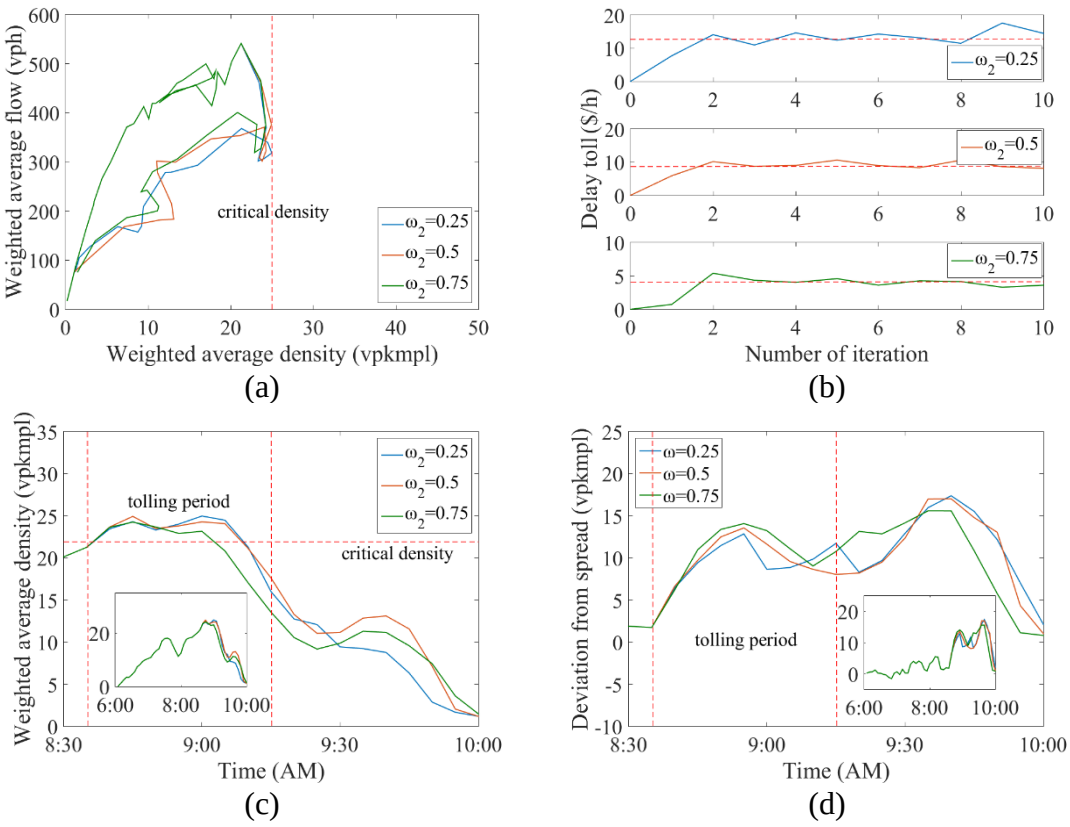


Figure 4.10 Sensitivity analysis on ω_2: (a) simulated NFDs, (b) convergence of the delay toll rate, (c) density time series, and (d) time series of the deviation from spread

4.5. Performance Comparison

Since the JDTT and the JDDT are optimized by the simultaneous and the sequential approaches, respectively, we further apply the sequential approach to the JDTT before performing the comparison. Figure 4.11 shows that the network performance is similar under the three tolling scenarios. Both the JDTT and the JDDT reduce the uneven distribution of congestion within the PZ while achieving the control objective. The reason why the difference is not significant may be because very few links in the PZ have an extra-long length. In a network where the link length varies considerably, the difference may be more remarkable.

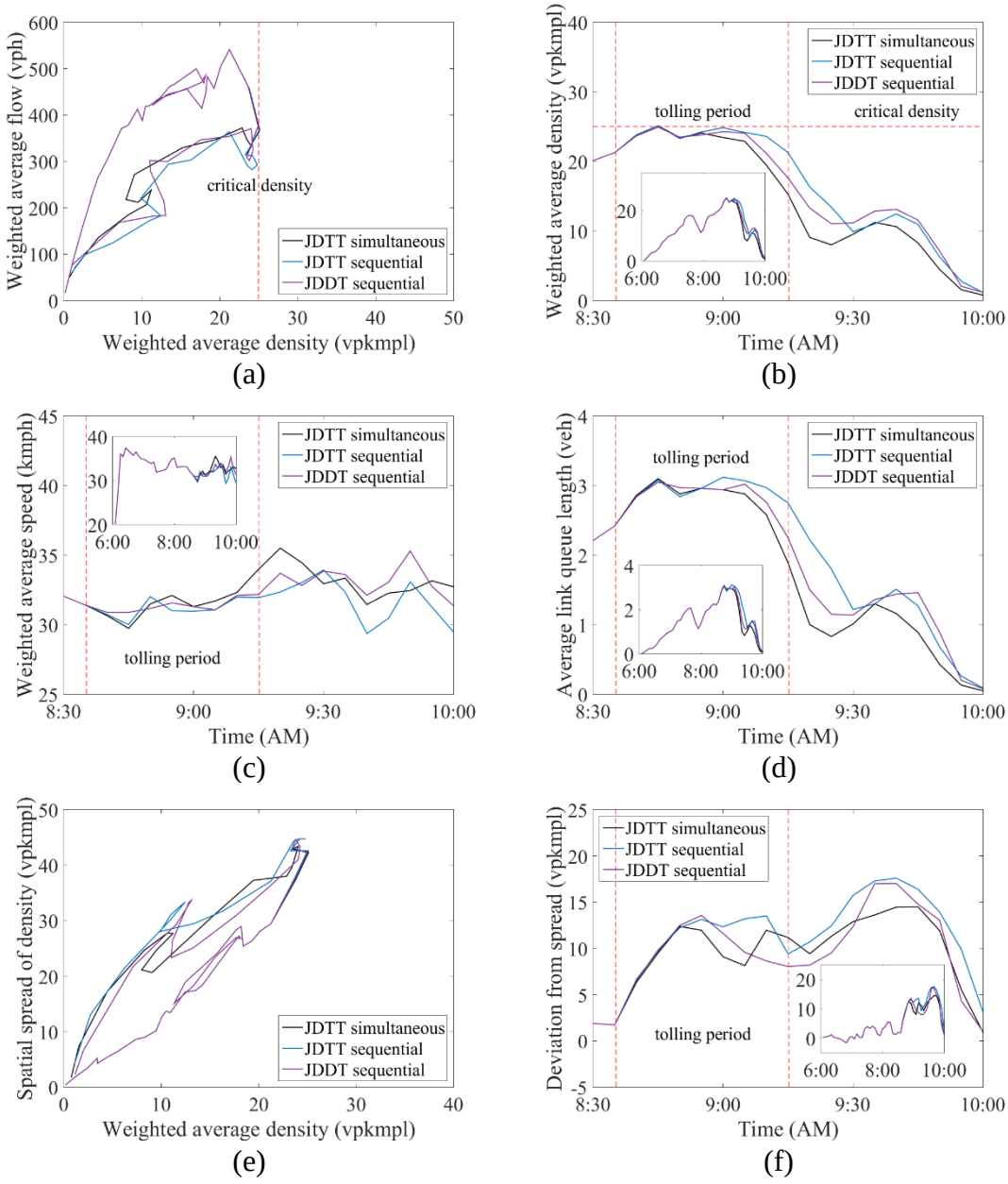


Figure 4.11 Comparing the JDTT and the JDDT: (a) simulated NFDs, (b-d) density, speed, and queue time series, (e) spread-accumulation relationships, and (f) time series of the deviation from spread

Table 4.2 shows a few selected network performance measures under different tolling scenarios. When considering the entire network, different tolls result in similar average travel times and speeds which are almost the same as those under the non-tolling scenario. The reason why the overall network performance does not change much is because the PZ only covers a relatively small area of the entire network, and hence the

effects of pricing are not significant by referring to the performance measures of the entire network. With a larger PZ, these effects shall become more remarkable. Nevertheless, with the current pricing set-up, we achieve a total travel time saving of more than 1,000 hours during the 4-h AM peak period.

When focusing on the PZ, the difference in the average travel time becomes obvious. Since the average travel time in the PZ ranges between 3 and 4 minutes, an 11-14% average travel time saving is achieved for all the tolls excluding the distance only toll. The distance only toll only improves the average travel time in the PZ by 6%, which is equivalent to a 5-10% increase in the average travel time in the PZ compared with the other tolls. This observation is consistent with the fact that the average speeds in the PZ under all the other tolling scenarios are improved by 5-14% than under the non-tolling scenario, whereas the distance only toll reduces the average speed in the PZ. Also, as expected, the distance only toll generates the lowest average distance traveled in the PZ.

Table 4.2 Selected network performance measures under different tolling scenarios

Network performance measures		Non-tolling	Tolling			
			Cordon	Distance	Time	Delay
Simulated vehicles (veh)	Entire network	348,789	348,211	346,671	348,455	347,524
	PZ	53,749	53,096	52,673	53,145	53,232
Total travel time (h)	Entire network	94,295	94,290	92,383	93,370	93,829
	PZ	3,763	3,189	3,464	3,200	3,243
Total distance traveled (km)	Entire network	2,169,537	2,182,403	2,146,487	2,160,127	2,162,866
	PZ	42,909	41,382	38,814	40,240	40,691
Average distance traveled (km/veh)	Entire network	6.22	6.27	6.19	6.20	6.22
	PZ	0.80	0.78	0.74	0.76	0.76
Average travel time (min/veh)	Entire network	16.22	16.25	15.99	16.08	16.20
	PZ	4.20	3.60	3.95	3.61	3.66
Average speed (km/h)	Entire network	23.01	23.15	23.23	23.14	23.05
	PZ	11.40	12.98	11.20	12.58	12.55

Network performance measures		Tolling		
		JDTT (simultaneous)	JDTT (sequential)	JDDT (sequential)
Simulated vehicles (veh)	Entire network	348,546	348,201	348,082
	PZ	53,121	52,633	53,549
Total travel time (h)	Entire network	95,024	94,788	93,784
	PZ	3,216	3,286	3,245
Total distance traveled (km)	Entire network	2,185,630	2,175,822	2,165,691
	PZ	39,930	39,438	40,226
Average distance traveled (km/veh)	Entire network	6.27	6.25	6.22
	PZ	0.75	0.75	0.75
Average travel time (min/veh)	Entire network	16.36	16.33	16.17
	PZ	3.63	3.75	3.64
Average speed (km/h)	Entire network	23.00	22.95	23.09
	PZ	12.42	12.00	12.40

4.6. Simulation Stochasticity

Results so far suggest that different tolls can effectively achieve the control objective. The resultant network performance, however, shows a major difference in the size of the hysteresis loop in the NFD. Since the hysteresis loop is an effect of uneven distribution of congestion, the deviation from spread is naturally a key criterion for quantitatively evaluating and comparing different tolls.

Given simulation stochasticity, we apply the feedback control approach to different tolls using ten different random seed numbers. While Figure 4.12 shows that all the tolls successfully keep the network from entering the congested regime of the NFD, Figure 4.13(a) reveals that all the other tolls outperform the distance only toll because of the less distinct hysteresis loop. This is also reflected in Figure 4.13(b) showing the distributions of the maximum deviation from spread under different tolling scenarios. As expected, the distance only toll generates the highest deviation from spread while all the other tolls perform almost similarly. While from a traffic control perspective, all the tolls are effective in reducing congestion in the PZ, from a network science perspective, the JDTT and the JDDT are more desirable than the distance only toll given their capability of reducing the uneven distribution of congestion and hence of better maintaining the network stability. While both the time only toll and the delay only toll perform equally well as the JDTT and the JDDT, they are not recommended mainly due to the safety and environmental concerns – they tend to encourage travelers to drive more aggressively and to use minor roads (May and Milne, 2000).

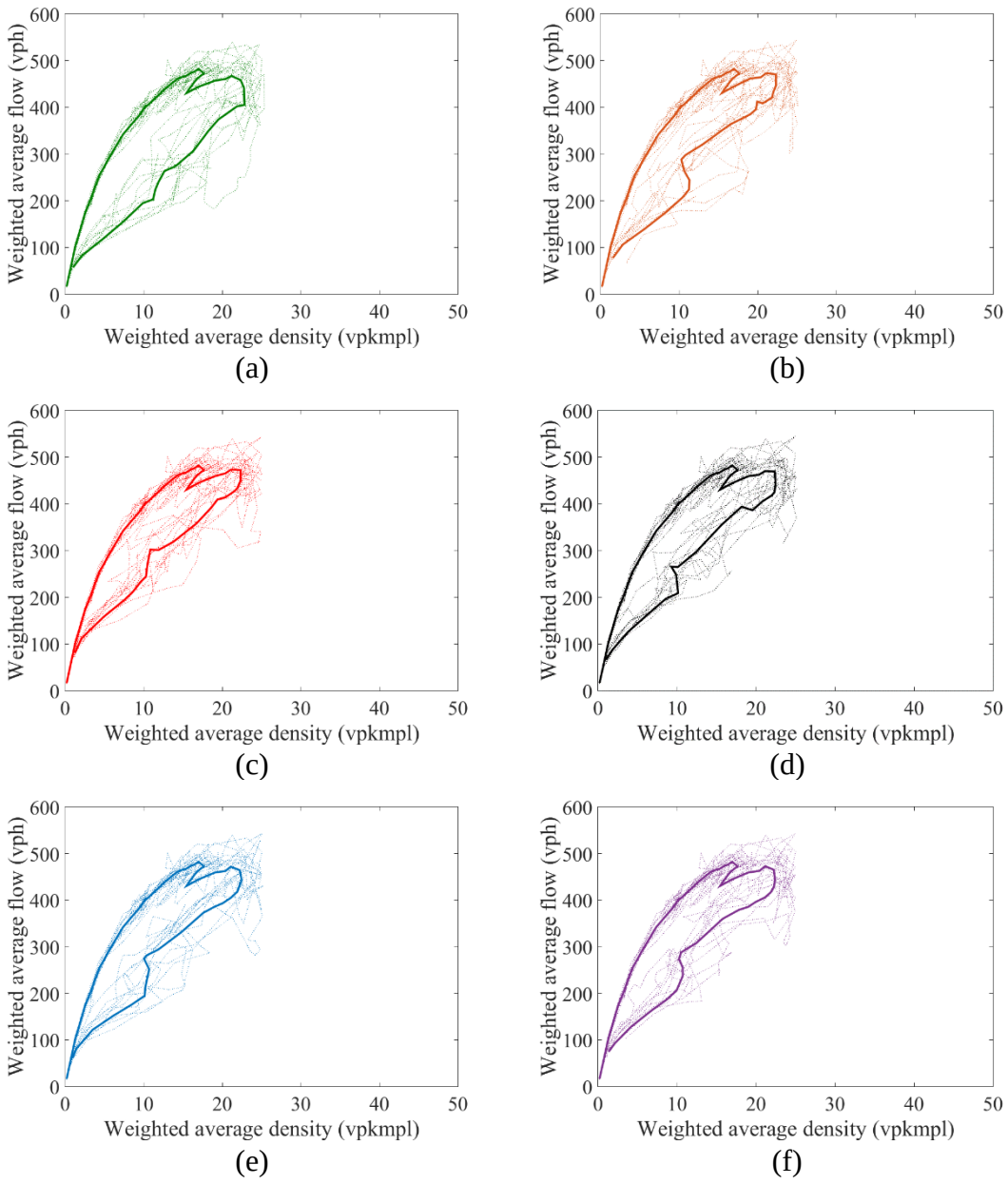
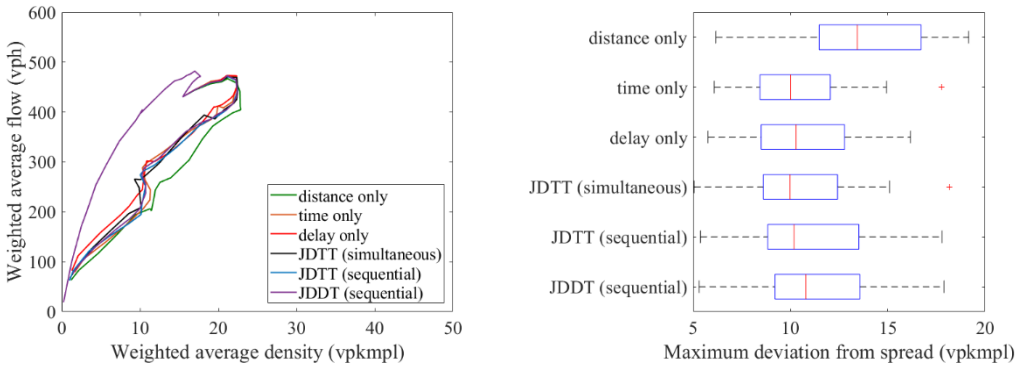


Figure 4.12 Averaged simulated NFDs with ten different random seed numbers under different tolling scenarios: (a) distance only toll, (b) time only toll, (c) delay only toll, (d) JDTT (simultaneous), (e) JDTT (sequential), and (f) JDDT (sequential)



(a) (b)

Figure 4.13 Comparing the simulation results of the PZ during the tolling period under different tolling scenarios: (a) averaged simulated NFDs, (b) distributions of the maximum deviation from spread

4.7. Global Convergence Guaranteed?

Throughout the analysis, we have repeatedly seen the global convergence of the feedback control. The question is whether it holds under all traffic scenarios. The answer is no. The feedback control is applicable with guaranteed convergence only if a prerequisite is satisfied – the periphery of the PZ should have enough capacity to accommodate the re-routed traffic. If the periphery becomes highly congested or gridlocked, it is likely that the pricing control fails.

We use the cordon toll to demonstrate the prerequisite and repeatedly apply the feedback control approach with incrementally increasing demand from 100% to 135% to manually create unreal congestion. When demand increases, the network becomes more congested and the peak-spreading phenomenon becomes more significant. Accordingly, the toll price increases and the tolling period extends. As shown in Figure 4.14, the pricing control manages to keep the network from entering the congested regime of the NFD even when demand is relatively high. However, attention should be paid to Figure 4.14(d) where a network reloading process first appears in the PZ reflected by the shape of the NFD. With a further increase in demand, the periphery of the PZ gets closer to gridlock

and hence, travelers are driven back into the PZ, although they must pay. As such, the network reloading process becomes more prominent. Under this traffic scenario, applying the feedback control largely worsens the traffic conditions outside the PZ. The highly congested periphery forces travelers to re-enter the PZ and the pricing control can no longer achieve the control objective. This is essentially a paradox where the toll price keeps rising but travelers still enter the PZ. If the prerequisite is violated, the feedback control does not necessarily result in a convergent solution.

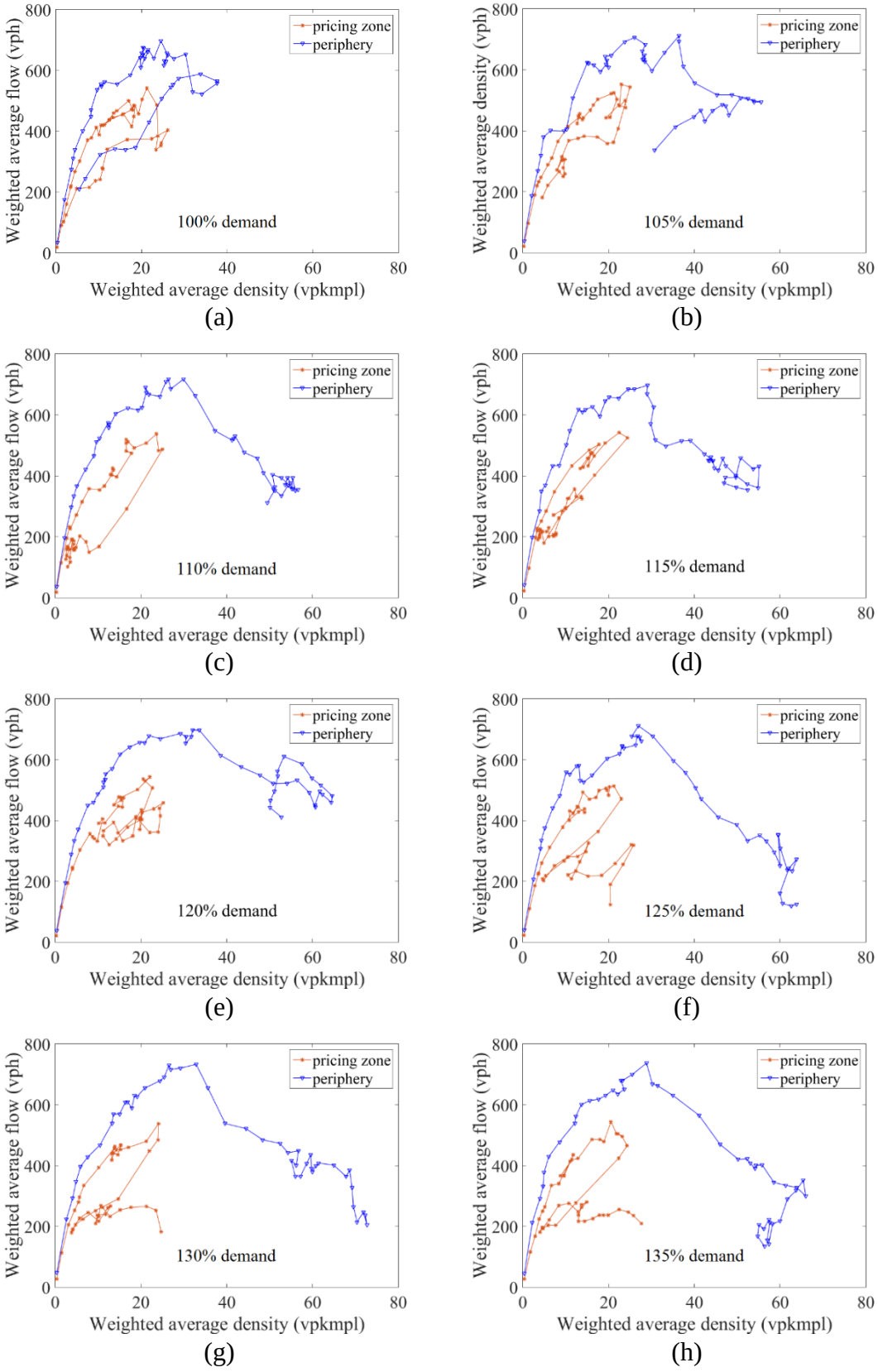


Figure 4.14 Simulated NFDs of the PZ and the periphery with incrementally increasing demand

To show that the distance only toll, the JDTT, and the JDDT do not result in degraded traffic conditions outside the PZ, we further present in Figure 4.15 the simulated NFDs of the periphery.

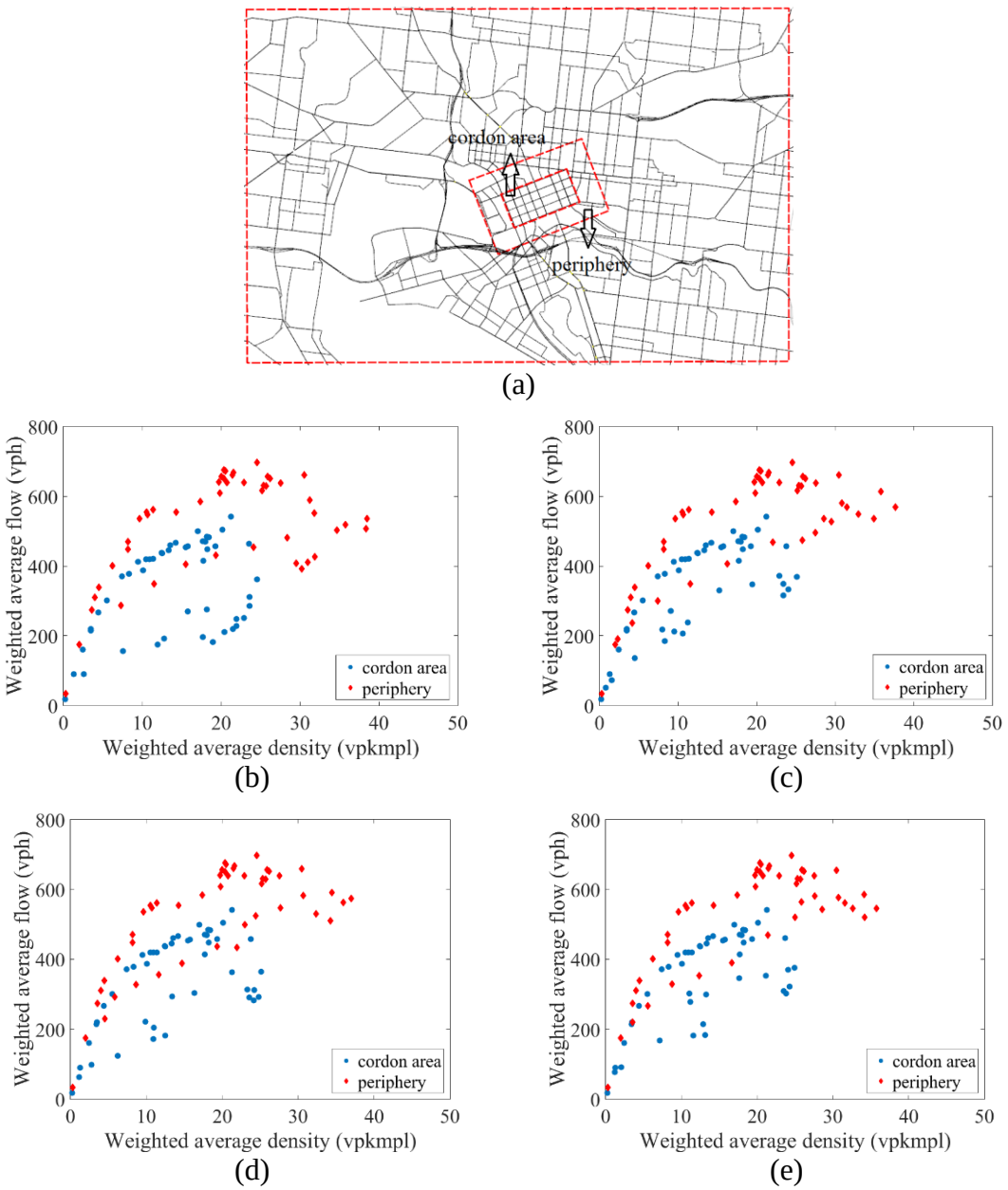


Figure 4.15 Simulated NFDs of the PZ and the periphery under different tolling scenarios: (a) layout of the periphery, (b) distance only toll, (c) JDTT (simultaneous), (d) JDTT (sequential), and (e) JDDT (sequential)

Depending on the network topology and the OD demand, the applicability of the feedback control does vary for different networks. In general, the prerequisite can be satisfied in a real-world traffic network for two reasons:

- City ring roads are often available around the urban center for detour traffic which provide enough capacity, e.g. the M1, M2, and M3 highways surrounding the city center of Melbourne.
- The OD demand needs to climb to a level that is seldom realistic for normal daily traffic, e.g. a 15% increase at least for Melbourne.

4.8. Chapter Remarks

This chapter presents detailed numerical results for the feedback control approach as an effective and efficient method for solving a simple TLP. Four key conclusions are summarized as follows:

- The distance only toll, by its nature, drives travelers into the shortest paths within the PZ, thereby increasing the heterogeneous distribution of congestion and hence the size of the hysteresis loop in the NFD.
- The JDTT and the JDDT can reduce the heterogeneity of congestion distribution while achieving the control objective.
- The feedback control approach requires that the periphery of the PZ have enough capacity to accommodate the re-routed traffic. Otherwise the pricing control may fail without reaching a globally convergent solution.
- The feedback control approach is particularly suited for solving a simple TLP featuring a set-point objective and bound constraints only.

CHAPTER 5. SURROGATE-BASED TOLL LEVEL OPTIMIZATION

This chapter provides a numerical study on the surrogate-based optimization approach for solving the TLP corresponding to Section 3.4. The pricing regime considered is the most efficient and equitable JDDT. As with CHAPTER 4, we use the simulation-based DTA model of Melbourne, Australia to evaluate the performance of different toll levels, and to find the optimum. 30% of travelers are assumed to have access to real-time information and their route choice is calculated and updated every 15 minutes.

The rest of this chapter is organized as follows. Section 5.1 presents the surrogate-based SO framework as the solution algorithm. Section 5.2 briefly discusses the base scenario for comparison purposes. Sections 5.3 and 5.4 present the numerical results for the single- and bi-objective toll optimization, respectively. A comprehensive comparison between the two is performed in Section 5.5. Section 5.6 concludes the chapter. The work of this chapter has been published:

- Gu, Z., Waller, S.T., Saberi, M., 2018. Surrogate-based toll optimization in a large-scale heterogeneously congested network. *Comput.-Aided Civ. Inf. Eng.,* 1-16.

To facilitate the presentation, the variables used in this chapter are first summarized in Table 5.1.

Table 5.1 Variables used in CHAPTER 5

Notation	Interpretation
m	Number of tolling intervals
τ_h	Toll rate for the h-th tolling interval
$\tau_{\min}/\tau_{\max}$	Lower/upper bound on the toll rate
$\bar{K}_h$	Average network density during the h-th tolling interval
K_{cr}	Critical network density
$N_{\max}$	Maximum number of iterations allowed
υ_h/ξ_h	Distance/delay toll rate
α/β	Toll pattern smoothing parameter for υ_h/ξ_h
$\bar{\delta}_h$	Average deviation from spread during the h-th tolling interval
$\delta_{\max}$	Upper bound on the deviation from spread

5.1. Surrogate-Based Simulation Optimization (SO) Framework

Consider the following complex single-objective TLP:

$$\min_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\ldots,\boldsymbol{\tau}_m} \mathrm{E}\left[\frac{1}{m}\sum_{h=1}^{m}|\bar{K}_h - K_{\text{cr}}|\right] \tag{5.1}$$

s.t.

$$|\upsilon_h - \upsilon_{h+1}| \le \alpha, \qquad h = 1,2,\ldots,m-1 \tag{5.2}$$

$$|\xi_h - \xi_{h+1}| \le \beta, \qquad h = 1,2,\ldots,m-1 \tag{5.3}$$

$$\bar{K}_h = DTA(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_m), \qquad h = 1,2,\ldots,m \tag{5.4}$$

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}_h \le \boldsymbol{\tau}_{\max}, \qquad h = 1,2,\ldots,m \tag{5.5}$$

where α and β are the toll pattern smoothing parameters for υ_h and ξ_h, respectively. The complete toll decision vector is $\boldsymbol{\tau} = [\upsilon_1, \upsilon_2, \ldots, \upsilon_m, \xi_1, \xi_2, \ldots, \xi_m]^{\mathrm{T}}$. Note that $\bar{K}_h$ is used in the objective function in place of $K_h^{\max}$ so that the optimal toll rates are less aggressive.

Problem (5.1-5.5) is similar to Problem (4.1-4.3) but with two additional constraints, namely Constraints (5.2) and (5.3), which renders the PI controller inapplicable. These are what we call the toll pattern smoothing constraints, or smoothing control constraints (Geroliminis et al., 2013), used to ensure that the optimal toll rates do not fluctuate unduly between adjacent tolling intervals, and that we obtain a smooth optimal toll pattern. It is practically infeasible to introduce a radically changing pricing system considering travelers' adaptivity and system stability. To solve Problem (5.1-5.5), the surrogate model to be built is essentially trying to learn and approximate $DTA(\cdot)$, the black-box function of the simulation model, so that optimization can be performed based on the approximated response surface.

Consider further the following complex bi-objective TLP:

$$\min_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\dots,\boldsymbol{\tau}_m} \mathrm{E}\left[\frac{1}{m}\sum_{h=1}^{m}|\overline{K}_h - K_{\mathrm{cr}}|\right] \tag{5.6}$$

$$\min_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\dots,\boldsymbol{\tau}_m} \mathrm{E}\left[\frac{1}{m}\sum_{h=1}^{m}\bar{\delta}_h\right] \tag{5.7}$$

s.t.

$$|v_h - v_{h+1}| \le \alpha, \qquad h = 1,2,\dots,m-1 \tag{5.8}$$

$$|\xi_h - \xi_{h+1}| \le \beta, \qquad h = 1,2,\dots,m-1 \tag{5.9}$$

$$\overline{K}_h = DTA(\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\dots,\boldsymbol{\tau}_m), \qquad h = 1,2,\dots,m \tag{5.10}$$

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}_h \le \boldsymbol{\tau}_{\max}, \qquad h = 1,2,\dots,m \tag{5.11}$$

where $\bar{\delta}_h$ is the average deviation from spread of the PZ during the h-th tolling interval calculated through Equation (3.4). We assume and fit a third-order polynomial function, $\gamma(K) = aK^3 + bK^2 + cK$, to the lower envelope of the spread-accumulation relationship

where a, b, and c are the coefficients to be estimated. Note that the lower envelope of the spread-accumulation relationship corresponds to the upper envelope of the NFD because for the same density, the least heterogeneity of congestion distribution contributes to the highest flow. Note also that the fitted $\gamma(K)$ here only serves as a mathematical approximation and hence does not necessarily represent the best functional form. Compared with Problem (5.1-5.5), Problem (5.6-5.11) considers an additional objective to minimize the heterogeneity of congestion distribution in the PZ for the m tolling intervals, thereby achieving further network productivity. This, to some extent, represents an approach when dealing with large-scale heterogeneous networks (Simoni et al., 2015), as an alternative to network partitioning (Ji and Geroliminis, 2012; Saeedmanesh and Geroliminis, 2016, 2017). Such an objective was previously used to develop a hierarchical perimeter control scheme (Ramezani et al., 2015). We do emphasize that while both clustering-based network partitioning and homogeneity control are effective in reducing heterogeneity, heterogeneity per se is an inherent nature of traffic networks that cannot completely disappear (Ramezani et al., 2015).

The unique feature of Problem (5.6-5.11) is that we know a priori that both objective functions have a lower bound of zero, although being too ideal to achieve. While we can still solve Problem (5.6-5.11) as it is, we can alternatively utilize this unique feature by keeping Equation (5.6) as the single main objective, same as Problem (4.1-4.3), and reformulating Equation (5.7) as an additional constraint. The original bi-objective TLP is therefore transformed into the following single-objective equivalent:

$$\min_{\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\ldots,\boldsymbol{\tau}_m} \mathrm{E}\left[\frac{1}{m}\sum_{h=1}^{m}|\overline{K}_h - K_{\mathrm{cr}}|\right] \tag{5.12}$$

s.t.

$$\mathrm{E}\left[\frac{1}{m}\sum_{h=1}^{m}\bar{\delta}_h\right] \le \delta_{\max} \tag{5.13}$$

$$|v_h - v_{h+1}| \le \alpha, \qquad h = 1,2,\dots,m-1 \tag{5.14}$$

$$|\xi_h - \xi_{h+1}| \le \beta, \qquad h = 1,2,\dots,m-1 \tag{5.15}$$

$$\bar{K}_h = DTA(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \dots, \boldsymbol{\tau}_m), \qquad h = 1,2,\dots,m \tag{5.16}$$

$$\boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}_h \le \boldsymbol{\tau}_{\max}, \qquad h = 1,2,\dots,m \tag{5.17}$$

where $\delta_{\max}$ is a constraint limit to ensure that the heterogeneity of congestion distribution is below a certain threshold. While we can alternatively keep Equation (5.7) as the objective and reformulate Equation (5.6) as the constraint, Problem (5.12-5.17) is stated in a more consistent fashion with Problem (5.1-5.5) and hence pursued. By transforming Problem (5.6-5.11) into Problem (5.12-5.17), we are able to further demonstrate the capability of surrogate-based optimization in dealing with complex constraints.

The surrogate-based SO framework is illustrated in Figure 5.1. To construct the starting surrogate model, a few initial sample points need to be generated through space-filling DOE, for each of which a network simulation is performed to evaluate the objective function. The constructed surrogate model is further subject to adding infill sample points via EI sampling until model validation is passed. For most practical applications with strict computational considerations, there is a maximum number of iterations allowed which is usually reached first before a good convergence is achieved (Amaran et al., 2016).

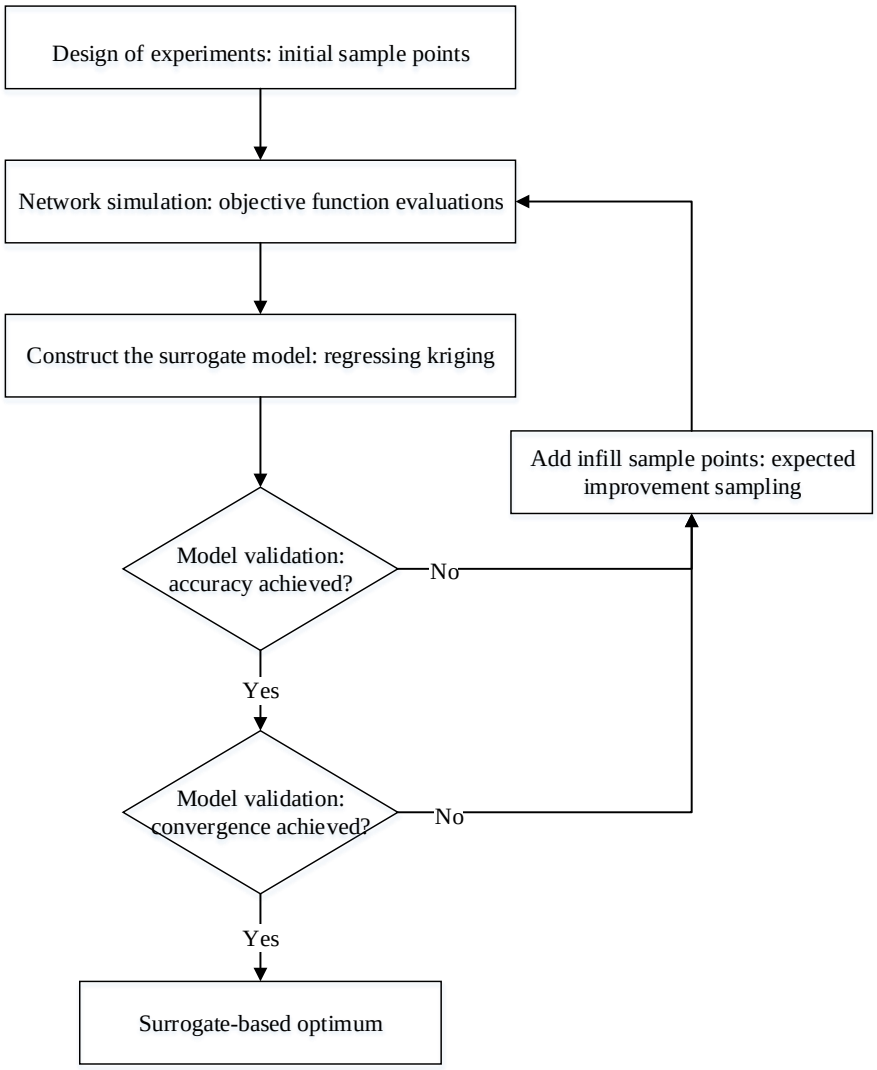


Figure 5.1 Flowchart representation of the surrogate-based SO framework

5.2. Base Scenario

We run the simulation without pricing and show the density time series and NFDs of the PZ in Figure 5.2(a) and (b), respectively. Results suggest that we set $K_{\mathrm{cr}} = 25$ vpkmpl which leads to a 2-h tolling period between 8 and 10 AM. To demonstrate the capability of surrogate-based optimization in dealing with high-dimensional problems, we use a 15-min duration and partition the entire tolling period into 8 small tolling intervals. Hence a total of 16 toll decision variables are to be optimized. Accordingly, in the maximin LHS plan, the total number of the initial sample points is 37. When applying

surrogate-based optimization, we allow a maximum of 100 iterations, i.e., the total number of sample points is 100 with 63 infill sample points. Without loss of generality, $\boldsymbol{\tau}_{\min}$ and $\boldsymbol{\tau}_{\max}$ are set at $[0,0,\dots,0,0,0,\dots,0]^{\mathrm{T}}$ and $[1,1,\dots,1,15,15,\dots,15]^{\mathrm{T}}$, respectively, and α and β are set at $\frac{1}{3}(1-0) \approx 0.33$ and $\frac{1}{3}(15-0) = 5$, respectively. Figure 5.2(c) shows the fitted $\gamma(K) = -0.0002032K^3 + 0.004432K^2 + 1.587K$.

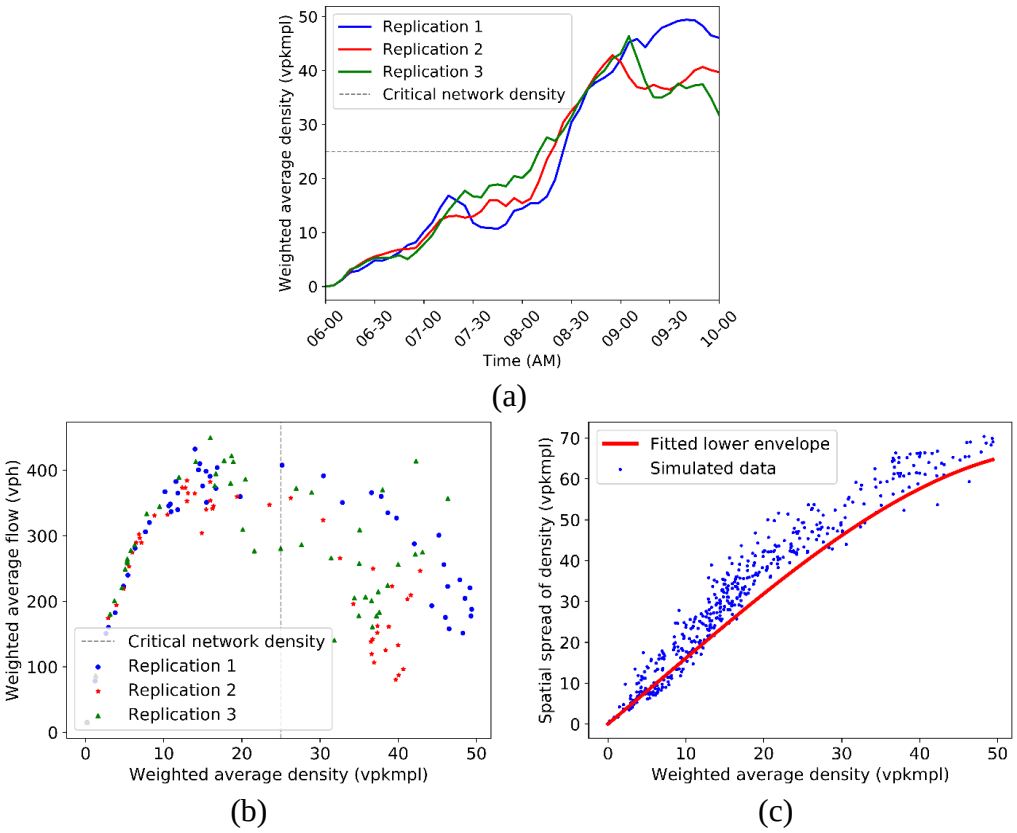
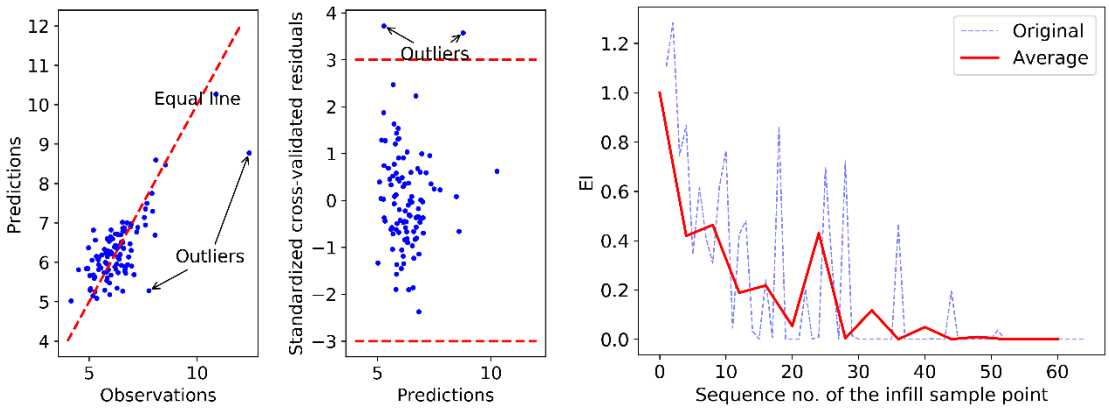


Figure 5.2 Simulation results of the PZ under the non-tolling scenario: (a) density time series, (b) simulated NFDs, and (c) spread-accumulation relationships

5.3. Solving the Single-Objective Optimization

Figure 5.3(a) validates the accuracy of the constructed surrogate model with 100 sample points. The model accuracy is sufficiently achieved with 98 standardized cross-validated residuals lying within $[-3,3]$. One outlier corresponds to the non-tolling

scenario with $\boldsymbol{\tau} = \boldsymbol{\tau}_{\min} = [0,0,\ldots,0,0,0,\ldots,0]^{\mathrm{T}}$. Since the non-tolling network produces the highest objective function value, the surrogate model makes little effort exploring the region surrounding the non-tolling sample point where the prediction becomes poor, as expected. Figure 5.3(b) illustrates the history of the EI metric. Although, due to the heuristic nature of the method, intermittent peaks representing possible significant improvements in the objective function value are observed, the overall trend of the change as represented by the average curve (averaged every four consecutive points) displays a relatively smooth convergence towards zero. This implies that, at the end of the optimization, the surrogate model is unable to locate a new solution that significantly improves the current best solution and hence, we can terminate the algorithm with confidence.



(a) (b)

Figure 5.3 Solving the single-objective TLP: (a) validating the accuracy of the constructed surrogate model, and (b) convergence of the EI metric

The solution to the single-objective TLP is shown in Figure 5.4(a). The changes in the distance and delay toll rates between adjacent tolling intervals are clearly bounded by the toll pattern smoothing constraints, respectively. Figure 5.4(b) shows the simulated averaged NFD of the PZ after applying the optimal toll rates. As expected, the congested regime of the NFD that appears and remains until the end of the simulation under the non-tolling scenario no longer exists and is substituted by a combination of a (near-)capacity

regime and a clockwise hysteresis loop. This finding is consistent with our previous finding in CHAPTER 4.

An interesting observation out of the comparison is that, compared with the non-tolling NFD, the tolling NFD undergoes a capacity drop immediately after the implementation of pricing, which, in part, contributes to the hysteresis loop in the NFD. This capacity drop results from the reduced inflow or demand to the PZ due to the presence of pricing. An extreme and obviously unrealistic scenario is that we implement an exceptionally high toll price whereby no one would enter the PZ. With such demand dropping sharply to zero, the hysteresis loop in the NFD is amplified most significantly (Mahmassani et al., 2013). A complete elimination of the capacity drop is too ideal and perhaps only possible with an extremely smooth toll pattern starting from zero, i.e. a very slow-varying toll. Figure 5.4(c-e) show, respectively, the density, speed, and queue time series of the PZ under the optimal tolling scenario in comparison with those under the non-tolling scenario. It is evident and consistent across different replications that pricing has brought significant performance improvement to the PZ represented by the area in between the two curves.

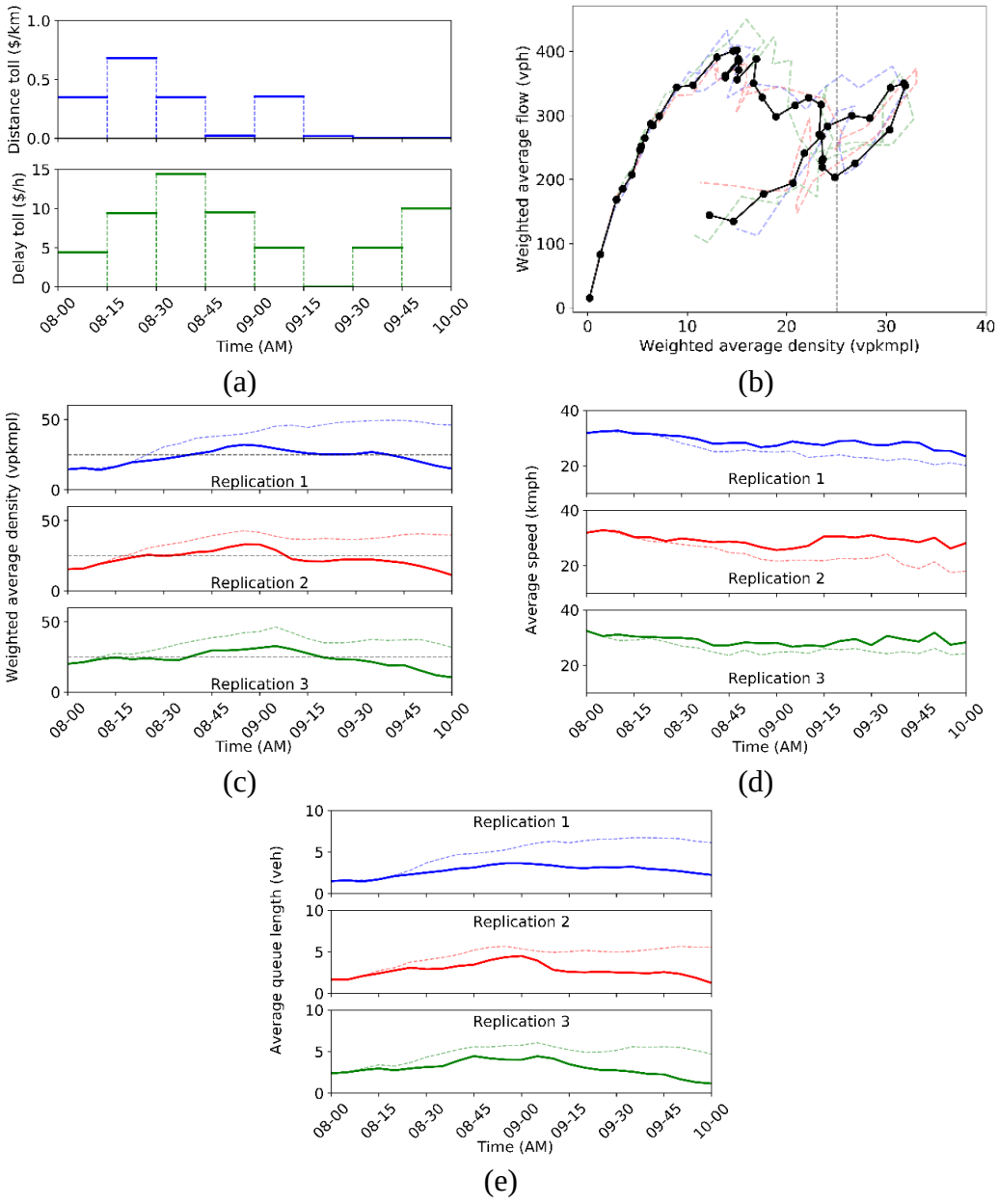


Figure 5.4 (a) Solution to the single-objective TLP and its simulation results of the PZ in comparison with those under the non-tolling scenario: (b) averaged NFD, (c-e) density, speed, and queue time series. The solid lines represent the after-pricing scenario while the dashed lines represent the before-pricing scenario

5.3.1. Sensitivity Analysis on the Toll Pattern Smoothing Parameters

To investigate the effect of toll pattern smoothing parameters, α and β, on the pricing control results, we perform a sensitivity analysis with two additional pairs of

parameters: (i) $\alpha = \frac{1}{5}(1-0) = 0.2, \beta = \frac{1}{5}(15-0) = 3$, and (ii) $\alpha = \frac{1}{2}(1-0) = 0.5, \beta = \frac{1}{2}(15-0) = 7.5$.

Mathematically speaking, a larger pair of α and β imposes less constraint on the optimization and hence would achieve a lower optimal objective function value, and vice versa. This is indeed confirmed by the optimization results. The optimal objective function value is 4.3887 for $\alpha = 0.2, \beta = 3$, 4.1614 for $\alpha = 0.33, \beta = 5$, and 4.0057 for $\alpha = 0.5, \beta = 7.5$. With a larger pair of α and β, the optimal toll pattern shown in Figure 5.5 (c) is, as expected, less smooth than that in Figure 5.5(a). Accordingly, the NFD shown in Figure 5.5(d) exhibits more chaotic behavior compared with that in Figure 5.5(b) probably due to the radical changes in the toll rates.

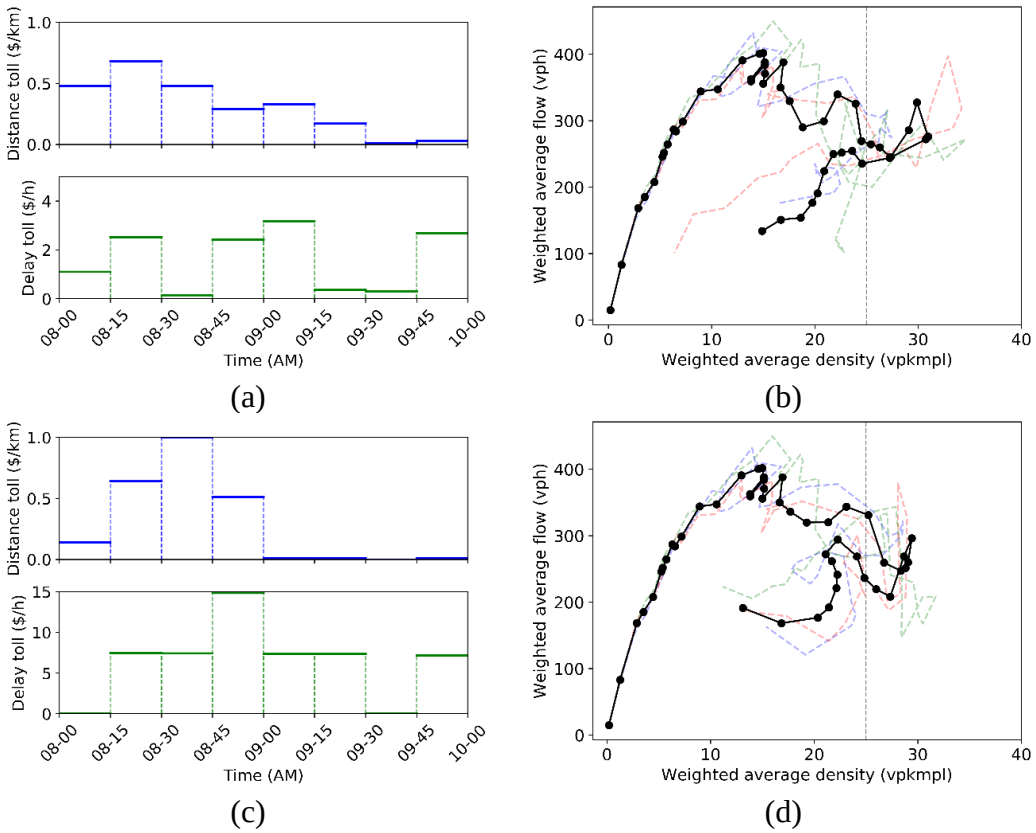
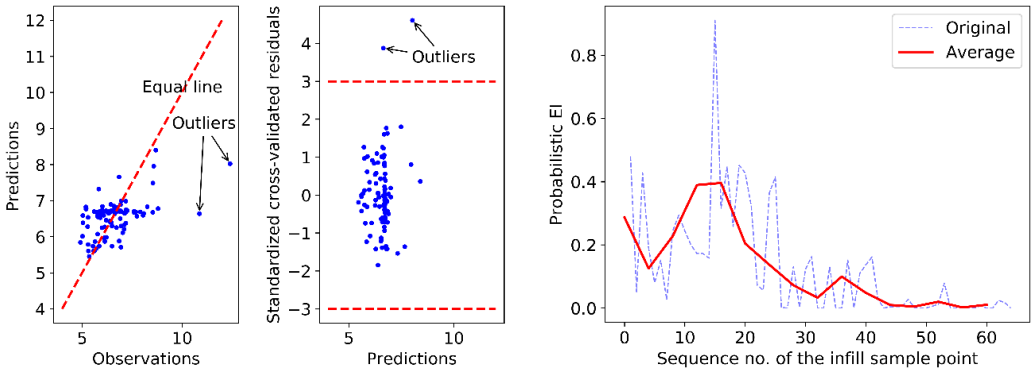


Figure 5.5 Solution to the single-objective TLP and its simulated averaged NFD of the PZ with (a) and (b) $\alpha = 0.2, \beta = 3$, and (c) and (d) $\alpha = 0.5, \beta = 7.5$

5.4. Solving the Bi-Objective Optimization

When solving the bi-objective TLP, we set $\delta_{\max}$ at 8 vpkmpl based on previous single-objective optimization results. Figure 5.6(a) validates the accuracy of the constructed surrogate model with 100 sample points. As with Figure 5.3(a), there are 98 well-predicted sample points plus two outliers. One of the outliers still corresponds to the non-tolling scenario with $\boldsymbol{\tau} = \boldsymbol{\tau}_{\min} = [0,0,\dots,0,0,0,\dots,0]^{\mathrm{T}}$, while the other outlier corresponds to the "full" tolling scenario with $\boldsymbol{\tau} = \boldsymbol{\tau}_{\max} = [1,1,\dots,1,15,15,\dots,15]^{\mathrm{T}}$. The reason is the same. $\boldsymbol{\tau}_{\min}$ undercharges drivers while $\boldsymbol{\tau}_{\max}$ overcharges drivers, both of which give rise to the highest objective function values and hence the lowest probability of finding the minimum solution in their proximity. To solve the minimization problem, the surrogate model naturally spends most of its effort exploring other regions in the design space, thereby predicting poorly for $\boldsymbol{\tau}_{\min}$ and $\boldsymbol{\tau}_{\max}$. Figure 5.6(b) shows the convergence of the probabilistic EI metric. While exhibiting a brief increasing trend at the beginning of the optimization, the pattern gradually and eventually converges to zero like Figure 5.3(b). Note that the probabilistic EI values in Figure 5.6(b) are generally smaller than those in Figure 5.3(b) because the probability of satisfying the constraint is always less than or equal to one.



(a) (b)

Figure 5.6 Solving the bi-objective TLP: (a) validating the accuracy of the constructed surrogate model, and (b) convergence of the probabilistic EI metric

Figure 5.7(a) shows the distribution of the 100 sample points based on their objective and constraint function values. Obviously, we are only interested in points lying below the constraint limit line represented by the blue filled circles. An interesting observation is that a Pareto front seems to appear suggesting a conflicting relation between the objective and the constraint. This observation, in part, supports our previous argument about the capacity drop. Specifically, while a higher toll price may decrease the objective function value, it may also increase the constraint function value by creating a more significant drop in the inflow to the PZ. A further reduced inflow equates to a more notable capacity drop and hence, a larger hysteresis loop in the NFD or a higher level of deviation from spread. Under the non-tolling scenario, the deviation from spread is the lowest as the PZ goes all the way to almost gridlock with no network recovery, see Figure 5.2(a). The solution to the bi-objective TLP is shown in Figure 5.7(b) which corresponds to the corner point at the intersection of the Pareto front and the constraint limit line in Figure 5.7(a). The solution to the single-objective TLP is also shown by the green cross which has a lower objective function value but a higher constraint function value, as expected. Figure 5.7(c-f) show, respectively, the simulated averaged NFD, density, speed, and queue time series of the PZ under the optimal tolling scenario. The tolling NFD

successfully maintains itself within the free-flow and at or near the capacity regimes without entering the congested branch of the non-tolling NFD. Traffic conditions in the PZ experience significant improvement with much lower densities and queues, and larger speeds.

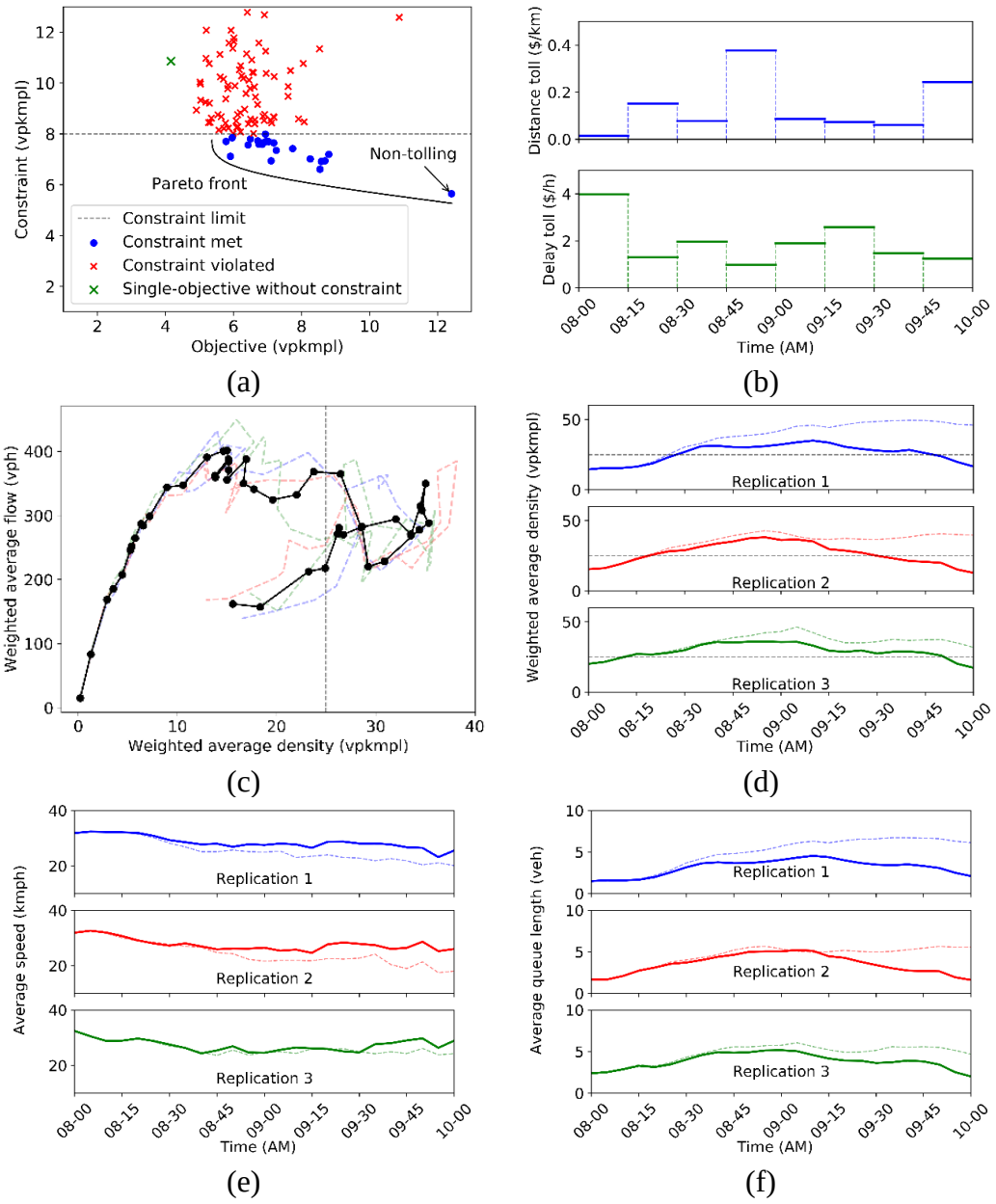


Figure 5.7 (a) Distribution of the 100 sample points based on their objective and constraint function values, (b) solution to the bi-objective TLP and its simulation results of the PZ in comparison with those under the non-tolling scenario: (c) averaged NFD, (d-f) density, speed, and queue time series. The solid lines represent the after-pricing scenario while the dashed lines represent the before-pricing scenario

5.5. Performance Comparison

As shown in Figure 5.8(a), the NFD from the bi-objective optimization shifts more to the right because the heterogeneity constraint results in a lower toll price and hence a higher objective function value. Nevertheless, due to a lower constraint function value, higher flows are achieved during network loading which equates to a reduced capacity drop. During the transition period, although the NFD from the bi-objective optimization works at higher densities, it produces similar or even slightly higher flows. Assuming a trapezoidal network exit function, there is a range of densities centering around the critical network density within which the flow can maintain at or near capacity (Daganzo, 2007; Mahmassani et al., 2013). Another observation is that the NFD from the single-objective optimization exhibits a more significant local oscillatory loop. While the density remains almost constant, the flow undergoes a near-vertical jump along with a more heterogeneous distribution of congestion, see Figure 5.8(a) and (b). This was also reported in Simoni et al. (2015). During network recovery, both NFDs exhibit a sizable hysteresis loop amplified by the very low demand entering the PZ at the end of the simulation. Figure 5.8(b) shows that, although the bi-objective optimization leads to higher densities, it produces slightly and consistently higher flows throughout the tolling period due to a lower level of the deviation from spread. Figure 5.8(c) and (d) show that compared with the non-tolling scenario, the two optimal TLP solutions reduce the average travel time in the PZ by an average of 29.5% and 21.6%, respectively. The bi-objective optimization achieves less travel time improvement in the PZ because it allows the density to evolve further beyond the critical network density. While one may immediately question the 7.9% loss of travel time improvement in the PZ, a comparison between the average travel time in

the entire network certainly provides the answer. Compared with the non-tolling scenario, the bi-objective optimization reduces the average travel time in the entire network by an average of 2.5%, which is 1.1% higher than that by the single-objective optimization. Therefore, the bi-objective optimization essentially manages to convert the 7.9% loss of travel time improvement in the PZ into the 1.1% gain of travel time improvement in the entire network. While producing less network-wide travel time improvement in the first two replications, the single-objective optimization slightly increases the average network travel time in the third replication probably due to overcharging the PZ and shifting congestion to the peripheral network. Two questions remain to be answered: (i) why is the travel time improvement in the entire network much lower than that in the PZ? And (ii) is it worthwhile to achieve the 1.1% gain of travel time improvement in the entire network at the cost of the 7.9% loss of travel time improvement in the PZ?

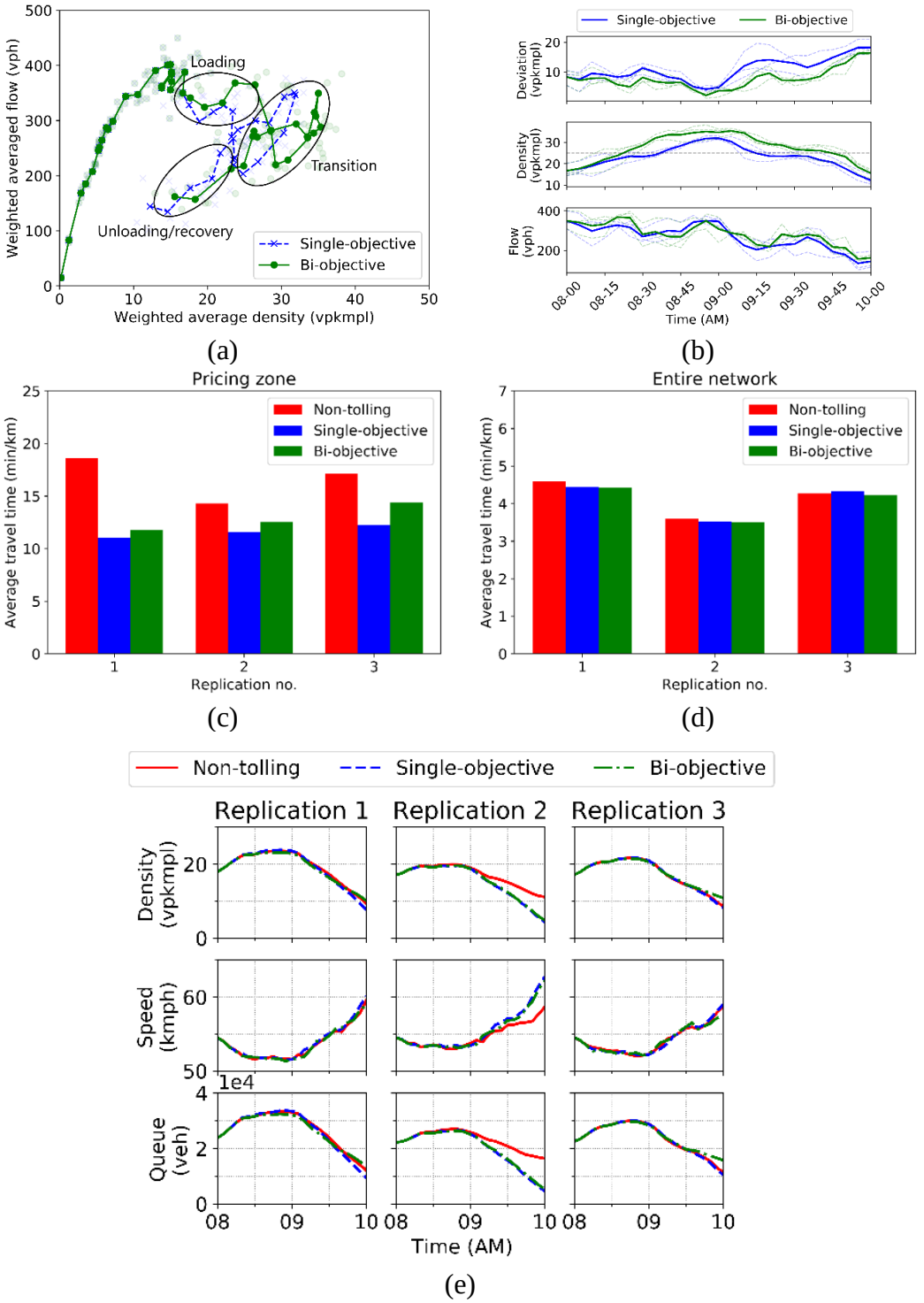


Figure 5.8 Comparing the simulation results of the two optimal TLP solutions: (a) averaged simulated NFDs of the PZ, (b) deviation, density, and flow time series of PZ, (c) average travel time in the PZ, (d) average travel time in the entire network, and (e) density, speed, and queue time series of the entire network

The answer to the first question is quite straightforward which has already been provided in CHAPTER 4. The scale effect is a major reason given that the PZ only covers

a relatively small area of the entire network. It is therefore no surprise that the performance of the entire network changes very little, see Figure 5.8(e), when pricing a relatively small sub-network. The performance of the entire network may even reduce, e.g. in the third replication, due to the redistribution of detour vehicles around the PZ which is highly dependent on the network configuration and the structure and magnitude of the demand, and hence case-specific. Since the surrogate-based SO framework represents an uncoordinated approach to pricing system design as our focus is explicitly and entirely on optimizing the performance of the PZ, we need to check and ensure in an unsystematic manner that the optimal solution does not create unintended evident deterioration in the performance of the entire network.

The answer to the second question is a quick yes, at least from the authors' perspective. While acknowledging the fact that 1.1% is much lower and hence less seemingly appealing than 7.9%, we emphasize that the average travel time is normalized against the total distance traveled. Given that the total distance traveled in the entire network is over 60 times of that in the PZ, the total travel time saving in the entire network offered by the 1.1% is accordingly much higher than that in the PZ offered by the 7.9%. Indeed, we achieve, on average, a further network-wide total travel time saving of almost 700 hours during the 4-h AM peak period. From a global perspective, it is certainly worthwhile to achieve the 1.1% gain of travel time improvement in the entire network at the cost of the 7.9% loss of travel time improvement in the PZ.

5.6. Chapter Remarks

This chapter presents detailed numerical results for the surrogate-based optimization approach as a computationally efficient method for solving a complex TLP. Two key conclusions are summarized as follows:

- Considering and reducing the heterogeneity of congestion distribution as part of the TLP helps achieve a higher network flow.
- Surrogate-based optimization is a more general approach that is particularly suited for solving a complex TLP, i.e. a TLP with either a complex objective or complex constraints.

CHAPTER 6. COMPARING DIFFERENT SIMULATION OPTIMIZATION (SO) METHODS

This chapter provides an in-depth investigation into the performance of different SO methods on two benchmark TLPs and compares their solution quality and computational efficiency as key application considerations. A recent comprehensive overview of different SO methods can be found in Amaran et al. (2016). Given the problem at hand, we focus on continuous SO but not discrete SO. To the best of our knowledge, different continuous SO methods can be classified into seven broad categories: (i) random search or metaheuristics, (ii) RSM, (iii) stochastic approximation (SA), (iv) direct search, (v) estimation of distribution algorithms (EDAs), (vi) Lipschitzian optimization, and (vii) feedback control. Given an expensive TLP, random search and EDAs are left out because of their demanding requirement of enormous function evaluations. Direct search as a usual local optimizer is not considered either as we emphasize global optimization that is immune to getting trapped in a bad local optimum.

We consider and compare the most representative and perhaps the best performing SO method for each of the four identified categories: (i) the PI controller method for feedback control (Section 3.3), (ii) RK for RSM (Section 3.4), (iii) SPSA for SA (Spall, 1992), and (iv) DIRECT for Lipschitzian optimization (Jones et al., 1993). To account for simulation noise of a stochastic traffic simulator commonly rendered by different random seed numbers, we can readily couple standard fixed- or "smarter" variable-number sample path optimization (Deng and Ferris, 2009) with the above methods. Do keep in mind that, as we previously touched upon, computer simulations often display what we

call numerical noise as well – the objective function evaluations tend to scatter about a smooth trend rather than lying on it (Forrester et al., 2006).

The four SO methods are briefly summarized in Table 6.1. We will later elaborate, respectively, on SPSA and DIRECT in Sections 6.1 and 6.2, while further details of the PI controller method and RK can be found in Sections 3.3 and 3.4, respectively. Compared with the other three methods, the PI controller method is highly demanding on the problem formulation – only set-point objective functions and box constraints can be considered. However, when the problem is indeed formulated in such a form, the PI controller method tends to converge much faster as we will show in Section 6.3. In Section 6.4, we will compare the performance of the other three methods on the complex TLP. In addition to the normal simulation or function evaluation cost and the decision vector adjustment or selection cost, each method has its own distinct overheads when applied of which one should be aware. DIRECT is perhaps the only exception that does not involve heavy overheads. The work of this chapter has been published and is currently under review:

- Gu, Z., Saberi, M., 2019. Continuous simulation-based optimization of expensive black-box traffic systems: A comparative review of algorithms and application to toll pricing. *Transp. Res. Part B*, under review.

Table 6.1 Summary of the four SO methods

Method	Mechanism	Capabilities		Overheads
		Objective	Constraint	
PI controller	Applying trial-and-error to gradually reduce the error from the set point	Set point	Box	Parameter tuning
RK	Approximating the simulation input-output mapping by a mathematical construct	Any	Any	Parameter estimation
SPSA	Using finite-difference approximation to enable gradient descent	Any	Any	Parameter tuning
DIRECT	Diving the parameter space into (hyper)rectangles by function evaluations	Any	Any	Almost none

To facilitate the presentation, the variables used in this chapter are first summarized in Table 6.2.

Table 6.2 Variables used in CHAPTER 6

Notation	Interpretation
k	Problem dimension
τ_i	Decision vector for the i-th iteration
$\boldsymbol{\Delta}_i$	Random perturbation vector for the i-th iteration
a, c, α, γ, A	User-specified parameters for SPSA
$\boldsymbol{c}_j$	Midpoint of the j-th hyperrectangle
d_j	Distance between $\boldsymbol{c}_j$ and the hyperrectangle vertices
$\varepsilon, \widetilde{K}$	User-specified parameters for DIRECT
$\mathcal{H}$	Potentially optimal hyperrectangles
$\mathcal{D}^{\hbar}$	Set of dimensions with the longest side length for $\hbar \in \mathcal{H}$
$\delta^{\hbar}$	One third of the longest side length for $\hbar \in \mathcal{H}$
$\boldsymbol{c}^{\hbar}$	Midpoint of $\hbar \in \mathcal{H}$
$\boldsymbol{e}_i$	i-th unit vector
m	Number of tolling intervals
K_{cr}	Critical network density
$P_{\mathrm{P}}/P_{\mathrm{I}}$	Proportional/integral gain parameter
$\boldsymbol{\epsilon}_{\mathrm{noise}}$	Error due to noise
$\boldsymbol{\epsilon}_{\mathrm{perturbation}}$	Error due to simultaneous perturbation

6.1. Simultaneous Perturbation Stochastic Approximation (SPSA)

SA is a widely used method in various engineering areas to solve a challenging optimization problem that does not have an analytical solution and/or is contaminated with noise. Usually, the solution to the optimization problem is a decision vector at which

the gradient of the objective function is zero. If information of the gradient is directly available, the problem can be solved using a gradient-based method, e.g. steepest descent. The biggest obstacle, however, is that in most practical applications especially where computer simulations are used, the gradient is like an inaccessible "black box" and one only has measurements of the objective function. In this context, SA comes into play which approximates the gradient using only objective function evaluations.

The two-sided finite-difference stochastic approximation (FDSA), e.g. Kiefer and Wolfowitz (1952), is a well-known SA method. It works fine for small dimensional problems but poorly in terms of computational efficiency for problems featuring a high dimensional decision vector. This is because the number of objective function evaluations required per iteration is twice the number of the problem dimension. In contrast, random direction stochastic approximation (RDSA) or SPSA as a special case is a highly efficient simultaneous perturbation (SP) approximation to the gradient that requires only two objective function evaluations per iteration irrespective of the problem dimension (Spall, 1992). This feature renders the method competitive in solving large-scale SO problems.

The essence of SPSA lies in how it approximates the gradient. Let us assume that, without loss of generality, the problem to be solved is formulated as a minimization problem with respect to a k-dimensional decision vector denoted by $\boldsymbol{\tau}_i = [\tau_1, \tau_2, \dots, \tau_k]^{\mathrm{T}}$ where i is the iteration counter. To approximate the gradient using SP, we generate a corresponding k-dimensional random perturbation vector denoted by $\boldsymbol{\Delta}_i$, each element of which, i.e. Δ_{il} where $l \in (1,2, \dots, k)$, is independently generated by Monte Carlo from a zero-mean probability distribution satisfying the SPSA regularity conditions (Spall, 1992). In short, the common uniform and normal distributions are not qualified whereas a simple, valid, and perhaps the most widely advocated choice is the Bernoulli distribution with 0.5

probability for ± 1. The SP approximation to the unknown true gradient, $\hat{\mathbf{g}}_i(\boldsymbol{\tau}_i)$, is therefore calculated as follows:

$$\hat{\mathbf{g}}_i(\boldsymbol{\tau}_i) = \frac{f(\boldsymbol{\tau}_i + c_i\boldsymbol{\Delta}_i) - f(\boldsymbol{\tau}_i - c_i\boldsymbol{\Delta}_i)}{2c_i} \begin{bmatrix} {\Delta_{i1}}^{-1} \\ \vdots \\ {\Delta_{ik}}^{-1} \end{bmatrix} \tag{6.1}$$

where $c_i = \frac{c}{(i+1)^\gamma}$ is usually a small number for gradient approximation with user-specified parameters c and γ. A practically effective and theoretically valid value of γ is 0.101 and c should not be set close to zero given a highly noisy objective function (Spall, 1998). Note that one can opt to average several SP approximations to the gradient per iteration when the noise level of the objective function is very high so as to increase the stability of the method in the early iterations.

With Equation (6.1), the SPSA-enabled SO framework is illustrated in Figure 6.1 together with the following detailed algorithmic steps.

Step 1. *Parameter initialization.* Set $i = 1$ and choose $\boldsymbol{\tau}_1 \in \mathbb{R}^k$ as an initial guess for the decision vector. Set $c_i = \frac{c}{(i+1)^\gamma}$ and the step size $a_i = \frac{a}{(A+i)^\alpha}$ where a, A, and α are user-specified parameters. A couple of guidelines for parameter selection (Spall, 1998) include (i) a practically effective and theoretically valid value of α is 0.602, (ii) A is usually set at 10% or less of the maximum number of expected or allowed iterations, and (iii) a is typically chosen such that $\frac{a}{(A+1)^\alpha}$ times the magnitude of the elements of $\hat{\mathbf{g}}_1(\boldsymbol{\tau}_1)$ approximately equates to the smallest desired change in the magnitude of the elements of $\boldsymbol{\tau}$ in the early iterations.

Step 2. *Random perturbation.* Generate a k-dimensional random perturbation vector, $\boldsymbol{\Delta}_i$, where each element is independently sampled via Monte Carlo from a Bernoulli distribution with 0.5 probability for ± 1.

Step 3. *Objective function evaluation.* Run the simulation for both simultaneously perturbed decision vectors and evaluate their objective function values.

Step 4. *Gradient approximation.* Calculate the SP approximation to the gradient using Equation (6.1).

Step 5. *Decision vector update.* Apply the following standard SA formulation to update the decision vector:

$$\boldsymbol{\tau}_{i+1} = \boldsymbol{\tau}_i - a_i \hat{\mathbf{g}}_i(\boldsymbol{\tau}_i) \tag{6.2}$$

Step 6. *Stop test.* Terminate the algorithm if there is little change in several successive gradient approximations or objective function evaluations, or the maximum number of iterations allowed is reached; otherwise set $i = i + 1$ and go back to Step 2.

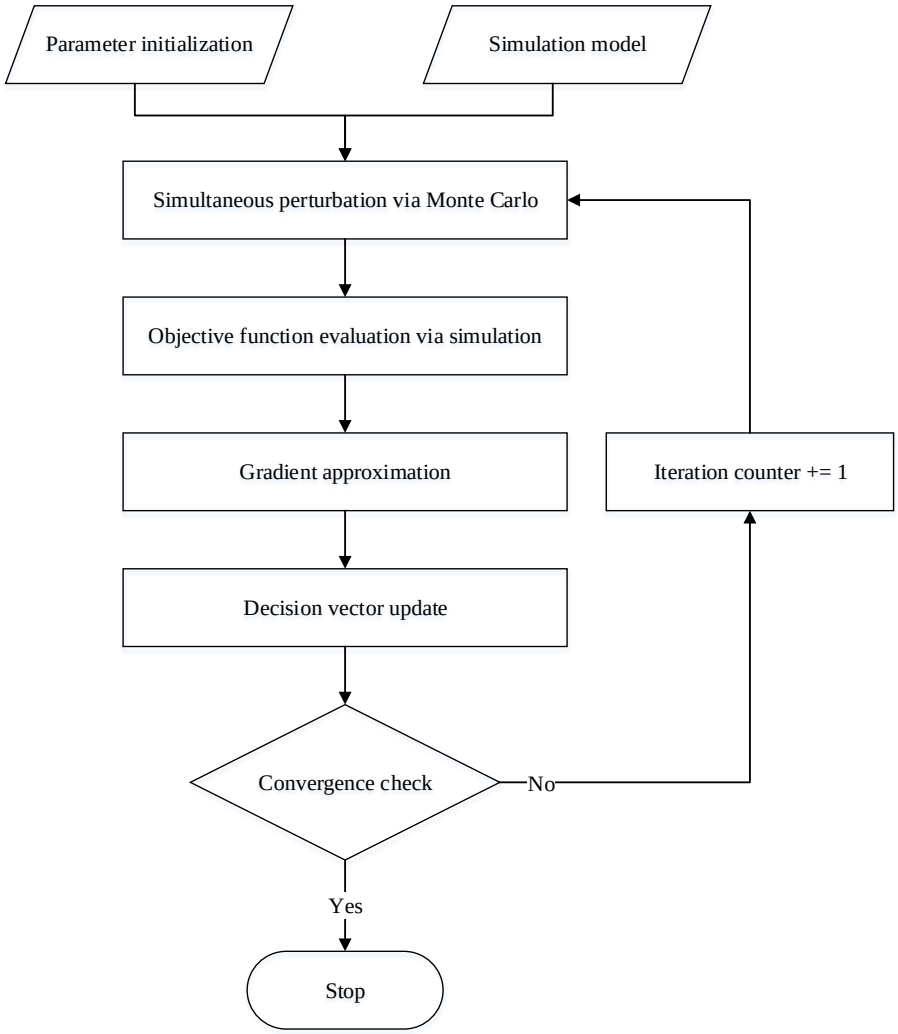


Figure 6.1 Flowchart representation of the SPSA-enabled SO framework

6.2. DIviding RECTangles (DIRECT)

Lipschitzian optimization, e.g. Shubert (1972), has always been an attractive method for finding the global optimum or, more generally, multiple global optima if more than one exists to an optimization problem. The reason behind its popularity is threefold:

- The method is deterministic without the need for multiple runs.
- Very few parameters are to be specified except for the Lipschitz constant – a bound on the rate of change of the objective function, and hence little effort is needed for parameter tuning.

- The method can generate a lower bound on the optimal objective function value that enables the adoption of more meaningful stopping criteria.

Nevertheless, Lipschitzian optimization also bears three disadvantages that prevent its further applications:

- The Lipschitz constant to be specified may not be easily calculated or simply does not exist.
- The method usually has a low speed of convergence. This is because the Lipschitz constant as an upper bound on the rate of change of the objective function is typically a large value that puts more emphasis on global exploration than on local exploitation.
- The method needs to sample and evaluate every vertex of the partitioned hyperrectangle from the original search space. Therefore, the computational complexity significantly increases for solving a high-dimensional optimization problem.

In view of the above limitations of the standard Lipschitzian optimization, Jones et al. (1993) proposed a new global optimization method termed DIRECT that eliminates the need for a user-specified Lipschitz constant. Equally contributive is that the method only samples and evaluates the midpoint of the partitioned hyperrectangle from the original search space irrespective of the problem dimension, which largely reduces the number of required objective function evaluations and achieves computational efficiency. In a nutshell, DIRECT works by iteratively partitioning the search space into multiple hyperrectangles and identify what are called potentially optimal hyperrectangles for further partitioning. As such, the method has two core components consisting of hyperrectangle partitioning and potentially optimal hyperrectangle identification.

Step 1. *Initialization.* Normalize the search space into a unit hypercube whose midpoint is denoted by $\boldsymbol{c}_1$. Run the simulation and evaluate the objective function value at $\boldsymbol{c}_1$, denoted by $f(\boldsymbol{c}_1)$. Set the current best solution $f_{\min} = f(\boldsymbol{c}_1)$ and the iteration counter $t = 1$.

Step 2. *Potentially optimal hyperrectangle identification.* Let us assume that the original search space is currently partitioned into m hyperrectangles. We use $\boldsymbol{c}_j$ to denote the midpoint of the j-th hyperrectangle and d_j to denote the distance between $\boldsymbol{c}_j$ and the hyperrectangle vertices. A hyperrectangle, j^*, is potentially optimal if the following two inequalities hold for some $\widetilde{K} > 0$ where $\varepsilon > 0$ is a small constant:

$$f(\boldsymbol{c}_{j^*}) - \widetilde{K} d_{j^*} \leq f(\boldsymbol{c}_j) - \widetilde{K} d_j, \qquad j \in (1,2,\dots,m) \tag{6.3}$$

$$f(\boldsymbol{c}_{j^*}) - \widetilde{K} d_{j^*} \leq f_{\min} - \varepsilon |f_{\min}| \tag{6.4}$$

Step 3. *Hyperrectangle partitioning.* Let $\mathcal{H}$ denote the set of potentially optimal hyperrectangles identified from Step 2. For each $\hbar \in \mathcal{H}$, denote the set of dimensions with the longest side length by $\mathcal{D}^{\hbar}$ and set $\delta^{\hbar}$ to one third of this length. Sample and evaluate the objective function values at points $\boldsymbol{c}^{\hbar} \pm \delta^{\hbar} \boldsymbol{e}_i$ where $\boldsymbol{c}^{\hbar}$ is the midpoint of $\hbar$ and $\boldsymbol{e}_i$ is the i-th unit vector, $i \in \mathcal{D}^{\hbar}$. Calculate $w_i = \min\left(f(\boldsymbol{c}^{\hbar} + \delta^{\hbar} \boldsymbol{e}_i), f(\boldsymbol{c}^{\hbar} - \delta^{\hbar} \boldsymbol{e}_i)\right)$, $i \in \mathcal{D}^{\hbar}$, and partition $\hbar$ into thirds along each $i \in \mathcal{D}^{\hbar}$ according to the ascending order of w_i – starting with the dimension having the smallest w_i and continuing to the dimension having the largest w_i.

Step 4. *Stop test.* Update $f_{\min}$. Terminate the algorithm if the maximum number of iterations allowed is reached; otherwise set $t = t + 1$ and go back to Step 2.

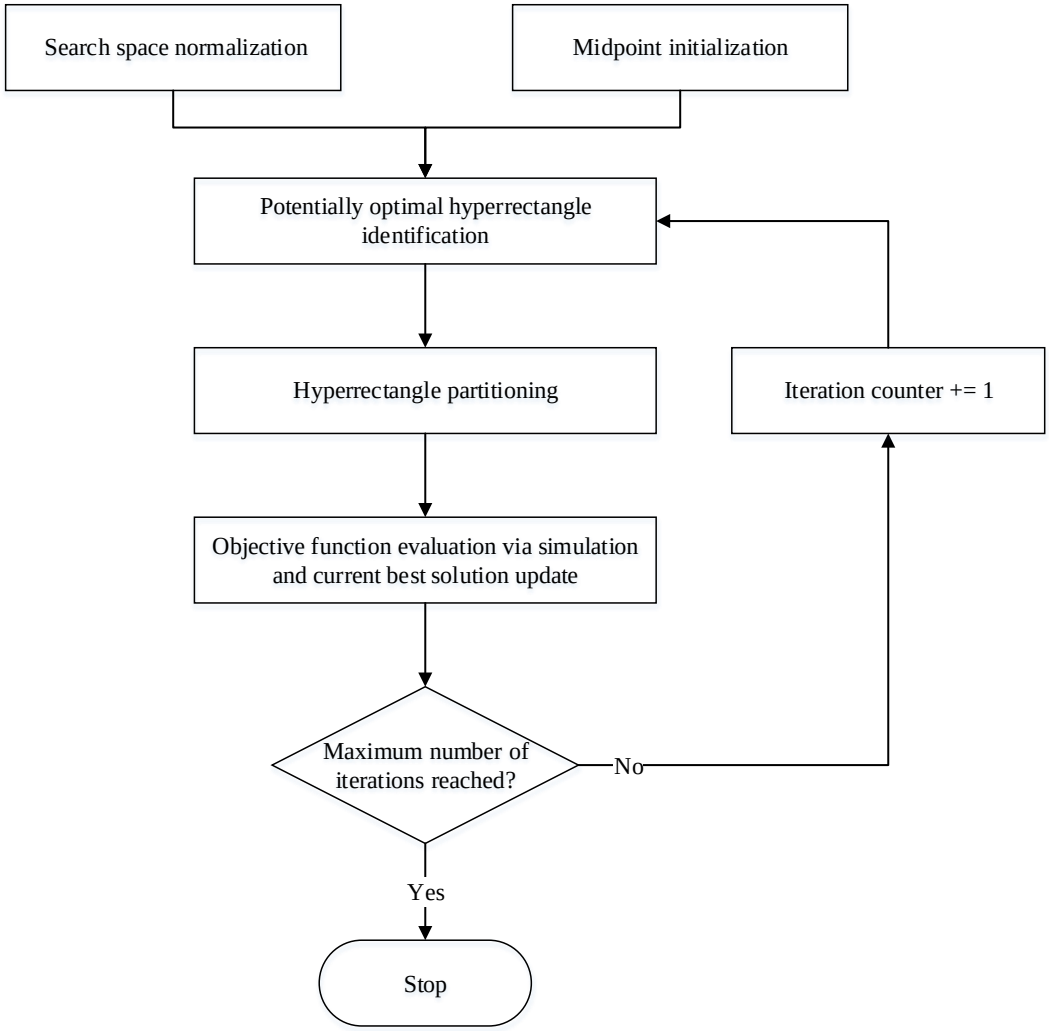


Figure 6.2 Flowchart representation of the DIRECT-enabled SO framework

Note that if one plots $\left(d_j, f\left(\boldsymbol{c}_j\right)\right)$ for all the hyperrectangles during an iteration, Equation (6.3) equates to finding the lower right convex hull of the dots, see Figure 6.3(a), while Equation (6.4) requires that the current best solution be exceeded by a nontrivial amount so as to avoid unnecessary local exploitation (Jones et al., 1993). Figure 6.3(b) shows a graphical representation of hyperrectangle partitioning in the two-dimensional space as a special case.

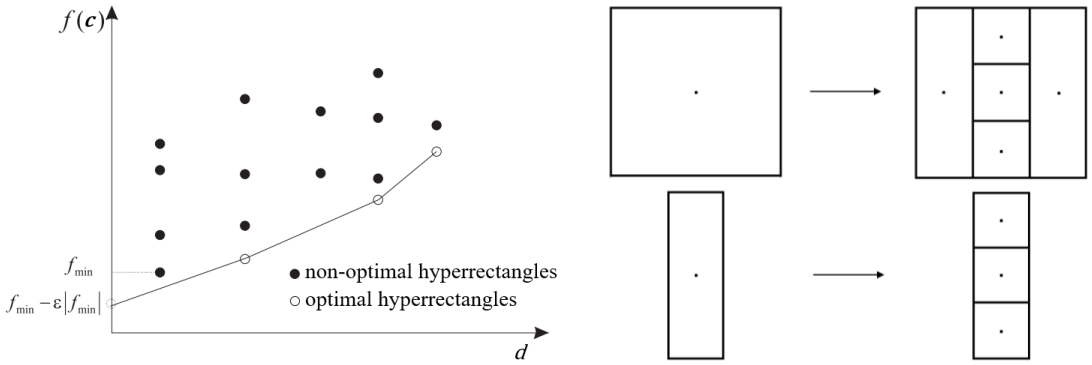


Figure 6.3 Graphical representation of (a) potentially optimal hyperrectangle identification, and (b) hyperrectangle partitioning in the two-dimensional space, modified based on Deng and Ferris (2007)

6.3. Solving the Simple Problem

Problem (4.1-4.3) with $m = 2$ and $K_{\mathrm{cr}} = 15$ vpkmpl is to be solved, respectively, by the four SO methods. Here, we consider the distance only toll and the tolling period covers the 8-9 AM peak period with two 30-min tolling intervals.

6.3.1. Proportional-Integral (PI) controller

As a deterministic method, the PI controller does not require multiple runs since there is no random component involved in searching for the optimum, although a bit effort is needed to tune the controller gain parameters. While a rule of thumb is provided in Section 4.2, we further show and compare Figure 6.4(a) and (b) to graphically interpret the effect of the gain parameters on the convergence. In Figure 6.4(a), we set $P_{\mathrm{P}} = 0.02$ and $P_{\mathrm{I}} = 0.005$, and the resultant convergence appears smooth without showing significant oscillatory behavior. In contrast, when we increase P_{P} and P_{I} to 0.1 and 0.03, respectively, in Figure 6.4(b), the toll rates undergo radical changes between successive iterations and hence, one can hardly tell what the convergent solution is especially for the

second tolling interval. Note that 30 function evaluations are enough to clearly demonstrate the oscillatory behavior in Figure 6.4(b). While making a guess for the solution is still possible by referring to the center lines drawn through the fluctuations, the accuracy or solution quality of the guess is by no means guaranteed and turns out to be much poorer than that in Figure 6.4(a). The result certainly highlights the importance of the parameter tuning step in the PI controller method.

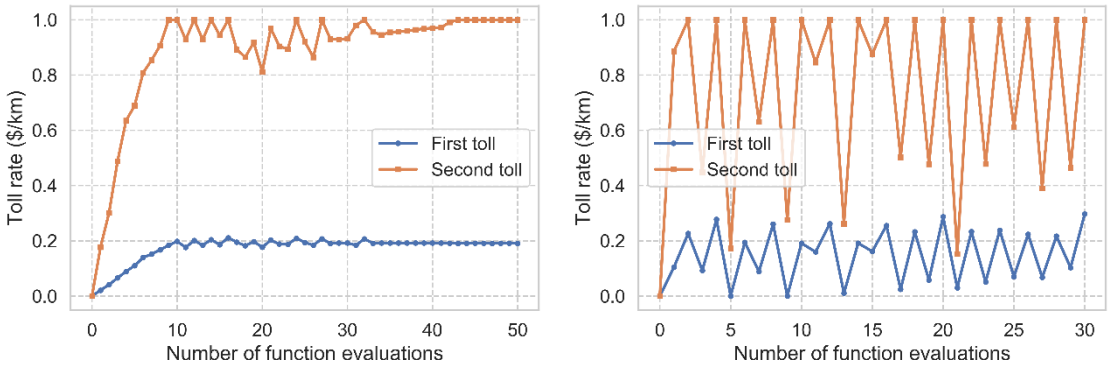


Figure 6.4 Convergence of the toll rates by the PI controller using (a) $P_{\mathrm{P}} = 0.02, P_{\mathrm{I}} = 0.005$, and (b) $P_{\mathrm{P}} = 0.1, P_{\mathrm{I}} = 0.03$

As shown in Figure 6.5(a), the average network densities of the PZ during both tolling intervals gradually reduce to and stabilize at about 15 vpkmpl (which is the critical network density as well as the pricing control threshold) as the number of function evaluations increases to 50. Accordingly, the objective function value corresponding to each function evaluation decreases as well to the ideal optimum of zero, although the generated decline curve appears quite non-smooth. The reason why this non-smoothness occurs is because of the existence of the numerical noise. We observe from Figure 6.4(a) that the toll rates quickly converge to their respective optimal values within about 10 function evaluations and only undergo minor changes afterwards. This is somewhat contradictory to the result in Figure 6.5(a) – the fluctuations in the objective function values prevail and

do not disappear at least for 30 function evaluations, which is also confirmed in Figure 6.5(b) showing the convergence of the optimal objective function value. What all this suggests is that there are still significant changes in the objective function values between the 10th and 30th function evaluations while the toll rates vary little. Figure 6.5(c) provides a graphical support for our argument by showing the search path of the PI controller method. As expected, the search path quickly orients towards an optimal region which, however, contains a wide range of function values despite being relatively small.

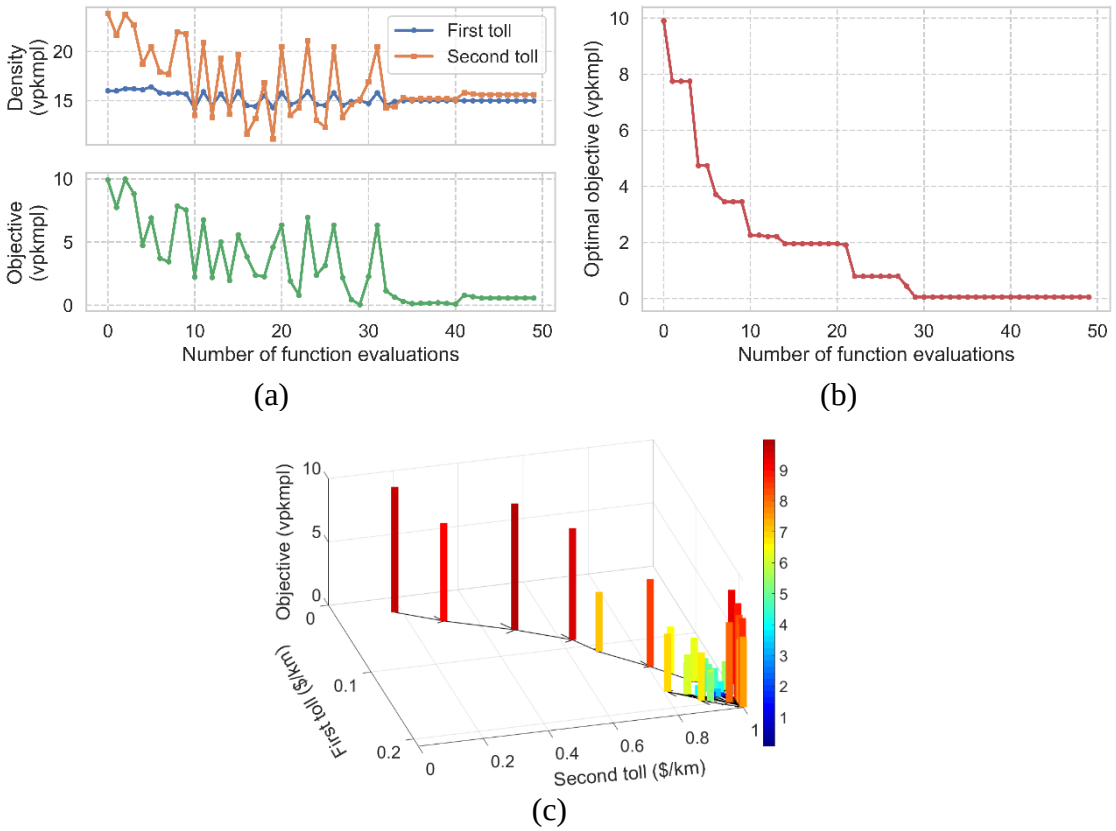
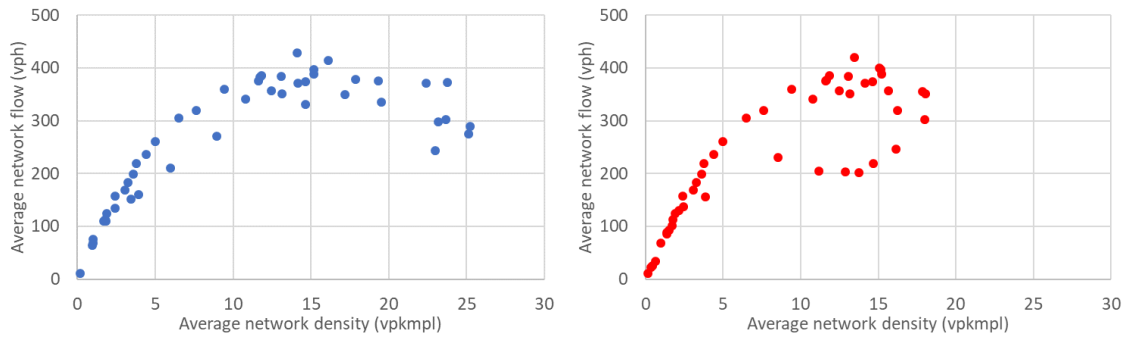


Figure 6.5 (a) Densities and objective function values as the number of function evaluations increases, (b) convergence of the optimal objective function value, and (c) search path of the PI controller method

Given that we keep track of the current best solution throughout the iterations, the optimal toll rates by the PI controller method are 0.19 $/km for the first tolling interval and 0.93 $/km for the second. The corresponding optimal objective function value is

0.055. As shown in Figure 6.6(b), the NFD after the optimal toll rates are applied successfully operates around the critical network density without entering the congested regime that appears in the non-tolling NFD, see Figure 6.6(a). There is, however, a large hysteresis loop in the tolling NFD, as expected, and part of the reason lies in the distance only toll per se – drivers tend to accumulate themselves into the shortest paths within the PZ resulting in a more heterogeneous distribution of congestion and hence a larger hysteresis loop in the NFD. This is one of our major findings in CHAPTER 4 and the result here is consistent further supporting our argument.



(a) (b)

Figure 6.6 NFDs of the PZ before and after the optimal toll rates are applied

6.3.2. Regressing Kriging (RK)

RK is an SO method that does not need parameter tuning. Instead, it requires several parameters to be estimated throughout the iterations via MLE so as to generate the optimal infill sample points for augmenting the response surface. Since the first step of RK is to generate an initial set of sample points through random sampling, and the GA is used to find both the MLEs of the parameters and the optimal infill sample points, we choose to perform multiple runs to take into account this effect of randomness.

As a means of validating the convergence of RK, the probabilistic EI metric is calculated and maximized during each iteration to generate an optimal infill sample point

and hence, does not apply to the initial sample points that comprise the first 11 function evaluations. Figure 6.7 shows, for each different run, the expected decreasing trend in the probabilistic EI metric as the number of infill function evaluations increases. While the shape of the decline curve appears irregular and differs from run to run for the first few infill function evaluations, all the curves quickly drop and stabilize at almost zero suggesting that the method is unable to find another sample point that significantly improves the current best solution, and that we can terminate the iterations with confidence. In general, with 11 initial function evaluations and 20 or less infill function evaluations (i.e. about 30 function evaluations in total), the method can be considered convergent by referring to the probabilistic EI metric.

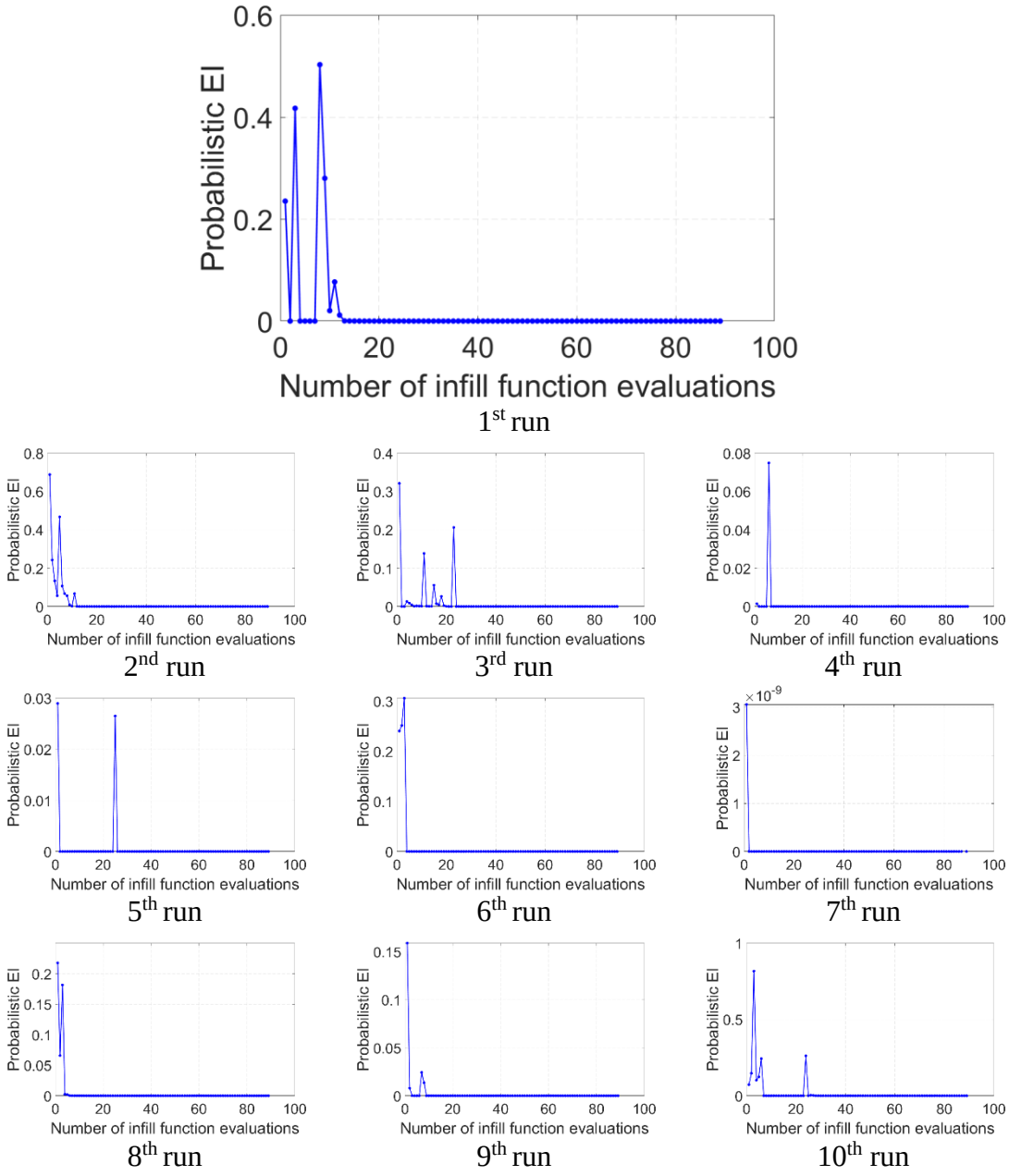


Figure 6.7 Validating the convergence of RK using the probabilistic EI metric for multiple runs

To validate the accuracy of RK, leave-one-out CV is performed for each different run and the results are shown in Figure 6.8. Clearly, all the constructed response surfaces work well – the points mapping predictions to observations lie neatly along the equal line and the standardized cross-validated residuals lie perfectly within $[-3,3]$ – suggesting

that at this stage, they can be used to approximate $DTA(\cdot)$, the "black box" function of the simulation model in Equation (4.2), and make accurate predictions.

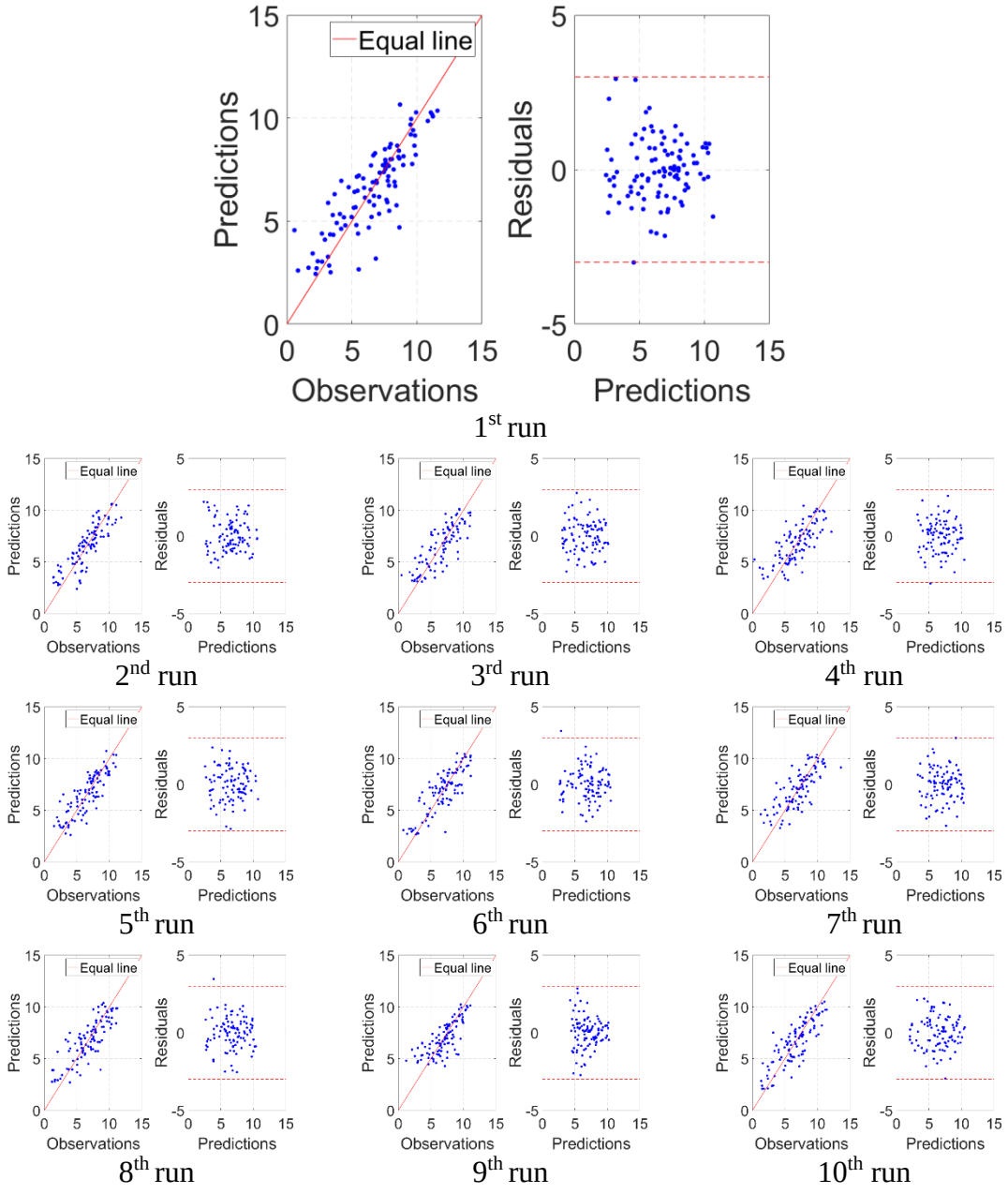


Figure 6.8 Validating the accuracy of RK using the standardized cross-validated residuals for multiple runs

The constructed response surface for each run is shown in Figure 6.9 as a two-dimensional heatmap together with the distribution of the 100 sample points. Two key observations are drawn below:

- The sample points for each run are distributed widely across the entire search space manifesting the global exploration property of the method. As we will compare and show later, if one applies RK without using the reinterpolation technique discussed in Sub-section 3.4.3, the method is prone to local exploitation resulting in a highly biased response surface.
- The constructed response surfaces for all runs exhibit a similar pattern featuring a common narrow strip of global optimal region centering around an abscissa value (i.e. the toll rate for the first tolling interval) of 0.2 $/km. The method, despite having random sampling and random search components, is therefore able to produce consistent results across multiple runs.

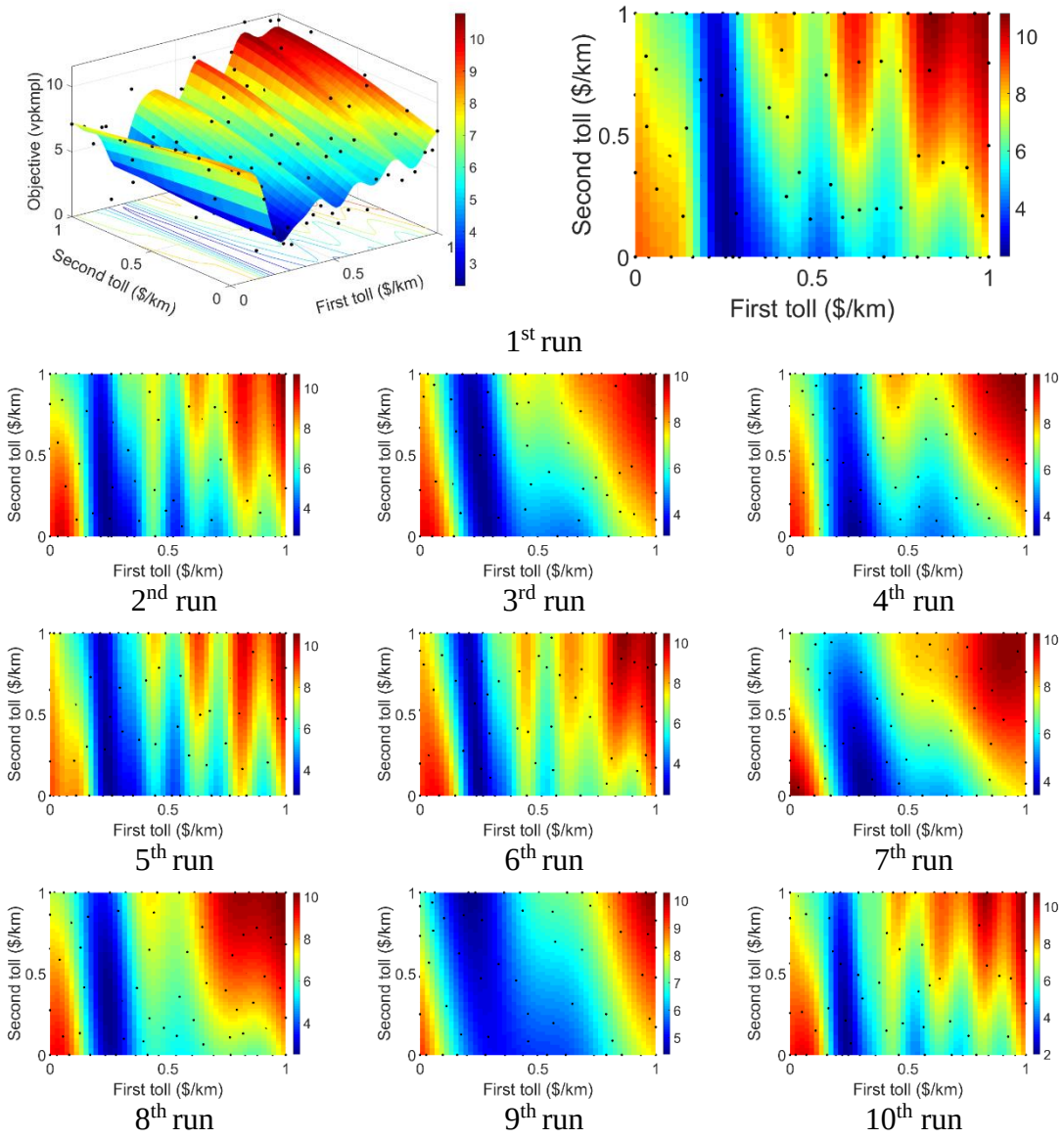
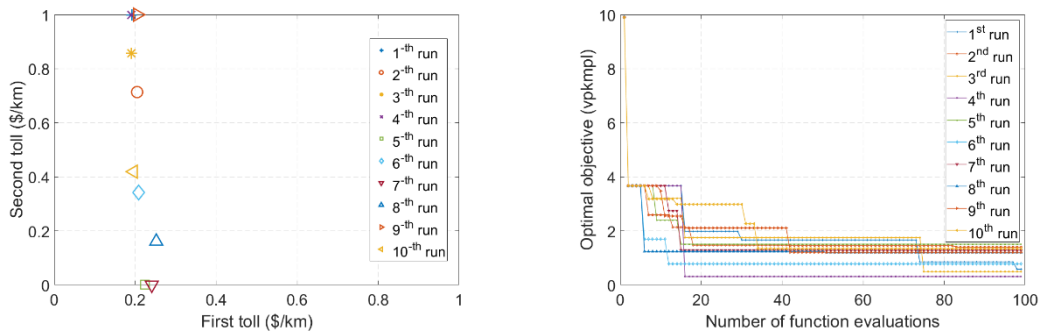


Figure 6.9 Constructed response surfaces for multiple runs represented as two-dimensional heatmaps where the black dots are the sampled and evaluated points

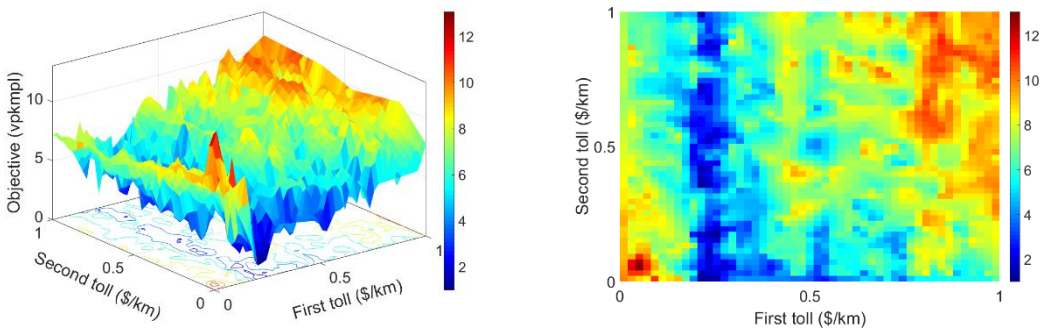
As expected, the optimal solutions from all runs lie within the identified narrow strip of global optimal region as shown in Figure 6.10(a). Although Figure 6.10(b) shows that the objective function does not converge to a same optimal value for multiple runs, the differences are relatively small and are partially attributed to the noisiness of the objective function per se. This is highlighted in Figure 6.11 where we construct the response surface by interpolation using $100 \times 10 = 1{,}000$ sample points from all runs. Clearly, with many more sample points, the interpolated response surface is much noisier than

those shown in Figure 6.9. One major reason, similar to our discussion on Figure 6.5(c), is the higher numerical noise embedded in this larger number of sample points. If one uses interpolation to construct the response surface just like what we did in Figure 6.11, the problem of overfitting is likely to occur because interpolating every single sample point is, unfortunately, equivalent to modeling the high numerical noise rather than filtering it out. This certainly supports our discussion in Sub-section 3.4.2 – we need a regression method like RK rather than an interpolation method like ordinary Kriging when the objective function is highly noisy. Note from Figure 6.10(b) that the optimal objective function value reduces mostly within the first 30 or so function evaluations and afterwards, reduces only slightly or remains unchanged. This is consistent with our previous finding from Figure 6.7 that the probabilistic EI metric generally converges to zero within about 30 function evaluations.



(a) (b)

Figure 6.10 (a) Distribution of the optimal solutions and (b) optimal objective function values when applying RK for multiple runs as the number of function evaluations increases



(a) (b)

Figure 6.11 (a) Three-dimensional and (b) two-dimensional representations of the interpolated response surface using 1,000 sample points from all runs

Finally, let us discuss the importance of the reinterpolation technique and why we need it as part of RK. By comparing Figure 6.12(a) and (b), which show the constructed response surface if RK is applied without using the reinterpolation technique, with Figure 6.9, one can immediately recognize the unduly strong local exploitation property due to the lack of reinterpolation as most sample points are gathered in a small area that is only part of the previously identified narrow strip of global optimal region. This further supports our discussion in Sub-section 3.4.3 that reinterpolation can help RK escape from a local optimal region and regain the global exploration property. While the method is still able to generate an optimal solution (one that is likely to be a local optimum), the constructed response surface is highly biased that predicts poorly for other parts of the search space. This is what we call underfitting due to the lack of global exploration. Figure 6.12(c) shows that the probabilistic EI metric without reinterpolation simply fluctuates without converging to the desirable zero, at least within the same 89 infill function evaluations. The cross-validated residuals as shown in Figure 6.12(d) go beyond $[-3,3]$ for some predictions, as expected. If one wishes to predict further for a few other points especially outside the identified small area, the predictions are likely to be significantly different from the true values with associated cross-validated residuals lying out of the desirable $[-3,3]$.

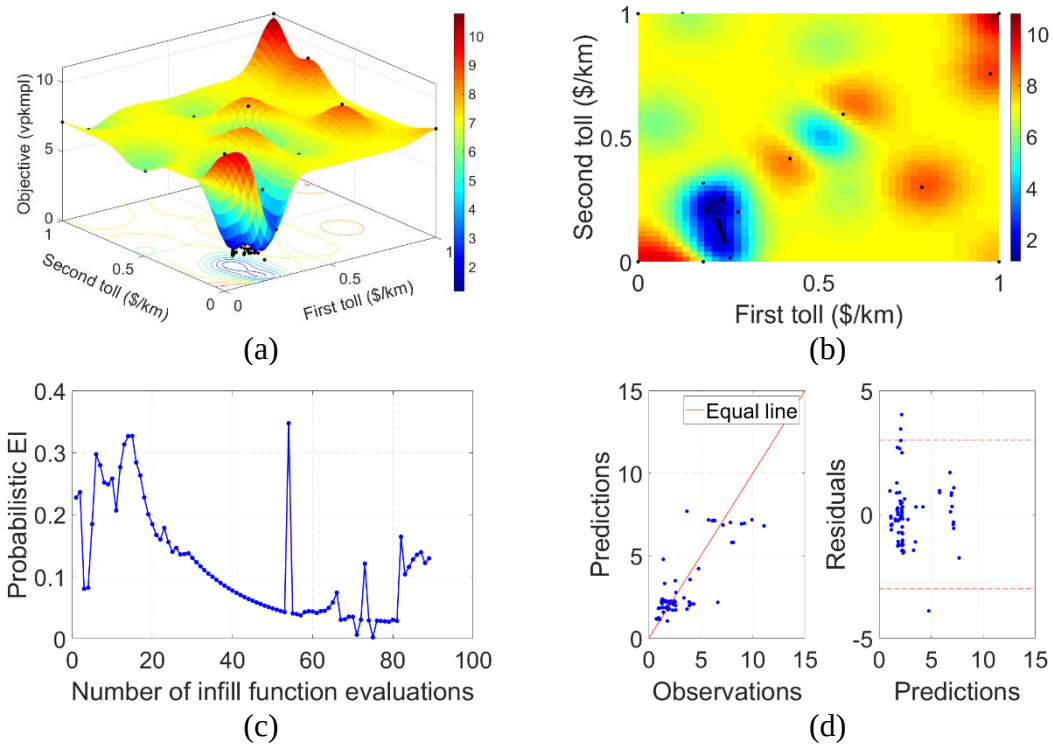


Figure 6.12 Applying RK without using the reinterpolation technique: (a) and (b) three- and two-dimensional representations of the constructed response surface, (c) variations of the probabilistic EI metric, and (d) predictions and cross-validated residuals

6.3.3. Simultaneous Perturbation Stochastic Approximation (SPSA)

SPSA is a gradient approximation method that entails parameter tuning. One needs to be very careful when tuning the parameters especially in the presence of high numerical noise. As discussed in Section 6.1, there are in total five user-specified parameters – a, c, α, γ, A – along with their respective selection guidelines. In short, α and γ can somewhat be treated as fixed parameters and set to 0.602 and 0.101, respectively (Spall, 1998), while A and a can be determined without much difficulty given the problem at hand as well as users' preferences. The parameter c which controls the accuracy of gradient approximation seems to require a bit more tuning effort. In general, c should be set close to zero if the numerical noise in the objective function is low. In the presence of a

higher level of numerical noise, c should be set farther from zero and a smaller a is advisable as well.

Since SPSA replies on gradient approximation, it should be perceived indeed as a local optimizer. To obtain a global optimizer, one can inject Monte Carlo noise into the right-hand side of Equation (6.2) to provide the needed "bounce" for the search to jump out of the possible local optima and hence to avoid premature entrapment (Maryak and Chin, 2008). This is, however, not the only way to achieve global optimization. It turns out that the basic SPSA without injected Monte Carlo noise can often be treated as a global optimizer as well due to the simultaneous perturbation for gradient approximation (Maryak and Chin, 2008). Specifically, Equation (6.2) can be re-expressed as:

$$\boldsymbol{\tau}_{i+1} = \boldsymbol{\tau}_i - a_i \mathbf{g}_i(\boldsymbol{\tau}_i) + a_i \boldsymbol{\epsilon}_{\text{noise}} + a_i \boldsymbol{\epsilon}_{\text{perturbation}} \tag{6.5}$$

where $\mathbf{g}_i(\cdot)$ is the true gradient, $\boldsymbol{\epsilon}_{\text{noise}}$ is the noise-incurred difference from $\mathbf{g}_i(\cdot)$, and $\boldsymbol{\epsilon}_{\text{perturbation}}$ is the difference from $\mathbf{g}_i(\cdot)$ arising from the simultaneous perturbation for gradient approximation. The term $a_i \boldsymbol{\epsilon}_{\text{perturbation}}$ offers similar statistical functionality to the injected Monte Carlo noise and hence, provides the basic SPSA with the needed "bounce" already for global optimization. Therefore, in this case, c should not be set close to zero so as to enable the proper functioning of $a_i \boldsymbol{\epsilon}_{\text{perturbation}}$ as a noise injector and to achieve a global optimizer. Compared with SPSA with injected Monte Carlo noise, the basic SPSA has, in theory, a much faster rate of global convergence and fewer user-specified parameters, although the former exhibits broader applicability in general (Spall, 2003). Note that one must specify an initial point when applying SPSA and hence, different initializations can be considered.

Figure 6.13 shows the results of SPSA under six scenarios with different c's and initial points $\boldsymbol{\tau}_0$'s, while all the other parameters stay the same, i.e. $\alpha = 0.602$, $\gamma = 0.101$, $A = 5$, and $a = 0.1$. Clearly, when (and only when) $c = 0.025$ and $\boldsymbol{\tau}_0 = [0.75,0.75]^{\mathrm{T}}$, SPSA is unable to lead the search towards the identified narrow strip of global optimal region in Figure 6.9. This is consistent with our previous discussion on how to choose c. We already observe from Figure 6.11 that the objective function is highly noisy, and c should therefore not be set close to zero with the aim of filtering out, to some extent, the numerical noise in the gradient approximation and guiding the search in a right direction. We also observe that apart from the narrow strip of global optimal region, many other local optimal regions exist resulting in a very high chance of getting trapped in one of them, which is exactly the case in Figure 6.13(f). $c = 0.025$ is simply not "strong" enough to provide the needed "bounce" for SPSA to escape from a local optimum and to act as a global optimizer, not to mention that $\boldsymbol{\tau}_0 = [0.75,0.75]^{\mathrm{T}}$ is relatively far away from the global optimal region.

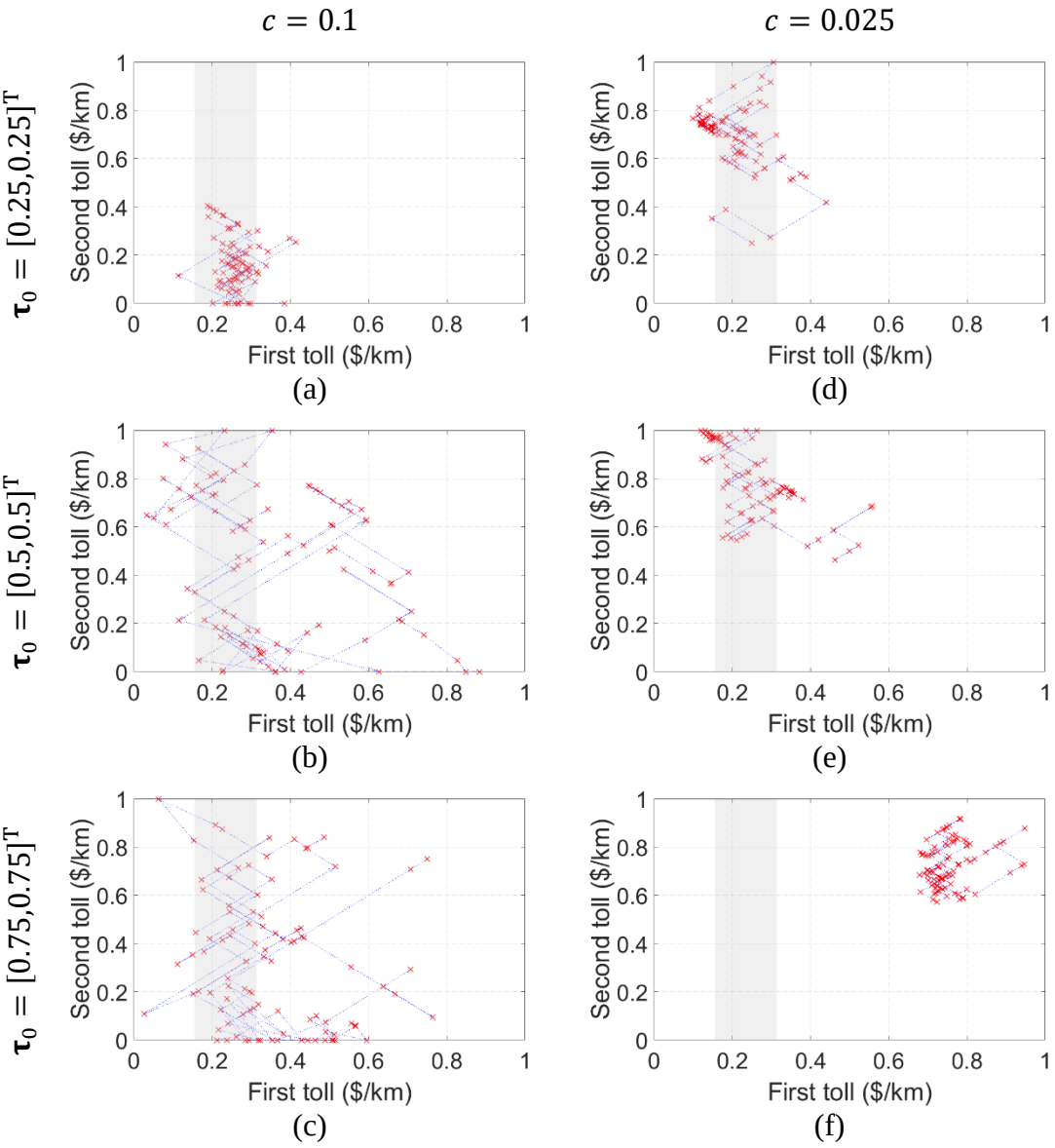


Figure 6.13 Effects of different parameter settings on the performance of SPSA where the red crosses are the calculated and evaluated points along the search paths and the shaded area roughly represents the narrow strip of global optimal region previously identified in Figure 6.9

In contrast, when we increase $c = 0.1$, SPSA quickly orients its search path towards the global optimal region rather than wandering around a local optimum. Perhaps the only concern is the resultant bigger move per iteration as is also observed in Figure 6.13(b), even though a_i in Equation (6.2) which controls the step size at every iteration is a decreasing function of the number of iterations. The reason is twofold:

- A larger c generally results in a larger magnitude of the gradient approximation around the narrow strip of global optimal region, as compared with that in other parts of the search space.
- The objective function is highly noisy that possibly increases the magnitude of the gradient approximation.

Figure 6.14(a-c) show that the magnitude of the gradient approximation keeps changing at every iteration and does not decay at least within the 100 iterations or, equivalently, 200 function evaluations. The difference between the objective function values corresponding to the two perturbed points at every iteration exhibits a similar non-decaying pattern as well, see those red vertical lines in Figure 6.14(d-f).

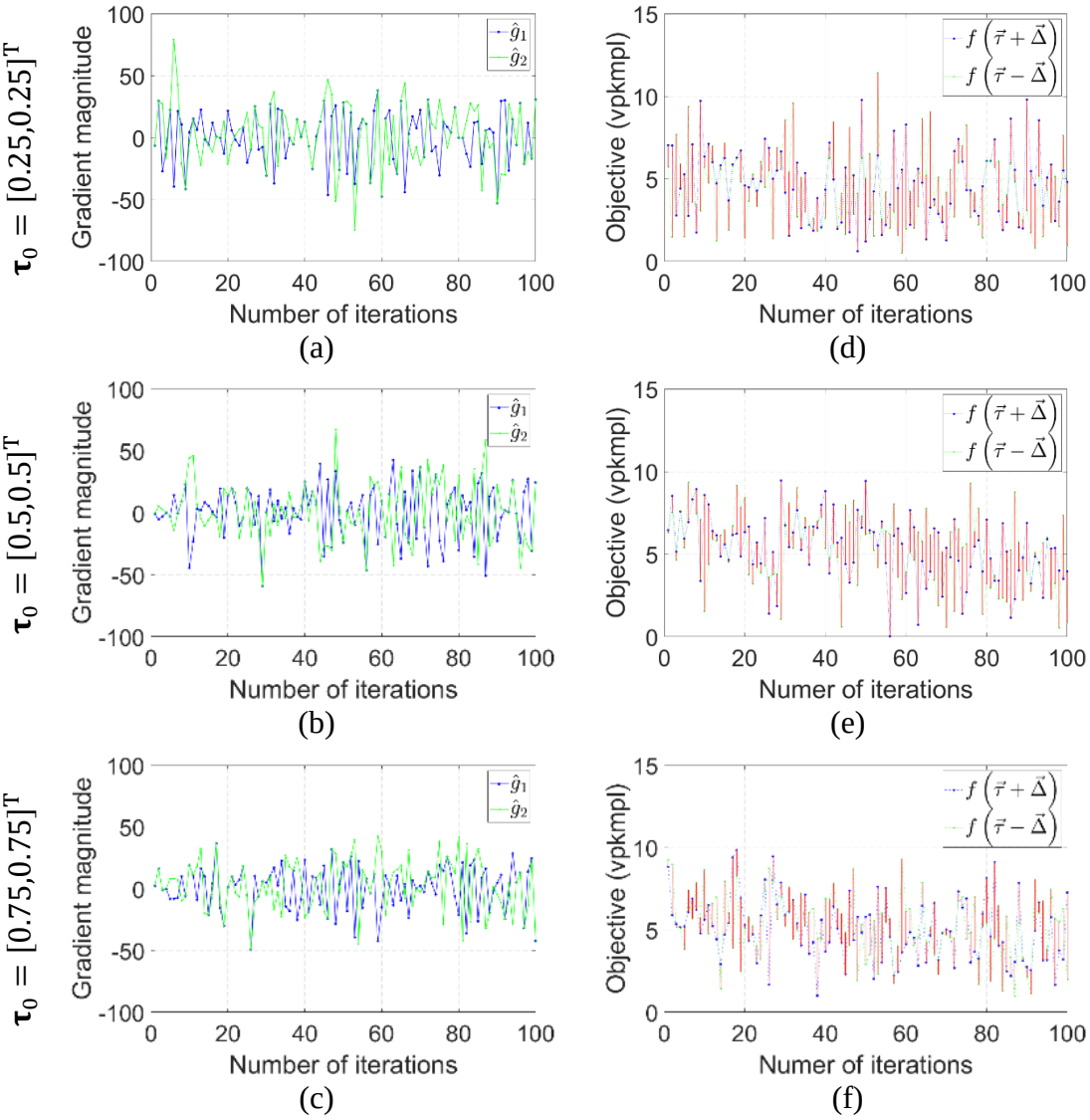


Figure 6.14 Results of SPSA with different initial points at every iteration along the search path: (a-c) magnitude of the gradient approximation, and (d-f) differences between the objective function values corresponding to the two perturbed points

To make moves smaller and hence less fluctuating, one can always choose to set a to a smaller value or, alternatively, scale down the gradient approximation, both of which have the same effect of reducing the step size along the search path at every iteration. As a comparative example, Figure 6.15(a) shows the result of SPSA when all the parameters are kept the same as in Figure 6.13(b) except that the gradient approximation is scaled down. The moves clearly become less "aggressive" and hence no longer "bounce" back and forth around the narrow strip of global optimal region. All this, however, does

not suggest that the gradient approximation is (asymptotically) decaying to the ideal zero, see Figure 6.15(b), neither is the difference between the objective function values at the two perturbed points, see Figure 6.15(c). In fact, even if one uses a small c value and the search path successfully reaches the global optimal region, the presence of high numerical noise makes it almost impossible for the two to vanish.

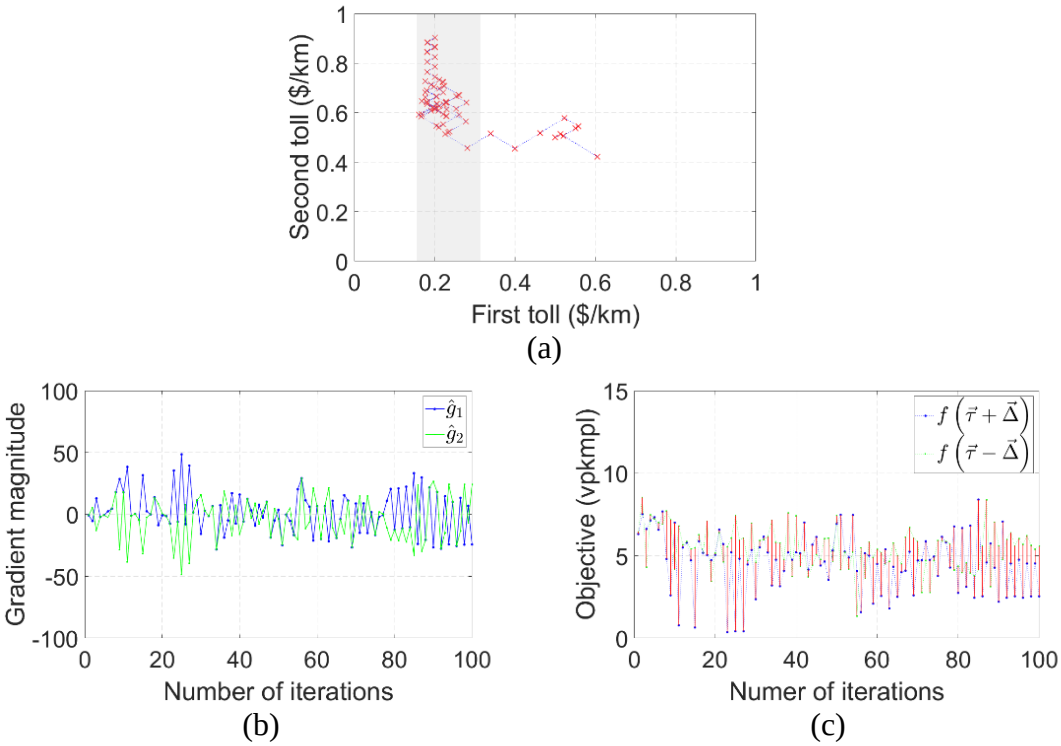


Figure 6.15 Effects of scaling down the gradient approximation on the performance of SPSA with $c = 0.1$ and $\boldsymbol{\tau}_0 = [0.5,0.5]^{\mathrm{T}}$

A notable concern about SPSA is that the method only evaluates the two perturbed points at every iteration but not the points along the search path. Therefore, to determine the optimal solution and the associated objective function value, one may need to spend a bit more effort evaluating some of the points presumably at the tail of the search path. Unfortunately, this can be indeterminate in the presence of high numerical noise. Perhaps a more practical strategy is to keep track of the objective function values at those

perturbed points and consider the best solution as the optimum. As shown in Figure 6.16, the optimal objective function values for different initial points all reduce mostly within the first 50 function evaluations and remain almost constant after 100 function evaluations. This turns out to be a very similar result compared with RK in Figure 6.10(b).

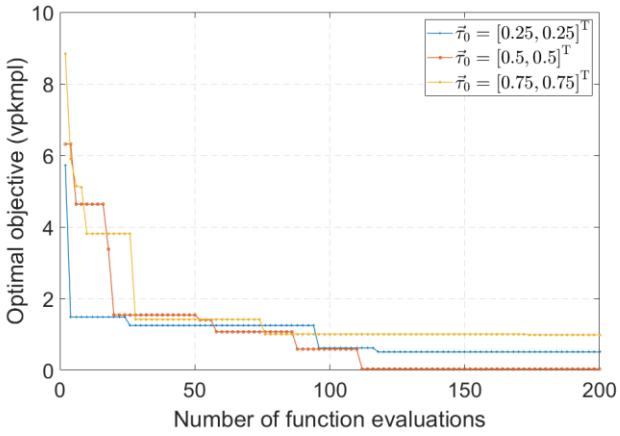


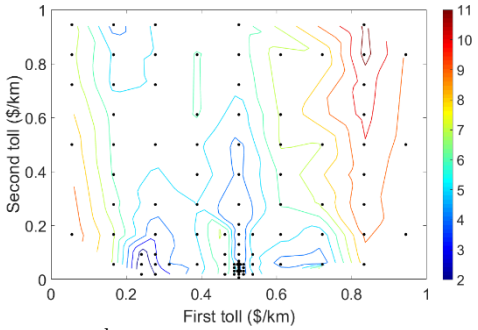
Figure 6.16 Optimal objective function values when applying SPSA with different initial points as the number of function evaluations increases

6.3.4. DIviding RECTangles (DIRECT)

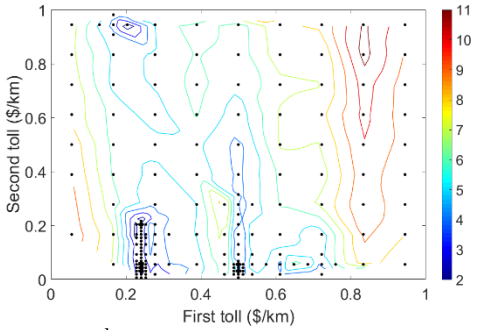
While DIRECT originates from Lipschitzian optimization, it is akin to the well-known direct search methods – another big category of SO – as both only require direct function evaluations. However, unlike most direct search methods that function as a local optimizer, DIRECT is a global optimization method and involves perhaps the least parameter tuning effort among the four SO methods. The only user-specified parameter, ε in Equation (6.4), turns out to have limited effect on the performance of DIRECT and hence can be fixed to a value, e.g. ranging from 10^{-7} to 10^{-3} (Jones et al., 1993). Since DIRECT is a deterministic method, there is no need to perform multiple runs.

Figure 6.17 shows multiple interpolated contour plots of the objective function as DIRECT proceeds with its iterations by sampling and evaluating more and more points.

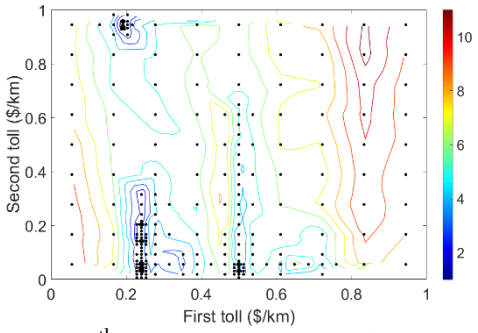
As shown in Figure 6.17(e), the final 35th iteration evaluates a total of 669 sample points and produces a contour plot that highly resembles Figure 6.11(b) – the narrow strip of global optimal region is readily recognizable. As far as this many function evaluations are concerned, the optimal toll rates are 0.19 \$/km for the first tolling interval and 0.95 \$/km for the second. The corresponding optimal objective function value is 0.005 vpkmpl which is extremely close to the ideal zero. Nevertheless, within about 100 function evaluations as shown in Figure 6.17(a), DIRECT is able to step into the global optimal region (more accurately, the lower part of it) producing an optimal solution of $[0.24,0.06]^{\mathrm{T}}$ and an optimal objective function value of 0.874. With this many function evaluations, the solution quality is roughly the same as that offered by RK (see Figure 6.10) and SPSA (see Figure 6.16). We do notice that a greater part of the computational effort is spent exploiting a local optimal region in the lower middle part of the search space. But, starting from Figure 6.17(b) and towards Figure 6.17(e), the global exploration property of DIRECT becomes increasingly prominent and the outline of the global optimal region gradually takes shape.



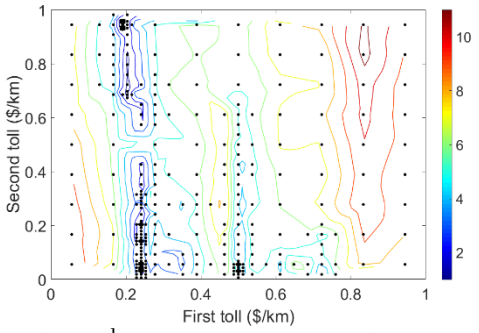
(a) 11th iteration, 103 sample points



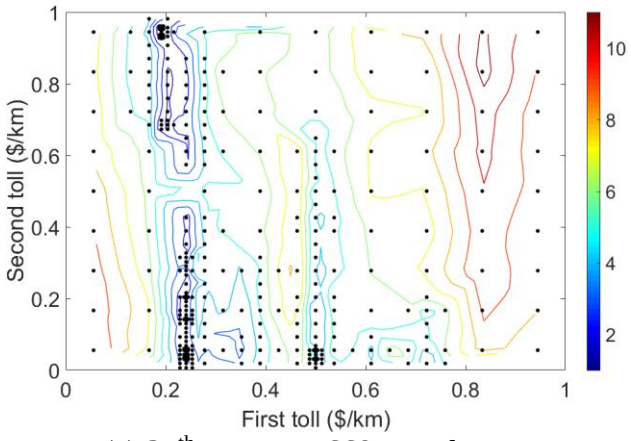
(b) 20th iteration, 217 sample points



(c) 26th iteration, 309 sample points



(d) 33rd iteration, 415 sample points



(e) 35th iteration, 669 sample points

Figure 6.17 Interpolated contour plots of the objective function for a few selected iterations of DIRECT where the black dots are the sampled and evaluated points

Figure 6.18 shows how the optimal objective function value reduces as the number of function evaluations increases. As we previously touched upon, 100 function evaluations are enough to locate a decent solution in the global optimal region, although not necessarily being a global one in an absolute sense. This result is similar to what we achieve by applying RK and SPSA. With more function evaluations, the optimal objective

function value reduces further, as expected, and eventually converges to almost zero as the absolute global optimum.

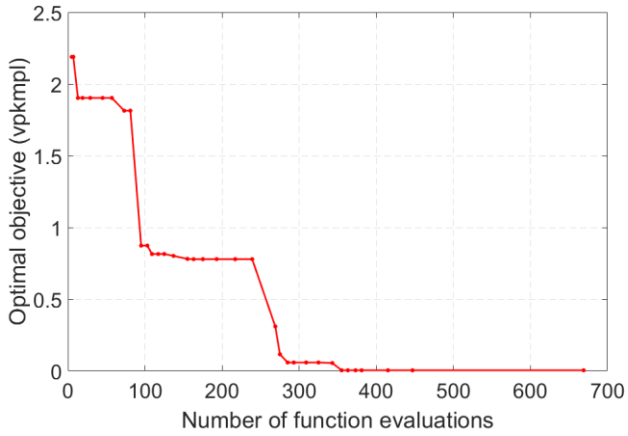


Figure 6.18 Optimal objective function values when applying DIRECT as the number of function evaluations increases

6.4. Solving the Complex Problem

When formulating the complex TLP, we assume that the shape of the NFD including the critical network density does not change significantly when the network is pricing-controlled. This assumption, however, does not always hold and can truly affect the effectiveness of the optimal tolls. Therefore, instead of specifying a critical network density around which the network should operate, we now look directly at the network flow which is to be maximized throughout the tolling period. With $m = 8$ and $K_{\mathrm{cr}} = 25$ vpkmpl, Problem (5.1-5.5) with a modified direct flow maximization objective is to be solved, respectively, by the three SO methods excluding the PI controller due to its inability to consider complex objective functions and constraints. Here, we consider the more efficient and equitable JDDT to further increase the problem dimension and

complexity. The tolling period now covers the 8-10 AM peak period with eight 15-min tolling intervals.

6.4.1. Regressing Kriging (RK)

When applying RK to solve the complex TLP, we incorporate the toll pattern smoothing constraints into the GA, thereby narrowing down the feasible domain when searching for the optimal infill sample point at each iteration. As before, we use 100 function evaluations as the computational budget and perform three runs to consider the randomness effect.

Figure 6.19(a) shows how the optimal objective function values increase as the number of function evaluations increases. There is a gradual increasing trend, as expected, toward the end of the 100 function evaluations because we are now directly maximizing the network flow throughout the tolling period. Figure 6.19(b) shows the simulated NFDs of the PZ under the optimal tolling scenarios where the network flow maintains almost at its maximum for a range of density values (roughly ranging from 20 to 40 vpkmpl). We have already observed that without pricing, the critical network density of the NFD is somewhere between 20 and 30 vpkmpl. Result here, however, suggests that the network density can be allowed to evolve to 40 vpkmpl while maintaining a high network flow. Given the objective of direct flow maximization, one need not to struggle with specifying the critical network density around which the network should be pricing-controlled. There is also no need to assume that the shape of the NFD including the critical network density does not change significantly when the network is pricing-controlled. This assumption, as we previously argued, does not always hold.

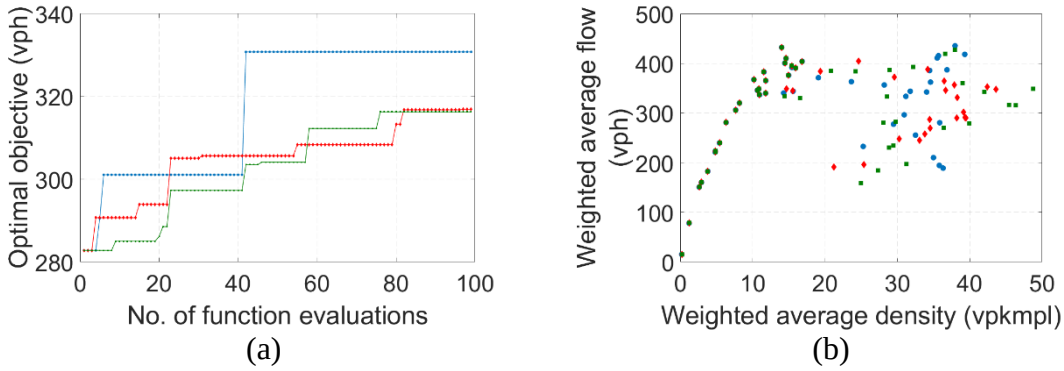


Figure 6.19 Applying RK to directly maximize the network flow: (a) how the optimal objective function values change as the number of function evaluations increases, and (b) simulated NFDs of the PZ under the optimal tolling scenarios

To demonstrate that the shape of the NFD after pricing might change affecting the effectiveness of the optimal tolls, we further apply RK to solve the same problem except that we now aim to operate the network around the specified critical network density (25 vpkmpl) rather than directly maximizing flow. Similar to Figure 6.19(a), Figure 6.20(a) shows that RK can quickly orient its search toward the global optimum within 100 function evaluations. However, the resulting NFDs as shown in Figure 6.20(b) exhibit an unexpected and undesirable shape. Although the network is well controlled around the critical network density, the NFDs have already entered the congested regime leading to a rather low network flow. The new critical network density appears to be 15 vpkmpl and the previous 25 vpkmpl no longer qualifies. Compared with the trapezoidal shape of the non-tolling NFD, the NFD after pricing has been somewhat squeezed to the left taking on a triangular shape. The optimal tolls are by no means optimal given this change in the shape of the NFD. While one may argue that this issue can be resolved by increasing the critical network density, there is no guarantee that the new critical network density can result in a similar shape of the NFD before and after pricing. Therefore, specifying the critical network density might become a tedious trial-and-error process.

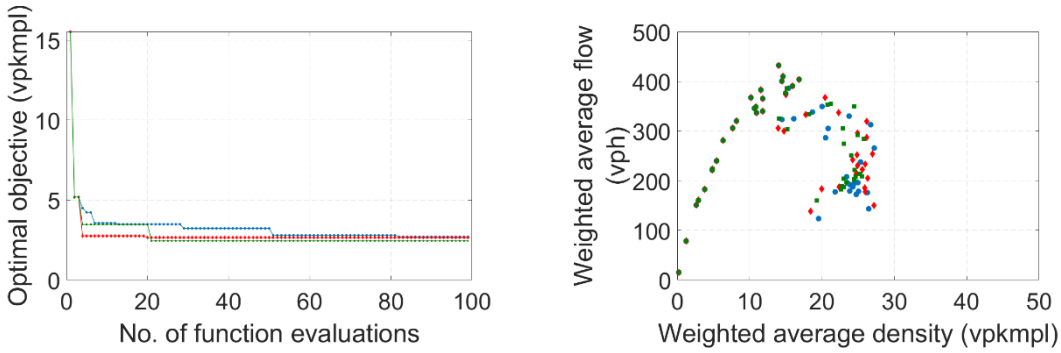


Figure 6.20 Applying RK to achieve the critical network density: (a) how the optimal objective function value changes as the number of function evaluations increases, and (b) simulated NFDs of the PZ under the optimal tolling scenarios

6.4.2. Simultaneous Perturbation Stochastic Approximation (SPSA)

Due to the presence of the toll pattern smoothing constraints, SPSA cannot be applied directly to solve the complex TLP. We therefore integrate SPSA with the penalty function method (Bazaraa et al., 2013) to transform the original constrained optimization problem into an unconstrained one. The penalty parameter here is set at 10 to be consistent with DIRECT. In a nutshell, if the toll pattern smoothing constraints are violated, there will be a penalty imposed on the objective function value forcing SPSA to search in the space where the constraints are satisfied.

Figure 6.21 shows how SPSA performs assuming different initial points. There is an expected increasing trend in the optimal objective function values as the number of function evaluations increases toward 200. A comparison between Figure 6.21 and Figure 6.22 suggests that in general, SPSA can improve the optimal objective function value at a faster rate than DIRECT. However, different initial points may affect the search for the optimal solution and an ill-conditioned initial point is likely to trap the search at a bad local optimum. When comparing Figure 6.21 with Figure 6.19(a), we find that RK is the best-performing method for solving the complex TLP which improves the optimal objective function value at a much faster rate than either SPSA or DIRECT.

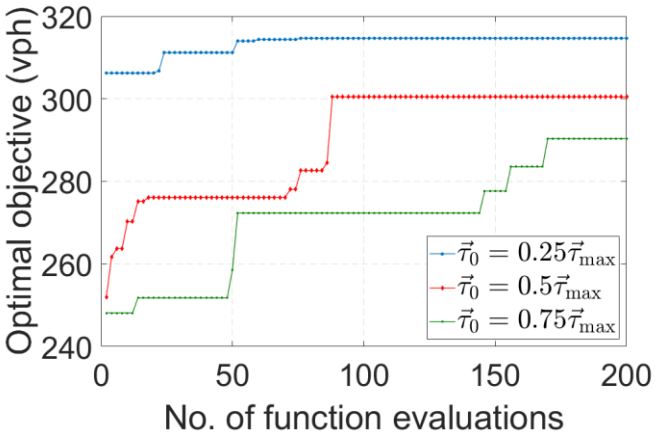


Figure 6.21 Applying SPSA to solve the complex TLP: how the optimal objective function values change as the number of function evaluations increases

6.4.3. DIviding RECTangles (DIRECT)

When applying DIRECT to solve the complex TLP, we again employ the penalty function method to deal with the toll pattern smoothing constraints. Figure 6.22 shows that the optimal objective function value gradually increases as the number of function evaluations increases. The penalty parameter is set at 10, but the result does not change much when we use 1, 50, and 100. When comparing Figure 6.22 with Figure 6.19(a), we find that DIRECT performs surprisingly slowly in improving the objective function value. We have already observed that within 100 function evaluations, RK can increase the optimal objective function value to at least almost 320 vph. However, DIRECT with the same 100 function evaluations can only improve the optimal objective function value to about 285 vph. Even with nearly 600 function evaluations, the optimal objective function value is only slightly above 300 vph. A possible reason, as we previously discussed, is the existence of high numerical noise that cannot be tolerated by DIRECT. Another possible reason is the presence of the toll pattern smoothing constraints that restrict the feasible domain to a small part of the original design space. As such, DIRECT may require

a lot more function evaluations than RK given its iterative and exhaustive partitioning property without taking into account directly the feasibility constraints.

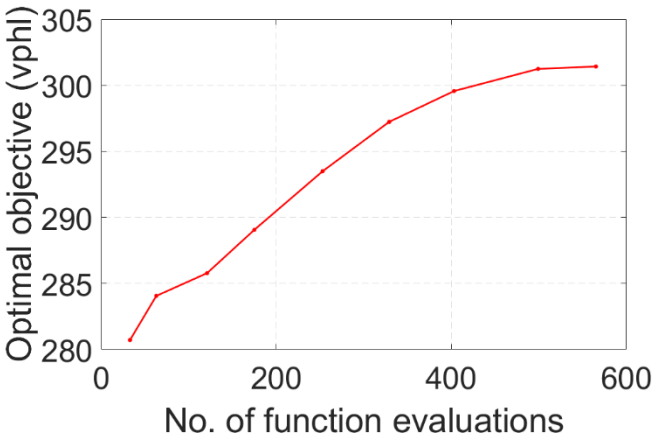


Figure 6.22 Applying DIRECT to solve the complex TLP: how the optimal objective function value changes as the number of function evaluations increases

6.5. Discussion

The four SO methods investigated and compared have their own distinct "intelligent" ways of leading the search for the global optimum and are, therefore considered representatives of computationally efficient SO methods. The applicability of these methods is not limited to solving TLPs. They can be applied to solve other types of NDPs that rely on computer simulation as well. The only requirement is perhaps that the defined objective and constraint functions should be evaluable by the simulation. Based on our results, a few recommendations are made for applying the four SO methods. These are authors' recommendations and hence by no means some universal laws that apply to every conceivable SO problem. We recommend considering some case-specific features or requirements as well if one is interested in applying one of these methods to solve the problem at hand.

When applied to solve the simple TLP, all the methods perform quite well, although, clearly, having their own pros and cons when compared with each other. If the optimization problem can be formulated as a simple problem having a low-dimensional decision vector, a set-point objective, and only bound constraints, the PI controller is particularly suited and will likely result in a much faster rate of convergence, although requiring a bit trial-and-error to tune the controller gain parameters. Note, however, that the PI controller is only able to search for a single global optimum. If multiple global optima coexist and one is interested in the overall distribution of the optimal solutions, RK and DIRECT are preferable and should be considered. When comparing RK, SPSA, and DIRECT, SPSA requires the greatest effort for parameter tuning which might be a major concern if the objective function is computationally expensive. Although having the potential to act as a global optimizer for many challenging optimization problems, it is not impossible to get trapped in a bad local optimum. This possibility might even be greater if the objective function is highly noisy and one accidentally chooses an ill-conditioned initial point that is far away from the global optimal region. A straightforward solution is to inject Monte Carlo noise into the basic SPSA to avoid premature convergence. This strategy, however, largely slows down the theoretical rate of convergence and requires even greater effort for parameter tuning. Finally, as with the PI controller, SPSA per run is only able to search (hopefully) for a single global optimum but not every one of them if multiple global optima coexist. A multi-start approach is handy but, again, significantly increases the computational intensity. RK and DIRECT are always preferred if one cares about the overall distribution of the optimal solutions rather than a single optimum. This is equivalent to providing multiple rather than a single choice for decision making. When the numerical noise in the objective function is not high, both SO methods should perform equally well. DIRECT might be a better option if parameter estimation required by RK

for constructing the RS turns out to be far more time-consuming than the simulation itself, although this can only happen when the problem dimension is very high. When the numerical noise increases to a rather high level, RK should be considered over DIRECT irrespective of the problem dimension given its capability of filtering out the noise and hence requiring possibly fewer function evaluations to locate the global optimum or optima.

When applied to solve the complex TLP, RK turns out to be the best-performing method amongst the three alternatives followed by SPSA and then DIRECT. Given the same amount of computational budget (i.e., the same number of function evaluations), RK can improve the optimal objective function value at a much faster rate than either DIRECT or SPSA and achieve the best solution quality, thereby being the most computationally efficient SO method for such a high-dimensional problem. Therefore, if the optimization problem features a high-dimensional decision vector, a complex objective, and/or a set of complex constraints, RK is the preferred method over SPSA and DIRECT. In general, RK seems to have a wider applicability amongst the four SO methods considered. It can be applied to solve both simple and complex SO problems and can handle both noisy and noiseless objective functions. Perhaps this explains why RK has been advocated quite frequently in recent SO studies.

6.6. Chapter Remarks

This chapter focuses on comparing the performance of four SO methods on two benchmark TLPs, including the PI controller method, RK, SPSA, and DIRECT. These methods are considered as computationally efficient representatives amongst the big family of SO methods. Our comparative results suggest that we use the PI controller method to solve a simple problem due to its much faster convergence, and that RK is the preferred

method for solving a complex problem given its capabilities of filtering out the numerical noise arising from computer simulations and of capturing the overall distribution of the optimal solutions.

CHAPTER 7. NETWORK PARTITIONING FOR TOLL AREA IDENTIFICATION

This chapter provides a numerical study on the network partitioning approach for solving the TAP corresponding to Section 3.5. After presenting the network partitioning framework in Section 7.1 as the solution algorithm, we show in Section 7.2 the numerical results of static partitioning, and go deeper in Sections 7.3 and 7.4 to investigate dynamic partitioning and the effect of missing data, respectively. Section 7.5 concludes the chapter. The work of this chapter is currently under revision:

- Gu, Z., Saberi, M., 2019. A bi-partitioning approach to congestion pattern recognition and toll area identification. *Transp. Res. Part C*, under revision.

To facilitate the presentation, the variables used in this chapter are first summarized in Table 7.1.

Table 7.1 Variables used in CHAPTER 7

Notation	Interpretation
$\mathbf{W}$	Composite similarity matrix
$\mathbf{W}_{\mathrm{K}}/\mathbf{W}_{\mathrm{D}}$	Density/distance similarity matrix
σ, γ	Level of accuracy
θ	Weight coefficient
$\bar{S}_{\mathrm{K}}^{m}/\bar{S}_{\mathrm{D}}^{m}$	Average density/distance similarity measure of the PZ for the m-th θ
δ^{m}	Percentage of improvement in the overall similarity for the m-th θ
$E_{\mathrm{Y}}/E_{\mathrm{N}}$	Set of links with and without density data
$\bar{S}_{\mathrm{K}}/\bar{S}_{\mathrm{D}}$	Average density/distance similarity measure of the PZ
$S_{\mathrm{K}}^{i}/S_{\mathrm{D}}^{i}$	Density/Distance similarity measure of link i
$\tilde{S}_{\mathrm{K}}/\tilde{S}_{\mathrm{D}}$	Density/distance threshold
$p_{\mathrm{K}}/p_{\mathrm{D}}$	Density/distance scaling parameter
P	Penetration rate

7.1. Network Partitioning Framework

The network partitioning framework is illustrated in Figure 7.1 followed by detailed algorithmic steps.

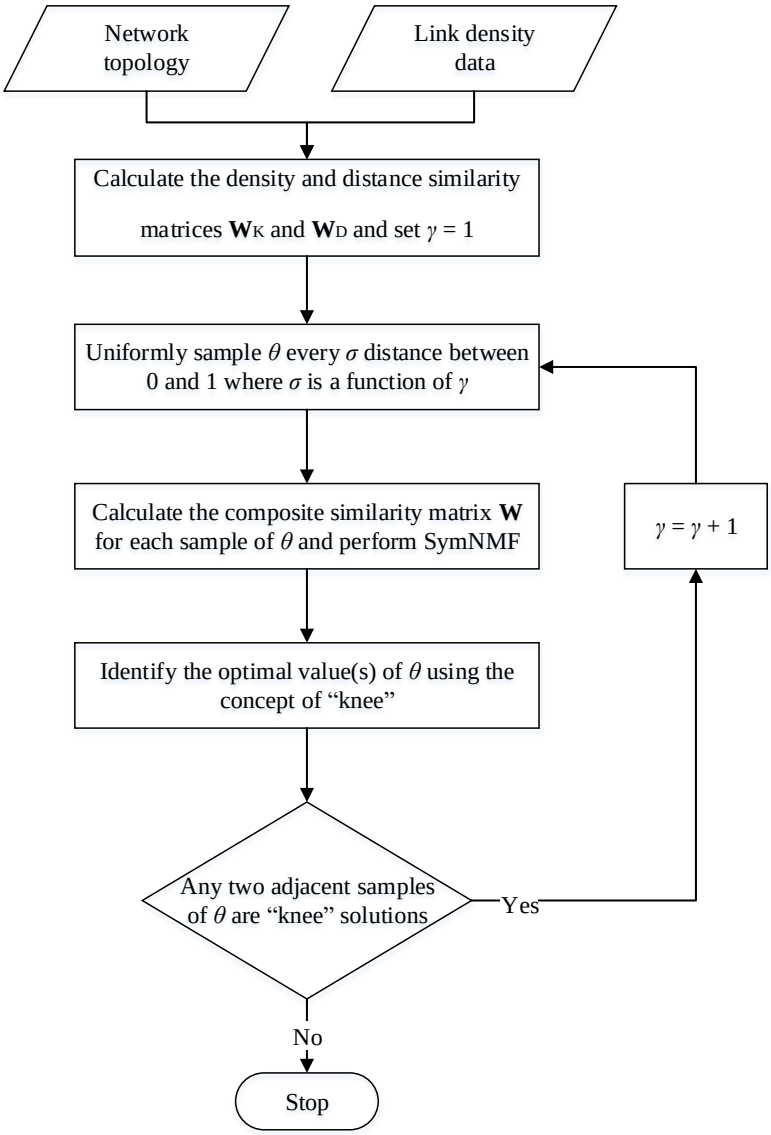


Figure 7.1 Flowchart representation of the proposed solution framework

Step 1. *Initialization.* Given the network topology and link density data, we calculate the density and distance similarity matrices, $\mathbf{W}_\mathrm{K}$ and $\mathbf{W}_\mathrm{D}$. n is the number of links in the network and $k = 2$ is the number of clusters.

Step 2. *Sampling.* Let $\sigma = 0.1^{\gamma}$ where γ represents the level of accuracy initially set at 1. We uniformly sample θ every σ distance in $[0,1]$ resulting in σ^{-1} intervals and $\sigma^{-1} + 1$ samples. For example, $\gamma = 1$ is equivalent to $\sigma = 0.1$ resulting in 10 intervals and 11 samples.

Step 3. *Network partitioning.* Given $\mathbf{W}_{\mathrm{K}}$ and $\mathbf{W}_{\mathrm{D}}$, we calculate the composite similarity matrix, $\mathbf{W}$, for each sampled θ and apply SymNMF to obtain the clustering assignment.

Step 4. *Solution identification.* Let $\bar{S}_{\mathrm{K}}^{m}$ and $\bar{S}_{\mathrm{D}}^{m}$ denote respectively the average values of S_{K}^{i} and S_{D}^{i} over all links in the partitioned PZ. $\theta^{m} = \sigma(m-1)$ where $m \in \{1, \dots, \sigma^{-1}+1\}$. We define an indicator, δ^{m}, corresponding to θ^{m} to show the percentage of improvement in the overall similarity between links in the partitioned PZ. Based on the concept of "knee", any θ with a large δ^{m} is considered as a significant solution from the Pareto front.

$$\delta^{m} = \left(\frac{\bar{S}_{\mathrm{K}}^{m} - \bar{S}_{\mathrm{K}}^{m-1}}{\bar{S}_{\mathrm{K}}^{m-1}} + \frac{\bar{S}_{\mathrm{D}}^{m} - \bar{S}_{\mathrm{D}}^{m-1}}{\bar{S}_{\mathrm{D}}^{m-1}} \right) \times 100\% \tag{7.1}$$

Step 5. *Stop test.* If any two adjacent θ's exhibit a large δ^{m} simultaneously, we cannot locate the significant solutions because the interval is not small enough to guarantee an unchanged clustering assignment. There may be additional significant solutions in between as well. We therefore increase γ by 1 to reduce σ, and go back to Step 2. We terminate the algorithm until any two adjacent θ's do not exhibit a large δ^{m} simultaneously.

The extended solution framework is illustrated in Figure 7.2.

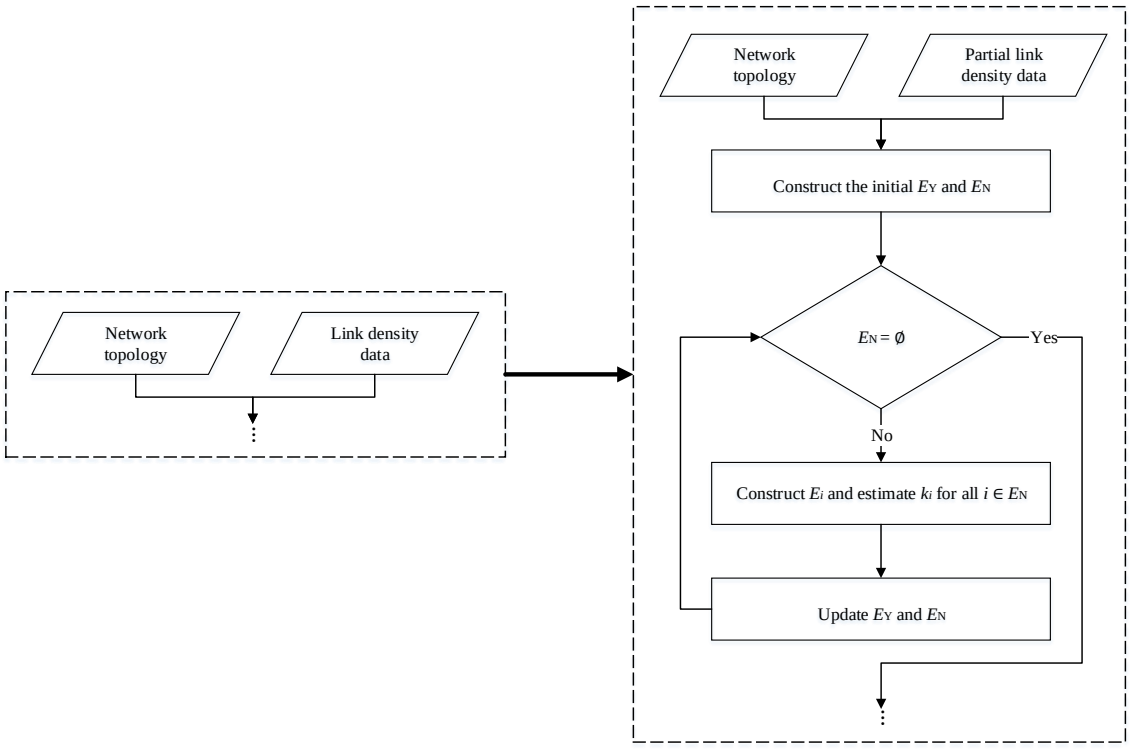
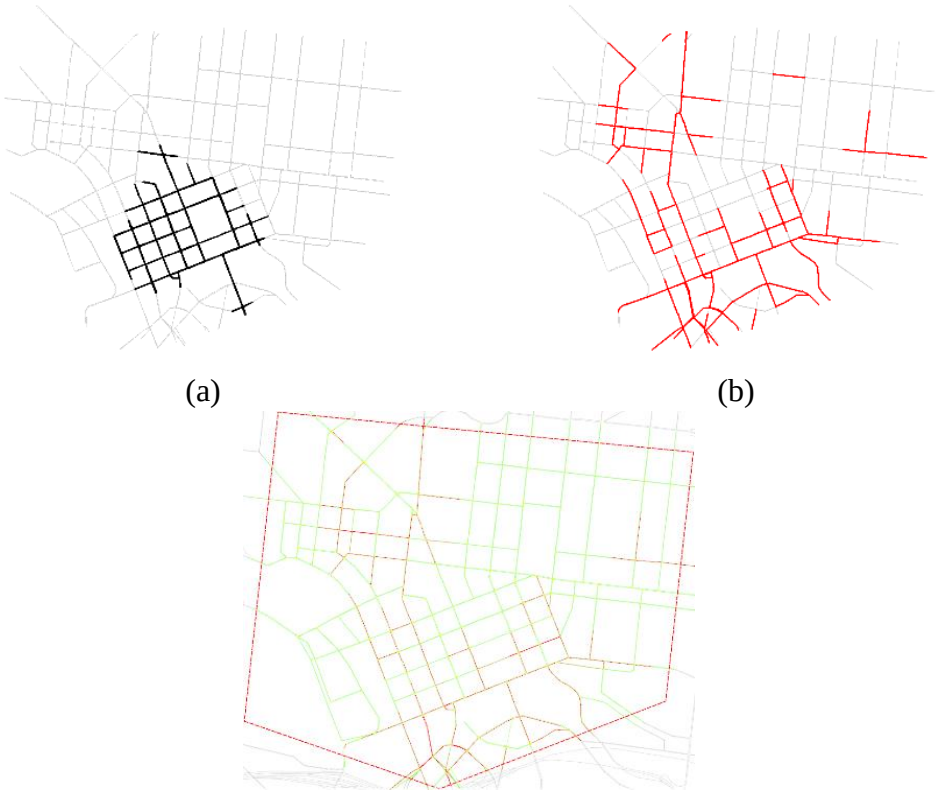


Figure 7.2 Extending Figure 7.1 to further consider missing data

7.2. Static Partitioning

When the weight coefficient $\theta = 1$, the composite similarity matrix, $\mathbf{W}$, is simply the density similarity matrix, $\mathbf{W}_{\mathrm{K}}$, suggesting that the network is partitioned based on the density similarity measure only. When $\theta = 0$, $\mathbf{W} = \mathbf{W}_{\mathrm{D}}$ and the network is partitioned based on the distance similarity measure only. Since $\theta \in [0,1]$, the lower and upper bounds naturally provide two extreme scenarios. As shown in Figure 7.3(a), when SymNMF is applied with $\theta = 0$ to consider the distance similarity measure only, the extracted cluster is highly compact and the included links are well connected, which confirms Proposition 3.2. By defining a distance threshold, we assume a spatial coverage of the initial PZ which is to be expanded or contracted for finding the optimum. We can vary the distance threshold to create different spatial coverages of the optimal PZ. Figure 7.3(b) shows the extracted cluster when SymNMF is applied with $\theta = 1$ to consider the density

similarity measure only, which can be compared with the simulation result in Figure 7.3(c). 229 out of 945 links are extracted into the cluster by SymNMF as congested links. Compared with the simulation result, there are 12 more links with densities slightly lower than the predefined density threshold. SymNMF considers these links as also being congested because their densities are not significantly lower than the threshold. Overall, SymNMF performs quite well in capturing the spatial congestion pattern in the network.



(a) (b)

(c)

Figure 7.3 (a) Network partitioning result when $\theta = 0$, (b) network partitioning result when $\theta = 1$, and (c) simulation result: red (green) links have densities higher (lower) than 50 vpkmpl

We proceed to find the optimal PZ. As shown in Figure 7.4(a), during the first iteration, $\sigma = 0.1$ results in 10 intervals and 11 samples. We apply SymNMF for each sampled θ and calculate the overall similarity improvement. The blue polyline implies that the termination criterion is not met as there is at least one pair of significant solutions

that are adjacent to each other. The algorithm therefore continues to the second iteration with $\sigma = 0.01$ which results in 100 intervals and 101 samples. Now, we can easily locate the significant solutions by referring to the multiple peaks of the orange polyline. Since the clustering assignment does not change or changes little between any two adjacent significant solutions, the algorithm terminates and finds six significant solutions from the Pareto front, see Figure 7.4(b). As shown in Figure 7.4(c), each significant solution results in a different pair of $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$. Here, we use $\bar{S}_{\mathrm{D}} = 0.75$ as a threshold above which the significant solution with the smallest $\bar{S}_{\mathrm{D}}$ is considered optimal. Recall that "optimal" only refers to a sensible trade-off between the two conflicting objectives. The optimal PZ represented by the shaded area in Figure 7.4(d) can be considered and used for further pricing control and optimization. As with Figure 7.3(c), the spatial congestion pattern distributes mainly along the north-south direction. There are three bottleneck corridors or sequences of congested links connecting the city center – one from the north and two from the south.

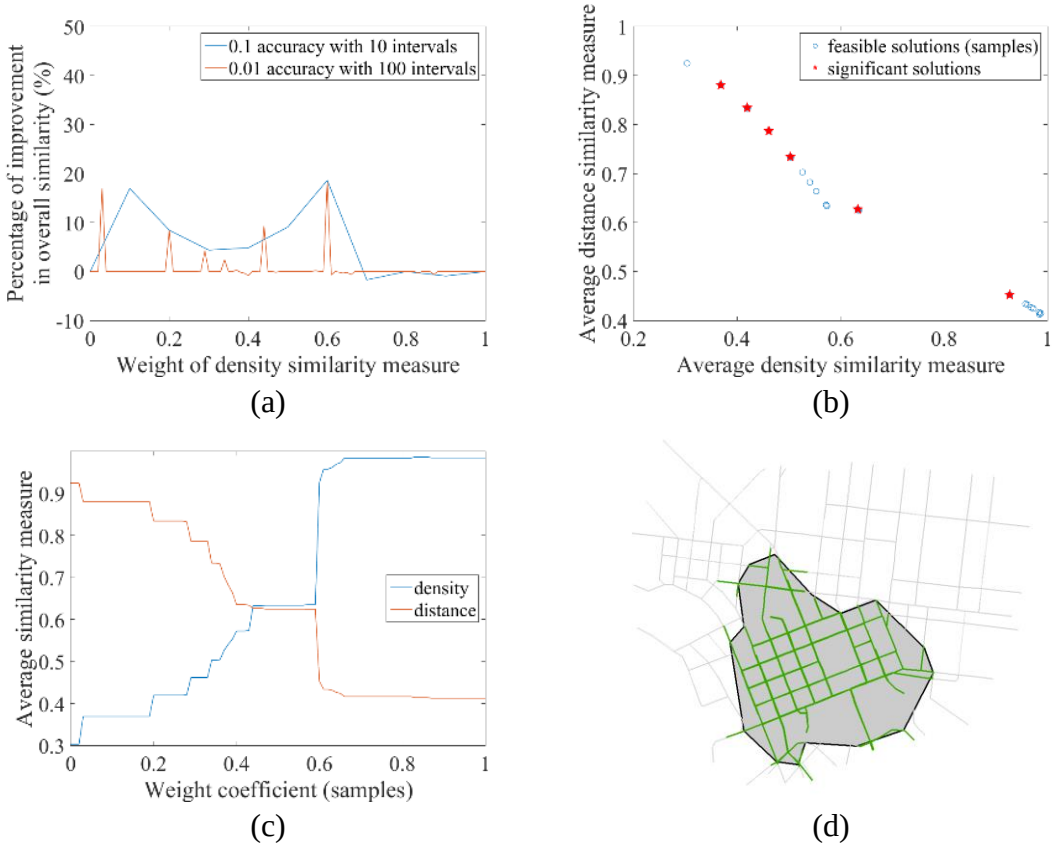


Figure 7.4 (a) Percentage of improvement in the overall similarity for each sampled θ, (b) six significant solutions from the Pareto front, (c) conflicting relationship between $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$, and (d) optimal PZ

7.2.1. Sensitivity Analysis

In this sub-section, we perform a few sensitivity analyses on different parameters to examine their effects on the network partitioning results.

Scaling Parameters

The scaling parameters, p_{K} and p_{D}, are used to calculate the density and distance similarity measures, S_{K}^{i} and S_{D}^{i}, respectively. We previously set $p_{\mathrm{K}} = p_{\mathrm{D}} = 5$ to impose a rather high penalty on dissimilarity. For example, if link density is 40 vpkmpl, $S_{\mathrm{K}}^{i} = \left(\frac{40}{50}\right)^{5} \approx 0.3$ which implies low similarity, although the difference of 10 vpkmpl seems not that significant. We therefore test three other combinations of p_{K} and p_{D}: (i) $p_{\mathrm{K}} = 3$

and $p_{\mathrm{D}} = 5$, (ii) $p_{\mathrm{K}} = p_{\mathrm{D}} = 3$, and (iii) $p_{\mathrm{K}} = 5$ and $p_{\mathrm{D}} = 3$. When p_{K} (p_{D}) is reduced, less penalty is imposed on the density (distance) dissimilarity. As shown in Figure 7.5, when p_{D} remains unchanged, different p_{K}'s result in similar significant solutions and hence, there is no major difference between the extracted clusters. When p_{D} reduces and p_{K} remains unchanged, the optimal PZ becomes slightly expanded, as expected, because a smaller p_{D} equates to a larger S_{D}^{i} that effectively increases the spatial coverage of the PZ. Overall, the network partitioning approach is a bit more sensitive to p_{D} than to p_{K}.

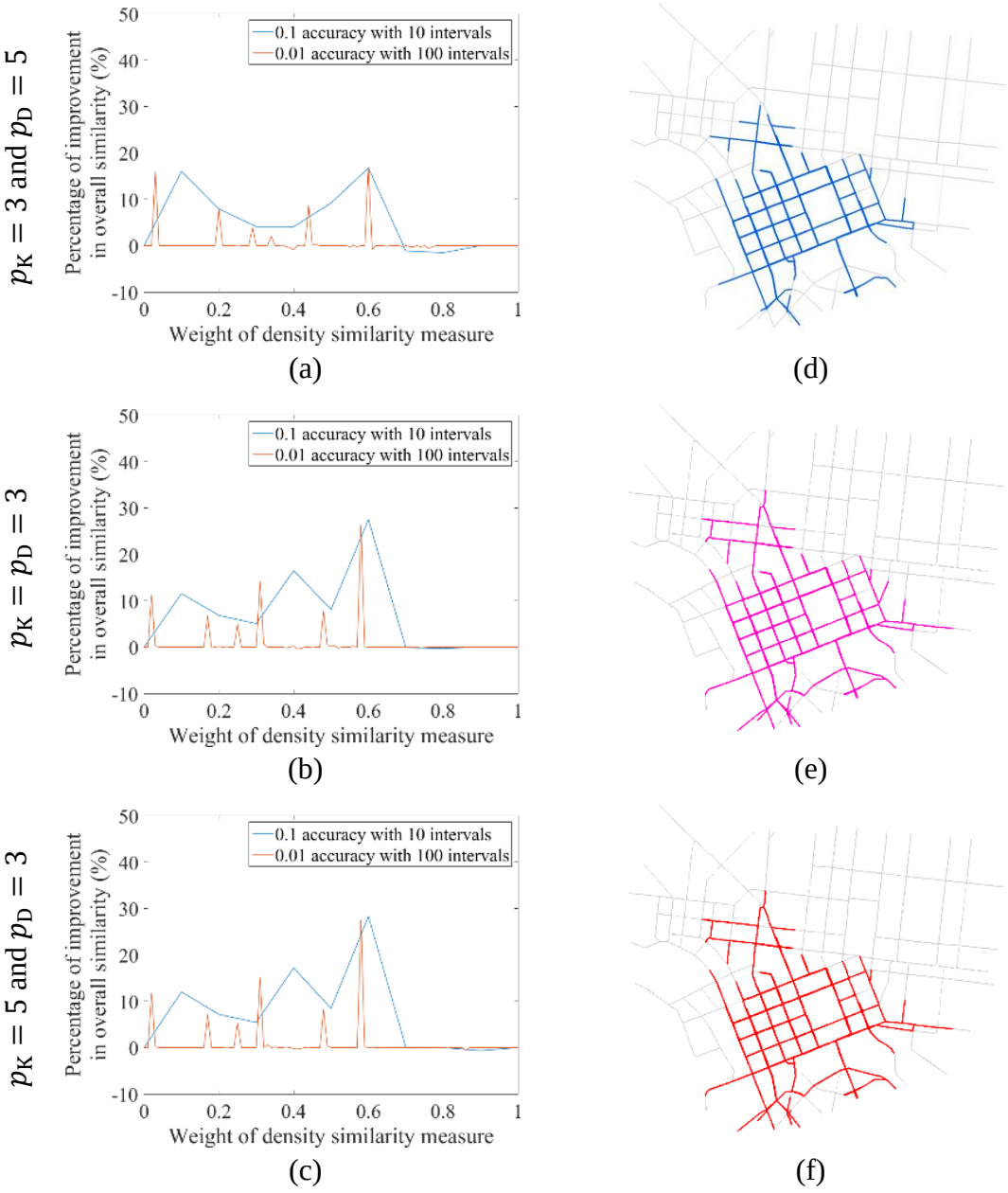


Figure 7.5 Sensitivity analysis on p_K and p_D: (a-c) overall similarity improvement for each sampled θ, and (d-f) optimal PZs

Location of the Source Link

The location of the source link is used to build the shortest path tree for calculating S_D^i. The only selection requirement is that the link should be located around the center of the congested sub-network.

Figure 7.6 tests two other source links highlighted by the thick lines. A comparison between Figure 7.6(c) and (d) and Figure 7.4(d) reveals some spatial differences of the optimal PZs as we move the source link to different locations in the network. This is because the calculation of S_{D}^{i} is largely dependent on the location of the source link which essentially determines the spatial coverage of the initial PZ and hence of the optimal PZ. Nevertheless, this type of sensitivity can be indirectly considered through varying the distance threshold, $\tilde{\mathrm{S}}_{\mathrm{D}}$, to create different spatial coverages.

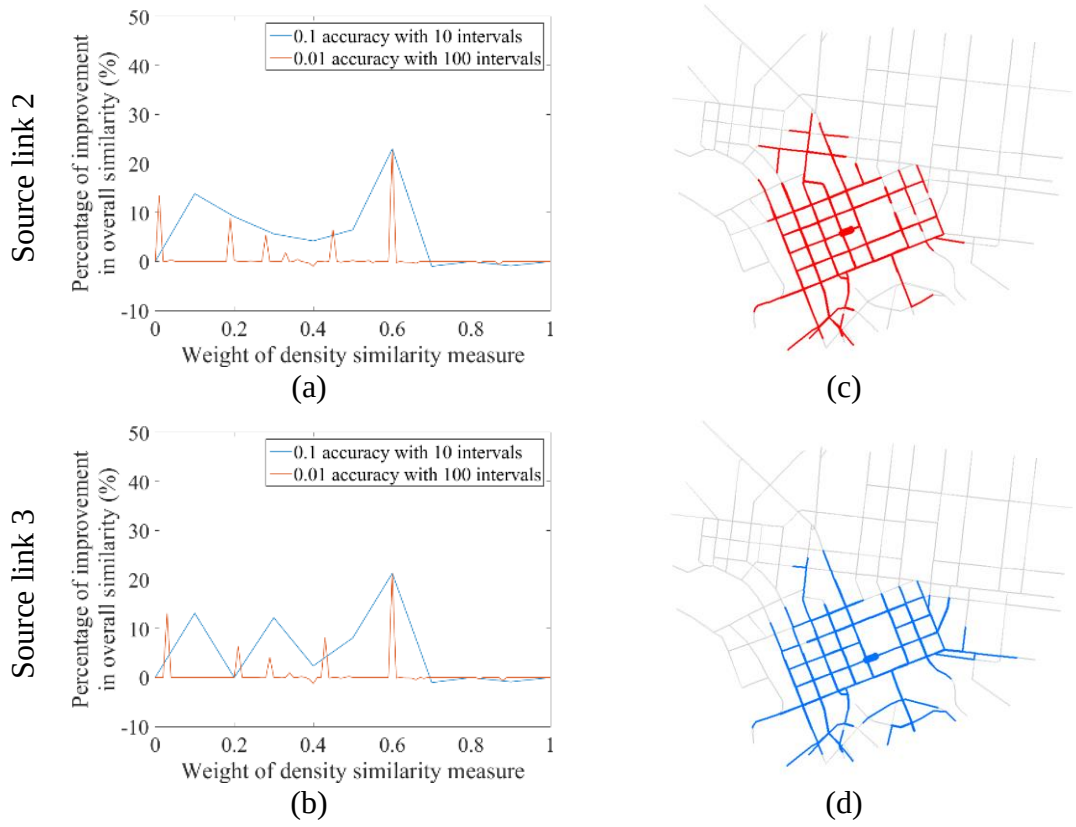


Figure 7.6 Sensitivity analysis on the location of the source link: (a) and (b) overall similarity improvement for each sampled θ, and (c) and (d) optimal PZs where the source links are highlighted by the thick lines

Density and Distance Thresholds

The density and distance thresholds, $\tilde{S}_{\mathrm{K}}$ and $\tilde{S}_{\mathrm{D}}$, are used to calculate S_{K}^{i} and S_{D}^{i}, respectively. While more (fewer) links are considered as being congested when we

decrease (increase) $\tilde{S}_K$, the overall spatial congestion pattern in the network remains similar and hence, the optimal PZs resulting from different $\tilde{S}_K$'s are not significantly different, see Figure 7.7.

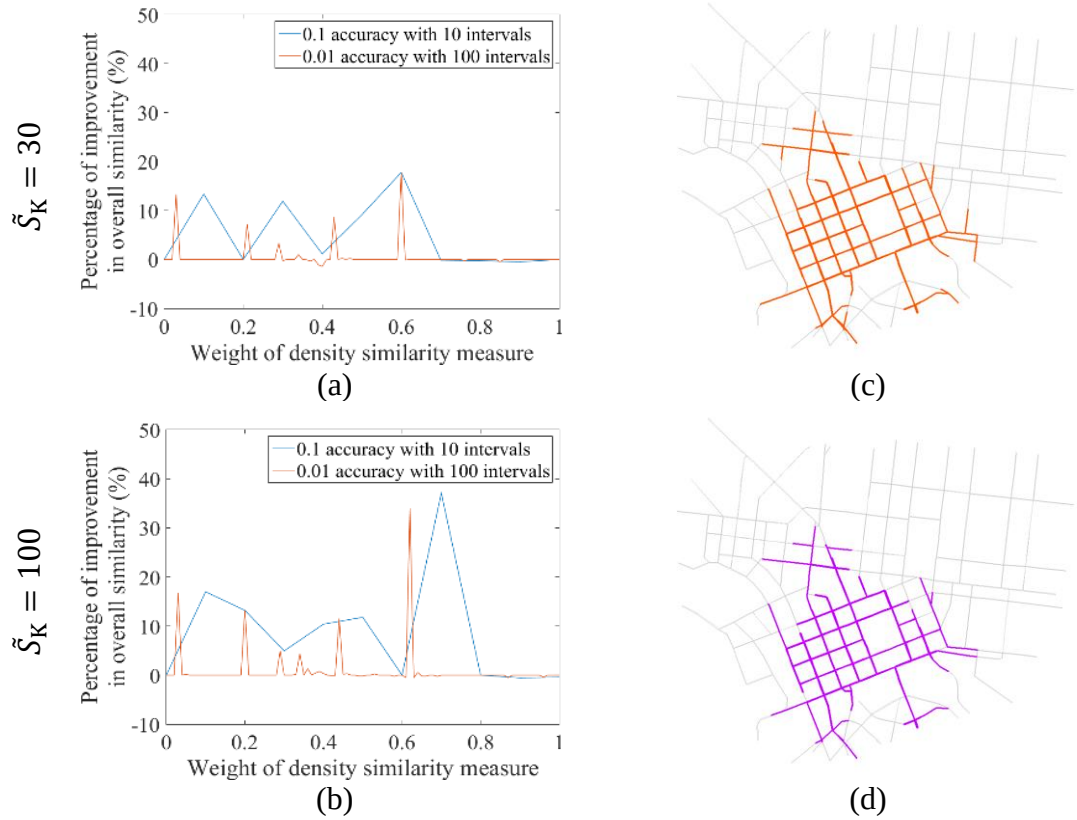


Figure 7.7 Sensitivity analysis on $\tilde{S}_K$: (a) and (b) overall similarity improvement for each sampled θ, and (c) and (d) optimal PZs

It is obvious that changing $\tilde{S}_D$ results in different spatial coverages of the initial PZ as well as of the optimal PZ. This parameter therefore has a more profound impact on the network partitioning result. We emphasize that a variable $\tilde{S}_D$ provides more flexibility for applying the network partitioning approach – instead of focusing on a specific $\tilde{S}_D$, one can choose and investigate multiple $\tilde{S}_D$'s to consider different spatial coverages of the PZ. A gradually increasing $\tilde{S}_D$ helps capture the spatial evolution of the optimal PZ, which, to a large extent, reflects how congestion propagates in the network. As shown in Figure

7.8(a-f), the optimal PZ varies in size and shape as $\tilde{S}_{\mathrm{D}}$ changes from 5 to 10. The captured spatial congestion pattern gradually evolves and increasingly resembles Figure 7.3(c). We observe from Figure 7.3(g) that $\bar{S}_{\mathrm{K}}$ remains relatively stable as $\tilde{S}_{\mathrm{D}}$ increases. $\bar{S}_{\mathrm{D}}$, however, exhibits a slightly increasing trend due to the decreasing difference between the spatial coverages of the initial PZ and the original network – more links having a large S_{D}^{i} naturally results in a larger $\bar{S}_{\mathrm{D}}$. If the initial PZ covers a relatively small area of the original network, this natural increase may become negligible. Although a changing $\tilde{S}_{\mathrm{D}}$ results in different optimal PZs, the performance of the network partitioning approach is robust given similar $\bar{S}_{\mathrm{K}}$'s and $\bar{S}_{\mathrm{D}}$'s. Note that a variable $\tilde{S}_{\mathrm{D}}$ enables designing a double- or multi-layered PZ and we provide an example in Figure 7.8(d). This helps implement a hierarchical pricing scheme – the highest price can be imposed on the innermost area which gradually reduces towards the outmost area.

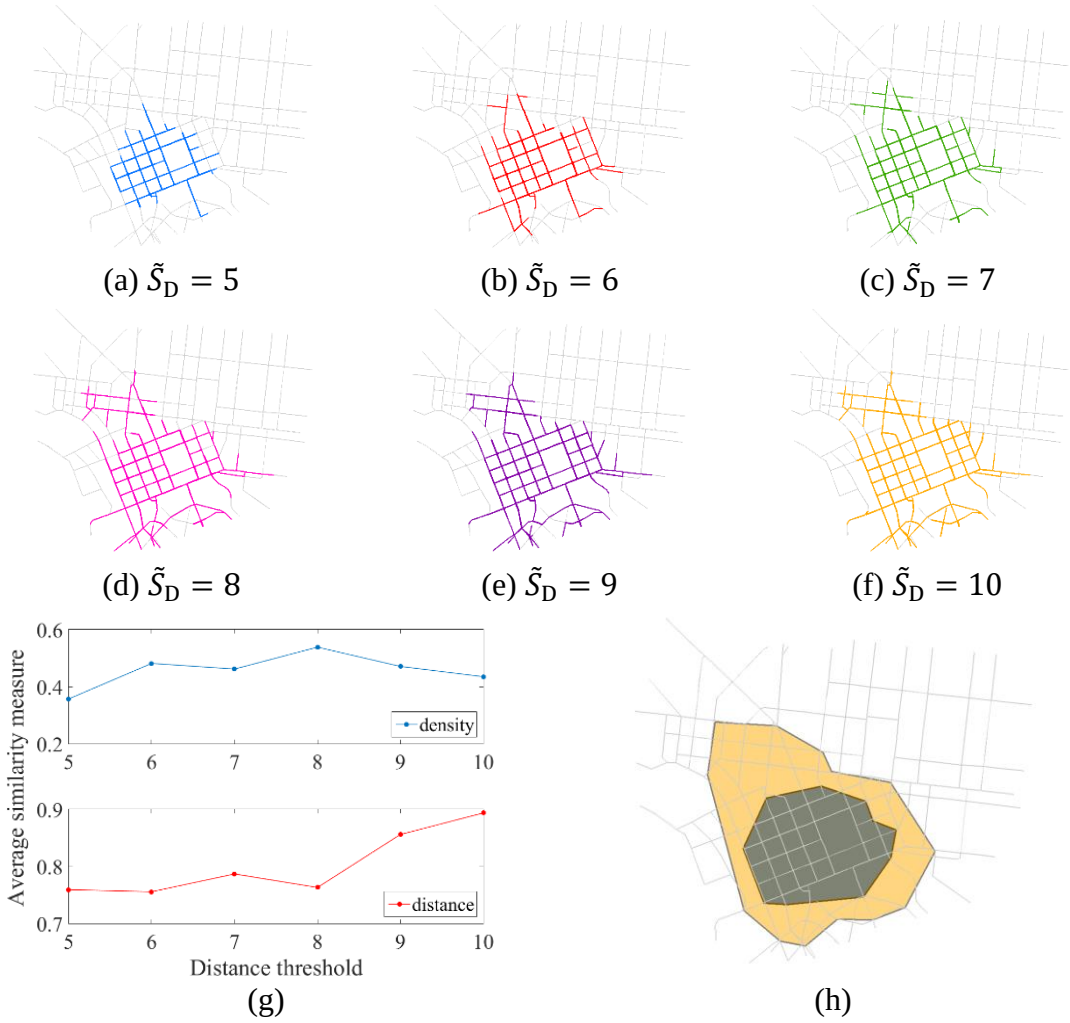


Figure 7.8 Sensitivity analysis on $\tilde{S}_D$: (a-f) optimal PZs corresponding to different $\tilde{S}_D$'s, (g) variations of $\bar{S}_K$ and $\bar{S}_D$ as $\tilde{S}_D$ changes, and (h) double-layered optimal PZ resulting from $\tilde{S}_D = 5, 10$

7.3. Dynamic Partitioning

Traffic is inherently dynamic. Link densities usually vary with time and hence, the spatial congestion pattern in the network typically changes from time to time. To consider this time-dependency nature, we further apply the network partitioning approach using link density data from different time intervals. As a result, the optimal PZ is dynamically changing with time rather than being time-invariant, which reflects the spatio-temporal propagation of congestion in the network.

Figure 7.9 shows the optimal PZs for three different time intervals. During the 7:30-7:45 AM interval, the network is almost free from congestion. There are only a few isolated congested links in the network and hence, the optimal PZ does not exhibit a well-defined shape, as expected. The implication is twofold:

- Congestion is currently at the embryonic stage without propagating throughout the network.
- The scattered distribution of congestion across the network does not justify a well-defined PZ for an area charge.

As congestion gradually builds up over time towards the 8:30-8:45 and 9:30-9:45 AM intervals, the isolated congested links become increasingly connected to form a well-defined PZ, suggesting that dynamic partitioning has the potential to inform when to activate an area charge.

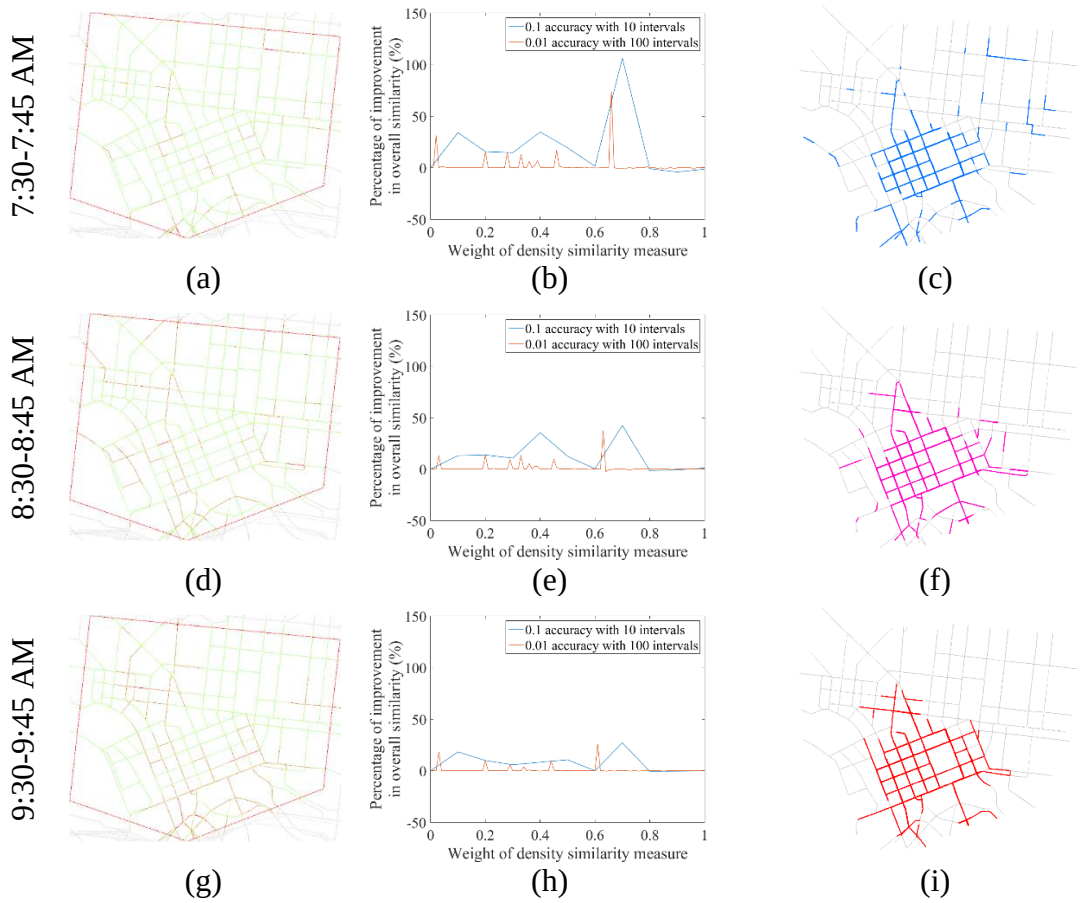


Figure 7.9 Network partitioning using link density data from different time intervals: (a-c) 7:30-7:45 AM, (d-f) 8:30-8:45 AM, (g-i) 9:30-9:45 AM

From a purely practical perspective, a time-varying PZ is not advisable at least for now as part of a congestion pricing policy. Since human-driven vehicles still account for the largest proportion of vehicles on roads, changing the PZ with time largely increases the complexity of the pricing system and hence makes it difficult for travelers to understand, adapt, and eventually accept. See our discussion in Sub-section 2.1.2. Nevertheless, we believe that a time-varying PZ is a promising policy option when vehicle automation and communications become more and more popularized to take up a much larger market share. These technologies allow vehicles to easily adapt to the ongoing traffic conditions in the network without the need for any human intervention.

7.4. Considering Missing Data

To investigate the performance of the network partitioning approach considering missing data, we design a few hypothetical scenarios where different penetration rates, P's, are assumed. Specifically, we assume that 30%, 50%, 70%, and 90% of links in the network are equipped with detectors to provide density data, respectively. Under each penetration rate scenario, we randomly generate three different sets of links with data and repeatedly apply the network partitioning approach.

As shown in Figure 7.10(a-d), the optimal PZs under different penetration rate scenarios are quite similar and resemble the one obtained assuming perfect information of link densities. A further look into Figure 7.10(e) reveals an almost linearly increasing trend in $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$ when P increases from 30% to 100%. This is sensible as the optimal network partitioning result is expected to improve with greater availability of link density data. Nevertheless, Table 7.2 shows that $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$ reduce only slightly even when P drops from 100% to 30%. Such a small difference suggest that the approach performs equally well even with a low penetration rate, a feature that is promising for practical applications that often violate the perfect information assumption. By using only a few detectors installed across the network, we can apply the network partitioning approach to achieve an equally good result. Note that a few studies (Ortigosa et al., 2013; Zockaie et al., 2018) have shown that the incomplete NFD from using only part of the network data can provide a good estimate of the actual complete NFD.

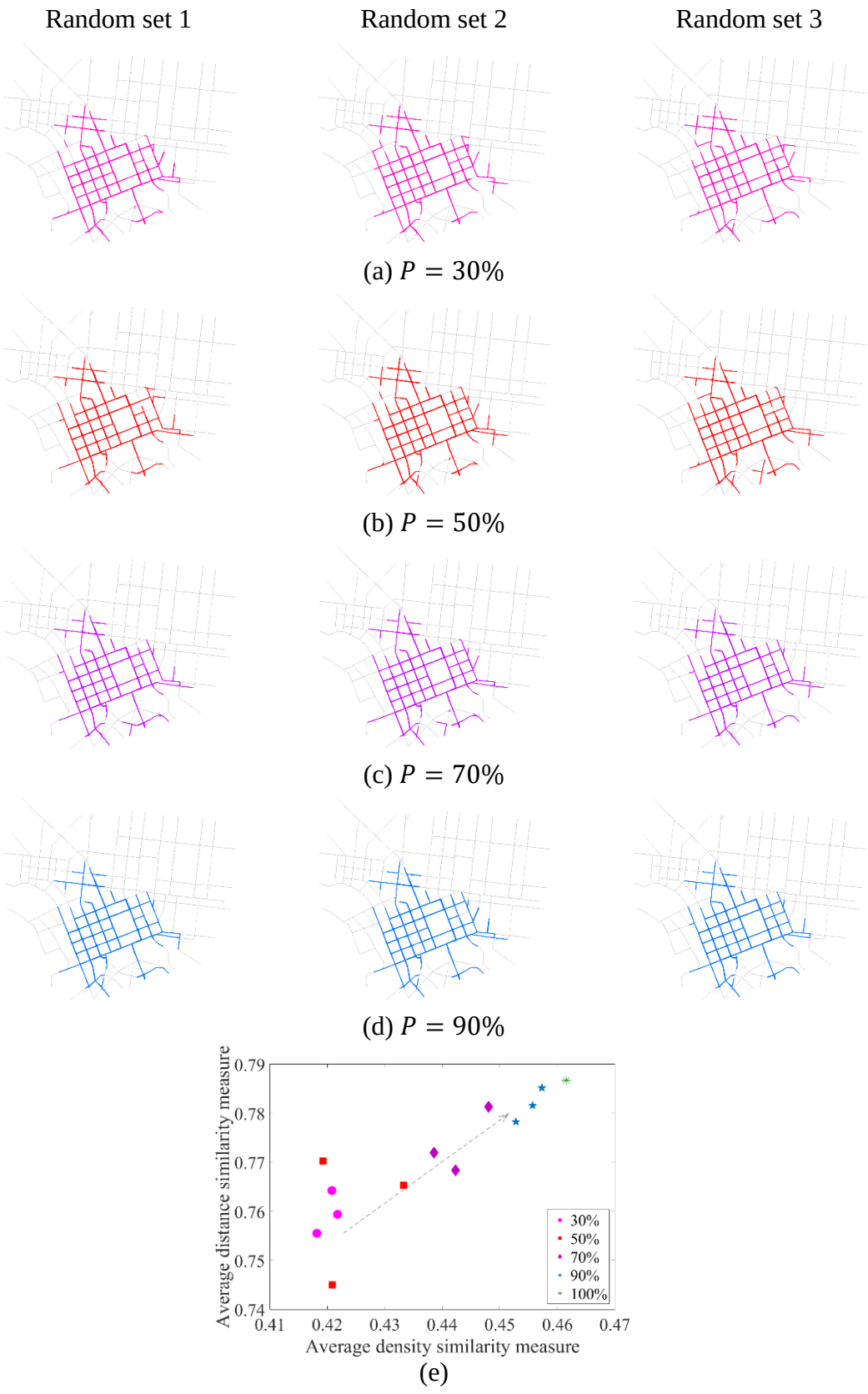


Figure 7.10 Network partitioning considering missing data under different penetration rate scenarios: (a-d) $P = 30\%, 50\%, 70\%, 90\%$, and (e) variations of $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$ as P changes

Table 7.2 Relative changes in $\bar{S}_{\mathrm{K}}$ and $\bar{S}_{\mathrm{D}}$ under different penetration rate scenarios compared with $P = 100\%$

Random set	$P = 30\%$		$P = 50\%$		$P = 70\%$		$P = 90\%$	
	$\bar{S}_{\mathrm{K}}$	$\bar{S}_{\mathrm{D}}$	$\bar{S}_{\mathrm{K}}$	$\bar{S}_{\mathrm{D}}$	$\bar{S}_{\mathrm{K}}$	$\bar{S}_{\mathrm{D}}$	$\bar{S}_{\mathrm{K}}$	$\bar{S}_{\mathrm{D}}$
1	-8.6%	-3.5%	-9.2%	-2.1%	-5.0%	-1.9%	-0.9%	-0.2%
2	-9.4%	-4.0%	-8.8%	-5.3%	-4.2%	-2.3%	-1.9%	-1.1%
3	-8.8%	-2.9%	-6.1%	-2.7%	-2.9%	-0.7%	-1.3%	-0.7%
Average	-8.9%	-3.4%	-8.0%	-3.4%	-4.0%	-1.6%	-1.4%	-0.6%

7.5. Chapter Remarks

This chapter presents detailed numerical results for the network partitioning approach tailored for solving the TAP. Three key conclusions are summarized as follows:

- Using different distance thresholds results in different spatial coverages of the optimal PZ. This flexibility enables designing a double- or multi-layered PZ for hierarchical pricing applications.
- The optimal PZ can vary with time provided that data from different time intervals are available. This time-dependency nature helps inform when to activate an area charge.
- The approach can handle different levels of missing data and provide robust network partitioning results.

CHAPTER 8. CONCLUSION

8.1. Summary

This thesis advances the study, design, and implementation of two-region urban pricing systems as a promising TDM policy to reduce the increasing level of traffic congestion in city centers. The proposed work extends the current congestion pricing theory by proposing and integrating advanced (i.e. more efficient and equitable) pricing regimes with the NFD and exploring and comparing computationally efficient SO methods in a simulation-based DTA environment. Results of this work not only help in developing effective pricing systems to mitigate urban traffic congestion, but also provide competitive solutions to other types of NDPs.

We investigate three types of pricing regimes in this thesis, namely the distance only toll, the JDTT, and the JDDT. The latter two are considered by the authors as advanced pricing regimes building upon the distance only toll which represents the state of the practice. Through computer simulations, we demonstrate the capability of the PI controller method as our first proposed approach in solving a simple TLP – the network is successfully pricing controlled to achieve its optimal state defined by the critical network density of the NFD. We also demonstrate the superiority of the JDTT and the JDDT over the distance only toll in that the latter naturally drives travelers into the shortest paths within the PZ resulting in a more uneven distribution of congestion. While congestion certainly reduces to the desired level, the increased heterogeneity of congestion distribution leads to a larger hysteresis loop in the NFD and hence lower network flows especially during network recovery. The JDTT and the JDDT can overcome this limitation of the distance only toll by taking into account, respectively, a time and a delay toll component.

To explicitly model and minimize the heterogeneity of congestion distribution rather than simply renovating the pricing mechanism, the PI controller method is no longer applicable. Therefore, to solve such a complex TLP, we refer to RSM, or, more concretely, RK, as our second proposed approach. Results, as expected, show that higher network flows are achieved once we consider reducing the heterogeneity of congestion distribution as part of the optimization problem. The method turns out to be very effective and efficient even though the problem to be solved involves a high-dimensional decision vector and a set of complex constraints.

Since part of this thesis is about SO methods, we perform a comprehensive comparison between the performance of the PI controller method and RK as well as of two other computationally efficient SO methods, namely SPSA and DIRECT, on a same benchmark TLP. While results show that all the methods work well to solve the benchmark TLP, they clearly have their own pros and cons when compared with each other. In general, we recommend applying the PI controller method to solve a simple problem due to its much faster convergence and RK to solve a complex problem given its capabilities of filtering out the numerical noise arising from computer simulations and of capturing the overall distribution of the optimal solutions.

As part of the overall TDP, we also study the TAP which is another important aspect of pricing system design but turns out to be much less researched in the literature compared with the TLP. In our work, we have employed a data-driven perspective and proposed a network partitioning approach to optimizing the PZ for a common two-region urban road network. The flexibility of the method enables designing a double- or multi-layered PZ for hierarchical pricing applications as well as time-dependent partitioning that helps inform when to activate an area charge. Even with different levels of missing data, the method is shown to work well producing consistent and robust results.

8.2. Limitations and Future Work

There are four main limitations of this work which can be extended as future research directions:

- Incorporating a demand model with the proposed work to jointly consider the effects of pricing on the mode and departure time choices. When applying different SO methods to solve a TLP, we assume that the demand is fixed without considering the effects of pricing on the mode and departure time choices. This assumption does not affect the performance of the proposed pricing optimization frameworks, but, to make the results more realistic, a demand model to be integrated with the simulation-based DTA model can be a research priority, although developing such a demand model per se is a non-trivial task.
- *Modeling and optimizing a coordinated multi-area pricing system.* The proposed work focuses on modeling and optimizing a two-region urban road network. This requirement or assumption applies to most cities in the world having a single congested CBD, but not those having multiple disjoint CBDs. Therefore, further research effort is needed to develop methods for modeling and optimizing a coordinated multi-area pricing system.
- *Developing SO methods for simultaneously solving the TLP and the TAP*. In this thesis, we consider the TLP and the TAP as two independent problems and hence solve them through different methods in a respective or sequential manner. Developing SO methods that can simultaneously solve the TLP and the TAP is accordingly a promising yet difficult research task. The biggest difficulty lies in how one can generate different cordon

samples and express the overall decision vector in an effective and efficient manner.

- *Investigating real-time network-wide pricing.* This thesis is all about pricing system design for planning purposes. Therefore, a future research direction, not necessarily being the limitation of the proposed work, is to investigate real-time network-wide pricing that is more suited for short-term or special-event traffic control and management.

REFERENCES

Amaran, S., Sahinidis, N.V., Sharda, B., Bury, S.J., 2016. Simulation optimization: a review of algorithms and applications. *Ann. Oper. Res.* 240(1), 351-380.

Antoniou, C., Koutsopoulos, H.N., Yannis, G., 2013. Dynamic data-driven local traffic state estimation and prediction. *Transp. Res. Part C* 34, 89-107.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M., 2013. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons.

Buisson, C., Ladier, C., 2009. Exploring the Impact of Homogeneity of Traffic Measurements on the Existence of Macroscopic Fundamental Diagrams. *Transp. Res. Rec.: J. Transp. Res. Board*(2124), 127-136.

Bureau of Infrastructure, Transport and Regional Economics (BITRE), 2015. Traffic and congestion cost trends for Australian capital cities.

Cascetta, E., Nuzzolo, A., Russo, F., Vitetta, A., 1996. A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, Lyon, France, pp. 697-711.

Chaudhari, P., Dharaskar, R., Thakare, V., 2010. Computing the most significant solution from Pareto front obtained in multi-objective evolutionary. *Int. J. Adv. Comput. Sci. Appl.* 1(4), 63-68.

Chen, X., Xiong, C., He, X., Zhu, Z., Zhang, L., 2016. Time-of-day vehicle mileage fees for congestion mitigation and revenue generation: A simulation-based optimization method and its real-world application. *Transp. Res. Part C* 63, 71-95.

Chen, X., Zhang, L., He, X., Xiong, C., Li, Z., 2014. Surrogate-Based Optimization of Expensive-to-Evaluate Objective for Optimal Highway Toll Charges in Transportation Network. *Comput.-Aided Civ. Inf. Eng.* 29(5), 359-381.

Chen, X., Zhang, L., He, X., Xiong, C., Zhu, Z., 2018. Simulation-based pricing optimization for improving network-wide travel time reliability. *Transportmetrica A* 14(1-2), 155-176.

Chow, J.Y.J., Regan, A.C., 2014. A surrogate-based multiobjective metaheuristic and network degradation simulation model for robust toll pricing. *Optim. Eng,* 15(1), 137-165.

Christin, T., Hug, S., Sciarini, P., 2002. Interests and information in referendum voting: An analysis of Swiss voters. *Eur. J. Polit. Res.* 41(6), 759-776.

Chung, B.D., Yao, T., Friesz, T.L., Liu, H., 2012. Dynamic congestion pricing with demand uncertainty: A robust optimization approach. *Transp. Res. Part B* 46(10), 1504-1518.

Daganzo, C.F., 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. *Transp. Res. Part B* 41(1), 49-62.

Daganzo, C.F., Lehe, L.J., 2015. Distance-dependent congestion pricing for downtown zones. *Transp. Res. Part B* 75, 89-99.

De Borger, B., Proost, S., 2012. A political economy model of road pricing. *J. Urban Econ* 71(1), 79-92.

de Palma, A., Kilani, M., Lindsey, R., 2005. Congestion pricing on a road network: A study using the dynamic equilibrium simulator METROPOLIS. *Transp. Res. Part A* 39(7-9), 588-611.

de Palma, A., Lindsey, R., 2011. Traffic congestion pricing methodologies and technologies. *Transp. Res. Part C* 19(6), 1377-1399.

Deng, G., Ferris, M.C., 2007. Extension of the DIRECT optimization algorithm for noisy functions, in: Henderson, S.G., Biller, B., Hsieh, M.H., Shortle, J., Tew, J.D., Barton, R.R. (Eds.), *Proceedings of the 2007 Winter Simulation Conference*, pp. 497-504.

Deng, G., Ferris, M.C., 2009. Variable-Number Sample-Path Optimization. *Math. Program.* 117(1-2), 81-109.

Ding, C., He, X., Simon, H.D., 2005. On the equivalence of nonnegative matrix factorization and spectral clustering, *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 606-610.

Edie, L.C., 1963. Discussion of traffic stream measurements and definitions. *Proceedings of the 2nd International Symposium on the Theory of Traffic Flow*, 139-154.

Ekström, J., Engelson, L., Rydergren, C., 2014. Optimal toll locations and toll levels in congestion pricing schemes: a case study of Stockholm. *Transp. Plan. Technol.* 37(4), 333-353.

Ekström, J., Kristoffersson, I., Quttineh, N.-H., 2016. Surrogate-based optimization of cordon toll levels in congested traffic networks. *J. Adv. Transp.* 50(6), 1008-1033.

Ekström, J., Sumalee, A., Lo, H.K., 2012. Optimizing toll locations and levels using a mixed integer linear approximation approach. *Transp. Res. Part B* 46(7), 834-854.

Farrell, S., Saleh, W., 2005. Road-user charging and the modelling of revenue allocation. *Transp. Policy* 12(5), 431-442.

Forrester, A.I.J., Keane, A.J., Bressloff, N.W., 2006. Design and analysis of "noisy" computer experiments. *AIAA J.* 44(10), 2331-2339.

Forrester, A.I.J., Sóbester, A., Keane, A.J., 2008. *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons, Chichester, UK.

Francke, A., Kaniok, D., 2013. Responses to differentiated road pricing schemes. *Transp. Res. Part A* 48, 25-30.

Gayah, V.V., Daganzo, C.F., 2011. Clockwise hysteresis loops in the Macroscopic Fundamental Diagram: An effect of network instability. *Transp. Res. Part B* 45(4), 643-655.

Geroliminis, N., Daganzo, C.F., 2008. Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transp. Res. Part B* 42(9), 759-770.

Geroliminis, N., Haddad, J., Ramezani, M., 2013. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach. *IEEE Trans. Intell. Transp. Syst.* 14(1), 348-359.

Geroliminis, N., Levinson, D., 2009. Cordon Pricing Consistent with the Physics of Overcrowding, in: Lam, W.H.K., Wong, S.C., Lo, H.K. (Eds.), *Transportation and Traffic Theory 2009: Golden Jubilee*. Springer US, pp. 219-240.

Geroliminis, N., Sun, J., 2011. Hysteresis phenomena of a Macroscopic Fundamental Diagram in freeway networks. *Transp. Res. Part A* 45(9), 966-979.

Godfrey, J.W., 1969. The mechanism of a road network. *Traffic Eng. Control* 11(7), 323-327.

Gu, Z., Liu, Z., Cheng, Q., Saberi, M., 2018. Congestion pricing practices and public acceptance: A review of evidence. *Case Stud. Transp. Policy* 6(1), 94-101.

Gu, Z., Saberi, M., Sarvi, M., Liu, Z., 2016. Calibration of Traffic Flow Fundamental Diagrams for Network Simulation Applications: A Two-Stage Clustering Approach, *Proceedings of the 19th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, pp. 1348-1353.

Gu, Z., Saberi, M., Sarvi, M., Liu, Z., 2018. A big data approach for clustering and calibration of link fundamental diagrams for large-scale network simulation applications. *Transp. Res. Part C* 94, 151-171.

Gu, Z., Shafiei, S., Liu, Z., Saberi, M., 2018. Optimal distance- and time-dependent area-based pricing with the Network Fundamental Diagram. *Transp. Res. Part C* 95, 1-28.

Hau, T.D., 1990. Electronic road pricing: developments in Hong Kong 1983-1989. *J. Transp. Econ. Policy*, 203-214.

He, X., Chen, X., Xiong, C., Zhu, Z., Zhang, L., 2017. Optimal Time-Varying Pricing for Toll Roads Under Multiple Objectives: A Simulation-Based Optimization Approach. *Transp. Sci.* 51(2), 412-426.

He, Z., Xie, S., Zdunek, R., Zhou, G., Cichocki, A., 2011. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Trans. Neural Netw.* 22(12), 2117-2131.

Hensher, D., Li, Z., 2013. Referendum voting in road pricing reform: a review of the evidence. *Transp. Policy* 25, 186-197.

Ji, Y., Geroliminis, N., 2012. On the spatial partitioning of urban transportation networks. *Transp. Res. Part B* 46(10), 1639-1656.

Jones, D.R., Perttunen, C.D., Stuckman, B.E., 1993. Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.* 79(1), 157-181.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* 13(4), 455-492.

Keyvan-Ekbatani, M., Gao, X., Gayah, V.V., Knoop, V.L., 2016. Combination of traffic-responsive and gating control in urban networks: Effective interactions, *Transportation Research Board 95th Annual Meeting*, Washington, DC.

Keyvan-Ekbatani, M., Kouvelas, A., Papamichail, I., Papageorgiou, M., 2012. Exploiting the fundamental diagram of urban networks for feedback-based gating. *Transp. Res. Part B* 46(10), 1393-1403.

Kiefer, J., Wolfowitz, J., 1952. Stochastic estimation of the maximum of a regression function. *Ann. Math. Stat.* 23(3), 462-466.

Knight, F.H., 1924. Some fallacies in the interpretation of social cost. *Q. J. Econ.*, 582-606.

Knoop, V., Hoogendoorn, S., 2013. Empirics of a Generalized Macroscopic Fundamental Diagram for Urban Freeways. *Transp. Res. Rec.: J. Transp. Res. Board*(2391), 133-141.

Kuang, D., Ding, C., Park, H., 2012. Symmetric nonnegative matrix factorization for graph clustering, *Proceedings of the 2012 SIAM International Conference on Data Mining*, pp. 106-117.

Kuang, D., Yun, S., Park, H., 2015. SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering. *J. Glob. Optim.* 62(3), 545-574.

Kulis, B., Basu, S., Dhillon, I., Mooney, R., 2009. Semi-supervised graph clustering: a kernel approach. *Mach. Learn.* 74(1), 1-22.

Lawphongpanich, S., Yin, Y., 2012. Nonlinear pricing on transportation networks. *Transp. Res. Part C* 20(1), 218-235.

Legaspi, J., Douglas, N., 2015. Value of Travel Time Revisited–NSW Experiment, *Proceedings of the 37th Australasian Transport Research Forum (ATRF)*, Sydney, NSW, Australia.

Li, M.Z., 2002. The role of speed–flow relationship in congestion pricing implementation with an application to Singapore. *Transp. Res. Part B* 36(8), 731-754.

Liu, L.N., McDonald, J.F., 1999. Economic efficiency of second-best congestion pricing schemes in urban highway systems. *Transp. Res. Part B* 33(3), 157-188.

Liu, Z., Meng, Q., Wang, S., 2013. Speed-based toll design for cordon-based congestion pricing scheme. *Transp. Res. Part C* 31, 83-98.

Liu, Z., Wang, S., Meng, Q., 2014. Optimal joint distance and time toll for cordon-based congestion pricing. *Transp. Res. Part B* 69, 81-97.

Liu, Z., Wang, S., Zhou, B., Cheng, Q., 2017. Robust optimization of distance-based tolls in a network considering stochastic day to day dynamics. *Transp. Res. Part C* 79, 58-72.

Lopez, C., Leclercq, L., Krishnakumari, P., Chiabaut, N., van Lint, H., 2017. Revealing the day-to-day regularity of urban congestion patterns with 3D speed maps. *Sci. Rep.* 7(1), 14029.

Mahmassani, H., Williams, J.C., Herman, R., 1987. Performance of urban traffic networks, *Proceedings of the 10th International Symposium on Transportation and Traffic Theory*. Elsevier, pp. 1-20.

Mahmassani, H.S., Saberi, M., Zockaie, A., 2013. Urban network gridlock: Theory, characteristics, and dynamics. *Transp. Res. Part C* 36, 480-497.

Mahmassani, H.S., Williams, J.C., Herman, R., 1984. Investigation of network-level traffic flow relationships: some simulation results. *Transportation Research Record* 971, 121-130.

Maryak, J.L., Chin, D.C., 2008. Global Random Optimization by Simultaneous Perturbation Stochastic Approximation. *IEEE Trans. Automat. Contr.* 53(3), 780-783.

May, A.D., 1992. Road pricing: an international perspective. *Transportation* 19(4), 313-333.

May, A.D., Liu, R., Shepherd, S.P., Sumalee, A., 2002. The impact of cordon design on the performance of road pricing schemes. *Transp. Policy* 9(3), 209-220.

May, A.D., Milne, D., Shepherd, S., Sumalee, A., 2002. Specification of optimal cordon pricing locations and charges. *Transp. Res. Rec.: J. Transp. Res. Board*(1812), 60-68.

May, A.D., Milne, D.S., 2000. Effects of alternative road pricing systems on network performance. *Transp. Res. Part A* 34(6), 407-436.

Mazloumian, A., Geroliminis, N., Helbing, D., 2010. The spatial variability of vehicle densities as determinant of urban network capacity. *Philos. Trans. Roy. Soc. A* 368(1928), 4627-4647.

Meng, Q., Liu, Z., Wang, S., 2012. Optimal distance tolls under congestion pricing and continuously distributed value of time. *Transp. Res. Part E* 48(5), 937-957.

Meng, Q., Wang, X., 2008. Sensitivity analysis of logit-based stochastic user equilibrium network flows with entry–exit toll schemes. *Comput.-Aided Civ. Inf. Eng.* 23(2), 138-156.

Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., Chung, E., 2016. Real-time traffic state estimation in urban corridors from heterogeneous data. *Transp. Res. Part C* 66, 99-118.

Ng, A.Y., Jordan, M.I., Weiss, Y., 2002. On spectral clustering: analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* 14, 849-856.

Noordegraaf, D.V., Annema, J.A., Van Wee, B., 2014. Policy implementation lessons from six road pricing cases. *Transp. Res. Part A* 59, 172-191.

Odeck, J., Kjerkreit, A., 2010. Evidence on users' attitudes towards road user charges—A cross-sectional survey of six Norwegian toll schemes. *Transp. Policy* 17(6), 349-358.

Olszewski, P., Fan, H.S., Tan, Y.-W., 1995. Area-wide traffic speed-flow model for the Singapore CBD. *Transp. Res. Part A* 29(4), 273-281.

Olszewski, P., Xie, L., 2005. Modelling the effects of road pricing on traffic in Singapore. *Transp. Res. Part A* 39(7-9), 755-772.

Ortigosa, J., Menendez, M., Tapia, H., 2013. Study on the number and location of measurement points for an MFD perimeter control scheme: a case study of Zurich. *EURO J. Transp. Logist.* 3(3-4), 245-266.

Osorio, C., Bierlaire, M., 2013. A simulation-based optimization framework for urban transportation problems. *Oper. Res.* 61(6), 1333-1345.

Osorio, C., Chong, L., 2015. A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transp. Sci.* 49(3), 623-636.

Papageorgiou, M., Hadj-Salem, H., Blosseville, J.-M., 1991. ALINEA: A local feedback control law for on-ramp metering. *Transp. Res. Rec.: J. Transp. Res. Board* 1320(1), 58-67.

Pigou, A.C., 1920. *The Economics of Welfare*. MacMillan, London.

Ramezani, M., Haddad, J., Geroliminis, N., 2015. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. *Transp. Res. Part B* 74, 1-19.

Rinnooy Kan, A.H.G., Timmer, G.T., 1989. Chapter IX Global optimization. *Handb. Oper. Res. Manag. Sci.* 1, 631-662.

Saberi, M., Gu, Z., 2018. Transport Strategy refresh background paper: Transport Pricing. City of Melbourne (CoM), Melbourne, Australia.

Saberi, M., Mahmassani, H., 2012. Exploring Properties of Network-wide Flow-Density Relations in A Freeway Network. *Transp. Res. Rec.: J. Transp. Res. Board*(2315), 153-163.

Saberi, M., Mahmassani, H., 2013. Hysteresis and capacity drop phenomena in freeway networks: Empirical characterization and interpretation. *Transp. Res. Rec.: J. Transp. Res. Board*(2391), 44-55.

Saberi, M., Mahmassani, H.S., Hou, T., Zockaie, A., 2014. Estimating Network Fundamental Diagram Using Three-Dimensional Vehicle Trajectories: Extending Edie's Definitions of Traffic Flow Variables to Networks. *Transp. Res. Rec.: J. Transp. Res. Board*(2422), 12-20.

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., 1989. Design and analysis of computer experiments. *Stat. Sci.*, 409-423.

Saeedmanesh, M., Geroliminis, N., 2016. Clustering of heterogeneous networks with directional flows based on "Snake" similarities. *Transp. Res. Part B* 91, 250-269.

Saeedmanesh, M., Geroliminis, N., 2017. Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. *Transp. Res. Part B* 105, 193-211.

Santos, G., 2005. Urban congestion charging: a comparison between London and Singapore. *Transp. Rev.* 25(5), 511-534.

Seo, T., Kusakabe, T., Asakura, Y., 2015. Estimation of flow and density using probe vehicles with spacing measurement equipment. *Transp. Res. Part C* 53, 134-150.

Shafiei, S., Gu, Z., Saberi, M., 2018. Calibration and Validation of a Simulation-based Dynamic Traffic Assignment Model for a Large-Scale Congested Network. *Simul. Modell. Pract. Theory* 86, 169-186.

Sheffi, Y., 1985. *Urban Transportation Network: Equilibrium Analysis with Mathematical Programming Methods*. Prentice Hall, Englewood Cliffs, NJ.

Shepherd, S., May, A., Koh, A., 2007. How to design effective road pricing cordons, *Proceedings of the 11th World Conference on Transportation Research*, Berkeley, USA.

Shepherd, S., Sumalee, A., 2004. A Genetic Algorithm Based Approach to Optimal Toll Level and Location Problems. *Netw. Spat. Econ.* 4(2), 161-179.

Shubert, B.O., 1972. A sequential method seeking the global maximum of a function. *SIAM J. Numer. Anal.* 9(3), 379-388.

Simoni, M.D., Pel, A.J., Waraich, R.A., Hoogendoorn, S.P., 2015. Marginal cost congestion pricing based on the network fundamental diagram. *Transp. Res. Part C* 56, 221-238.

Sørensen, C.H., Isaksson, K., Macmillen, J., Åkerman, J., Kressler, F., 2014. Strategies to manage barriers in policy formation and implementation of road pricing packages. *Transp. Res. Part A* 60, 40-52.

Spall, J.C., 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Automat. Contr.* 37(3), 332-341.

Spall, J.C., 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Trans. Aerosp. Electron. Syst.* 34(3), 817-823.

Spall, J.C., 2003. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, Hoboken, New Jersey.

Sumalee, A., 2004. Optimal road user charging cordon design: a heuristic optimization approach. *Comput.-Aided Civ. Inf. Eng.* 19(5), 377-392.

Sumalee, A., 2007. Multi-concentric optimal charging cordon design. *Transportmetrica* 3(1), 41-71.

Sumalee, A., May, T., Shepherd, S., 2005. Comparison of judgmental and optimal road pricing cordons. *Transp. Policy* 12(5), 384-390.

Tan, Z., Yang, H., Guo, R., 2015. Dynamic congestion pricing with day-to-day flow evolution and user heterogeneity. *Transp. Res. Part C* 61, 87-105.

Transurban, 2016. Changed Conditions Ahead: The Transport Revolution and What it Means for Australians, Melbourne, Australia.

TSS, 2014. Aimsun 8 Dynamic Simulators Users' Manual, Barcelona, Spain.

Tyagi, V., Kalyanaraman, S., Krishnapuram, R., 2012. Vehicular traffic density state estimation based on cumulative road acoustics. *IEEE Trans. Intell. Transp. Syst.* 13(3), 1156-1166.

Verhoef, E.T., 2002. Second-best congestion pricing in general networks. Heuristic algorithms for finding second-best optimal toll levels and toll points. *Transp. Res. Part B* 36(8), 707-729.

Walters, A.A., 1961. The theory and measurement of private and social cost of highway congestion. *Econometrica: J. Econometric Soc.*, 676-699.

Wang, Y., Cao, J., Li, W., Gu, T., Shi, W., 2017. Exploring traffic congestion correlation from multiple data sources. *Pervasive Mob. Comput.* 41, 470-483.

Whitty, J.M., 2007. Oregon's Mileage Fee Concept and Road User Fee Pilot Program. Oregon Department of Transportation, USA.

Yang, H., Huang, H., 2005. *Mathematical and economic theory of road pricing*. Elsevier, Oxford.

Yang, H., Meng, Q., Lee, D.-H., 2004. Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions. *Transp. Res. Part B* 38(6), 477-493.

Yang, H., Zhang, X., 2003. Optimal toll design in second-best link-based congestion pricing. *Transp. Res. Rec.: J. Transp. Res. Board*(1857), 85-92.

Yang, H., Zhang, X., Huang, H., 2002. Determination of optimal toll levels and toll locations of alternative congestion pricing schemes, *Proceedings of the 15th International Symposium on Transportation and Traffic Theory*, Adelaide, Australia, pp. 519-540.

Yang, H., Zhang, X., Meng, Q., 2004. Modeling private highways in networks with entry–exit based toll charges. *Transp. Res. Part B* 38(3), 191-213.

Ye, H., Yang, H., Tan, Z., 2015. Learning marginal-cost pricing via a trial-and-error procedure with day-to-day flow dynamics. *Transp. Res. Part B* 81, 794-807.

Yin, Y., Lou, Y., 2009. Dynamic tolling strategies for managed lanes. *J. Transp. Eng.* 135(2), 45-52.

Zhang, L., Garoni, T.M., de Gier, J., 2013. A comparative study of Macroscopic Fundamental Diagrams of arterial road networks governed by adaptive traffic signal systems. *Transp. Res. Part B* 49, 1-23.

Zhang, X., Yang, H., 2004. The optimal cordon-based network congestion pricing problem. *Transp. Res. Part B* 38(6), 517-537.

Zhang, Z., Wang, Y., Ahn, Y.-Y., 2013. Overlapping community detection in complex networks using symmetric binary matrix factorization. *Phys. Rev. E* 87(6), 062803.

Zheng, N., Rérat, G., Geroliminis, N., 2016. Time-dependent area-based pricing for multimodal systems with heterogeneous users in an agent-based environment. *Transp. Res. Part C* 62, 133-148.

Zheng, N., Waraich, R.A., Axhausen, K.W., Geroliminis, N., 2012. A dynamic cordon pricing scheme combining the Macroscopic Fundamental Diagram and an agent-based traffic model. *Transp. Res. Part A* 46(8), 1291-1303.

Zheng, Z., Liu, Z., Liu, C., Shiwakoti, N., 2014. Understanding public response to a congestion charge: A random-effects ordered logit approach. *Transp. Res. Part A* 70, 117-134.

Zockaie, A., Saberi, M., Saedi, R., 2018. A resource allocation problem to estimate network fundamental diagram in heterogeneous networks: Optimal locating of fixed measurement points and sampling of probe trajectories. *Transp. Res. Part C* 86, 245-262.

APPENDIX A. MESOSCOPIC SIMULATION MODEL

The developed mesoscopic DTA model of Melbourne, Australia is deployed in AIMSUN as a discrete-event lane-based simulation. See Shafiei et al. (2018) for a thorough description of how the model has been developed. Each link has information about its geometry and necessary traffic flow parameters such as capacity, speed limit, and jam density are defined. Each node is modeled as a queue server. While being able to replicate traffic dynamics and phenomena such as queue spillback, mesoscopic simulation as compared with the microscopic counterpart largely eases the computational complexity of simulating large-scale dynamic traffic networks by using a simplified car-following model (TSS, 2014).

The network configuration of the Melbourne metropolitan area is obtained from the Victorian Integrated Transport Model (VITM). Figure A. 1 shows the extracted sub-network from the greater Melbourne area model that is used in this thesis for toll optimization. The sub-network bounded by the red dash lines has in total 4,375 links, 1,977 nodes, and 492 centroids. The inner rectangle covers the Melbourne CBD where congestion tends to be the severest and hence, represents the PZ. There are totally 282 links, 91 nodes, and 30 centroids in the PZ. In the simulation model, signal timing at major intersections is set as actuated control using the Sydney Coordinated Adaptive Traffic System (SCATS) data including the maximum cycle time, the minimum green time, and the turning movements for each phase. While traffic flow parameters for freeway links are calibrated against loop detector data from multiple months (Gu et al., 2016, 2018), the time-dependent OD demand is calibrated and validated using multi-source traffic data (Shafiei et al., 2018).

Figure A. 1 The extracted sub-network from the greater Melbourne area model

APPENDIX B. LIST OF ABBREVIATIONS

Abbreviation	Full name
ALS	Area licensing scheme
CBD	Central business district
CV	Cross validation
DIRECT	DIviding RECTangles
DTA	Dynamic traffic assignment
DOE	Design of experiments
EDA	Estimation of distribution algorithm
EI	Expected improvement
ERP	Electronic road pricing
GA	Genetic algorithm
HSA	Hierarchical search algorithm
JDDT	Joint distance and delay toll
JDTT	Joint distance and time toll
LHS	Latin hypercube sampling
MCP	Marginal-cost pricing
MLE	Maximum likelihood estimate/estimation
MPEC	Mathematical programming with equilibrium constraints
NFD	Network fundamental diagram
OD	Origin-destination
PI	Proportional-integral
PZ	Pricing zone
RSM	Response surface method
RK	Regressing kriging
SA	Stochastic approximation
SO or SBO	Simulation-based optimization
SPSA	Simultaneous perturbation stochastic approximation

STA	Static traffic assignment
SymNMF	Symmetric nonnegative matrix factorization
TAP	Toll area problem
TDM	Travel demand management
TDP	Toll design problem
TLP	Toll level problem
VTT	Value of travel time