Computationally Efficient Simulation-based Optimization Algorithms for Large-scale Urban Transportation Problems

by

Linsen Chong

B.S., Southeast University (2009)
M.S., Virginia Polytechnic Institute and State University (2011)

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

Doctor of Philosophy in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Signature redacted

Author .....
Department of Civil and Environmental Engineering
May 18, 2017

Signature redacted

Certified by .....
Carolina Osorio
Assistant Professor of Civil and Environmental Engineering
Thesis Supervisor

Signature redacted

Accepted by .....
Jesse Kroll
Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee

Computationally Efficient Simulation-based Optimization Algorithms for Large-scale Urban Transportation Problems

by

Linsen Chong

Submitted to the Department of Civil and Environmental Engineering
on May 18, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Transportation

Abstract

In this thesis, we propose novel computationally efficient optimization algorithms that derive effective traffic management strategies to reduce congestion and improve the efficiency of urban transportation systems. The proposed algorithms enable the use of high-resolution yet computationally inefficient urban traffic simulators to address large-scale urban transportation optimization problems in a computationally efficient manner.

The first and the second part of this thesis focus on large-scale offline transportation optimization problems with stochastic simulation-based objective functions, analytical differentiable constraints and high-dimensional decision variables. We propose two optimization algorithms to solve these problems. In the first part, we propose a simulation-based metamodel algorithm that combines the use of an analytical stationary traffic network model and a dynamic microscopic traffic simulator. In the second part, we propose a metamodel algorithm that combines the use of an analytical transient traffic network model and the microscopic simulator. In the first part, we use the first metamodel algorithm to solve a large-scale fixed-time traffic signal control problem of the Swiss city of Lausanne with limited simulation runs, showing that the proposed algorithm can derive signal plans that outperform traditional simulation-based optimization algorithms and a commercial traffic signal optimization software. In the second part, we use both algorithms to solve a time-dependent traffic signal control problem of Lausanne, showing that the metamodel with the transient analytical traffic model outperforms that with the stationary traffic model.

The third part of this thesis focuses on large-scale online transportation problems, which need to be solved with limited computational time. We propose a new optimization framework that combines the use of a problem-specific model-driven method, i.e., the method proposed in the first part, with a generic data-driven supervised machine learning method. We use this framework to address a traffic responsive control problem of Lausanne. We compare the performance of the proposed framework with the performance of an optimization framework with only the model-driven method and an optimization framework with only the data-driven method, showing that the

proposed framework is able to derive signal plans that outperform the signal plans derived by the other two frameworks in most cases.

Thesis Supervisor: Carolina Osorio
Title: Assistant Professor of Civil and Environmental Engineering

Acknowledgments

First, I would like to thank my advisor Professor Carolina Osorio. It has been truly a great pleasure to work with her at MIT for the last six years. I constantly learn from her expertise, her guidance, her advice, her critical thinking, and her positive attitude towards everything. She has always set very high standard for me, and has taught by example. Carolina is always kind and encouraging, which makes our discussions fruitful and enjoyable. Most of the time, I feel like I was sharing my thoughts with an older sister than an advisor. She helped me keep motivated, gain courage, and be confident when I was struggling with many difficulties during this journey. I could not have been who I am without her. This thesis is as much a product of her guidance and advice as it is of my effort. Carolina, No podría encontrar palabras para expresar mi agradeciento a ti. ¡Mil gracias!

I would like to thank my thesis committee members, Professor Moshe Ben-Akiva and Professor Saurabh Amin. They have provided me with insightful comments and suggestions on this research. In addition, I would like to thank Professor Saurabh Amin for his help in my job search. I would also like to thank Professor Moshe Ben-Akiva for serving as my academic advisor during my first year at MIT.

Additionally, many thanks go to Roberta Pizzinato for being so helpful for scheduling meetings and travel arrangements. I would also like to thank CEE staff Kris Kipp, Kiley Clapper, Angela Mickunas, Sarah Smith, Katie Rosa and Eunice Kim for their administrative assistance.

Many thanks go to my friends and colleagues at MIT, without whom I could have never gone so far. It has been a pleasure to have my lab mates Xiao Chen, Krishna Kumar Selvam, Chao Zhang, Tianli Zhou, Nate Bailey, Timothy Tay and many others, with whom I have shared many discussions on research, course work and other interesting topics. I would like to thank my friends at MIT, Seonkyoo Yoon, Enyang Huang, Zeid Alghareeb, Yan Zhao, Yin Wang, Xiao Fang, Zhan Zhao, Haizheng Zhang, Chiwei Yan, Manxi Wu and many others for accompanying me during this long journey. A special thanks go to Feifei Yu for being a true friend

since my first day at MIT. Our weekly exploration of neighbours and restaurants in Somerville has become an integral part of my life at MIT.

I am indebted to Professor Montasir Abbas, my former advisor at Virginia Tech, who brought me into the world of transportation and continuously supports me until today. I must thank Ms. Alejandra Medina at Virginia Tech Transportation Institute, for her support during my stay at Virginia Tech and here at MIT. I would also like to thank my friends at Virginia Tech, especially Ao Chen and Asmita Gharat, for their friendship over these years.

Most importantly, I would like to thank my parents for their constant love, support, understanding and encouragement. This thesis is dedicated to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Motivation and Objective

Urban transportation systems are becoming indispensable with the surge of daily passenger and vehicle mobility demand over the past few years. With this increase in travel demand, urban transportation systems of many cities have faced congestion problems, causing them to be less efficient. Congestion causes significant delays, fuel overconsumption and high concentration of harmful emissions. According to Texas Transportation Institute's Urban Mobility Report (Schrank et al.; 2015), in 2014 alone, congestion in urban areas caused 6.5 billion hours of delay and 3.1 billion gallons of wasted fuel in the Unites States.

Developing effective traffic management strategies of urban transportation systems is considered to be a cost-effective approach to reduce congestion. Developing these strategies can be viewed as addressing city-scale optimization problems, with the purpose of finding strategies that improve network performance (e.g., reduce average trip travel time, etc.).

Recently, there is an interest in using high-resolution simulation models, such as microscopic traffic simulators, to address optimization problems. These simulation models can describe intricate traffic dynamics observed in urban areas. However, since they are computationally expensive to evaluate, using them to address large-scale urban traffic management problems with limited simulation budgets (i.e., simulation

runs) is challenging. Traditional optimization algorithms, such as many black box optimization algorithms (e.g., genetic algorithms), require numerous simulation runs to address even low or medium dimensional problems (e.g., Stevanovic et al. (2009) and Park et al. (2009)), which makes them not suitable to address many high-dimensional optimization problems for large-scale urban networks. Therefore, there is a need to develop optimization algorithms that use these high-resolution yet computationally expensive simulation models to address high-dimensional optimization problems for large-scale urban networks.

In addition, with the emergence of data from ubiquitous sensors, data-driven methods have been received much attention in transportation research, such as in the field of traffic estimation (see Antoniou et al. (2013) for a review), traffic management (see Anderson (2015, Chap. 6) for a review), etc. These methods are generic methods that are developed based mainly on historical data and do not use problem-specific information (e.g., network topology) to address transportation problems. These data-driven methods are computationally efficient and are likely to be more realistic. However, they may not be able to perform well (e.g., provide good estimates of traffic conditions on the network level) when there is a low coverage of sensors in urban networks (i.e., data is sparse). On the other hand, model-driven optimization methods, i.e., methods that use problem-specific traffic network models, may be able to perform well even with sparse data. This is because they can rely on many network modelling assumptions in order to provide good performance estimates on the unobservable parts of the networks. Thus, we believe that there is a need to combine model-driven methods with data-driven methods in optimization algorithms, such that they can benefit from the advantages of both methods.

In this thesis, we develop novel computationally efficient optimization algorithms that use high-resolution traffic simulators to address large-scale optimization problems with limited computational budgets (i.e., limited number of simulation runs). These optimization algorithms are built upon a simulation-based metamodel algorithm proposed by Osorio and Bierlaire (2013). The latter is computationally efficient for medium scale problems yet not scalable for large-scale problems. Our algorithms,

on the other hand, are scalable and computationally efficient for optimization problems of high dimension. This research allows transportation agencies to use their high-resolution simulators to address a variety of practical transportation problems.

In this thesis, we are interested in addressing two types of optimization problems: offline and online problems. For the offline problems, we are interested in problems with high-dimensional decision variables, simulation-based stochastic objective functions and analytical differentiable constraints. For online problems, we are interested in problems that consist of a sequence of time intervals where the solution of each interval needs to be found in a time efficient manner (e.g., in real time). The goals of this thesis are as follows:

1. To provide practical computationally efficient optimization algorithms for large-scale offline simulation-based transportation problems. This allows transportation agencies to use high-resolution yet computationally expensive stochastic traffic simulators to address a variety of traffic management problems within a limited computational time. These traffic management problems include fixed-time traffic signal control problems and time-dependent traffic signal control problems.

2. To provide an optimization framework that combines real-time feasible model-driven methods and data-driven methods to address online large-scale transportation optimization problems, such as traffic responsive control problems. The framework aims to combine the advantages of model-driven methods and data-driven methods in order to make a trade-off between realism, scalability, robustness, and computational efficiency. The framework also allows high-resolution simulators to be used for online optimization problems, where computational time becomes a hard constraint.

1.2 Thesis Contributions

This thesis focuses on high-dimensional simulation-based optimization problems for large-scale networks. The contributions of this thesis include:

Optimization algorithms:

This thesis proposes computationally efficient algorithms for a variety of offline

and online transportation optimization problems.

To address offline problems, we develop novel simulation-based optimization algorithms that allow the use of high-resolution yet computationally expensive traffic simulators to address high-dimensional optimization problems for large-scale networks. Unlike traditional simulation-based optimization algorithms that normally require thousands of simulation runs, the proposed algorithms can address high-dimensional simulation-based optimization problems with tens or hundreds of runs. The efficiency of the proposed algorithms is achieved through the formulation of problem-specific analytical traffic network models, which are tractable network models. The proposed algorithms combine the use of computationally efficient analytical traffic network models and computationally expensive traffic simulators.

To address online problems, we propose a novel optimization framework that combines the use of a problem-specific model-driven method and a data-driven method. To the best of our knowledge, this is the first time that these two types of methods are used jointly in an online optimization framework. In our case study, we have shown that our framework outperforms a framework with only the model-driven method and a framework with only the data-driven method.

Analytical traffic models:

In this thesis, we have developed two analytical traffic network models: a stationary model and a transient model. They are tractable models and are suitable to be used for addressing large-scale optimization problems. Since they provide problem-specific structural information of the underlying network, they can help our proposed optimization algorithms identify good solutions with limited simulation runs.

The stationary traffic model is developed by combining ideas from traffic flow theory and finite capacity queueing theory. It captures the spatial distribution of congestion throughout the network. The transient traffic model is built upon the stationary traffic model and it also takes into account the temporal propagation of congestion. Thus, it can capture both spatial and temporal distributions of congestion throughout the network.

Applications:

The proposed optimization algorithms can be used in a variety of online and offline optimization problems. In the case study of this thesis, we address three types of traffic control problems of the Swiss city of Lausanne: 1) a fixed-time traffic signal control problem, 2) a time-dependent traffic signal control problem, and 3) a traffic responsive control problem. The network consists of over 600 links, 200 intersections with 99 control variables. It is considered to be large-scale in the field of simulation-based optimization and traffic signal control (Aboudolas et al.; 2007).

To address Problem (1), we develop a simulation-based optimization algorithm that embeds the stationary analytical traffic model. We demonstrate that, in terms of the quality of solutions (i.e., signal plans), it outperforms a traditional simulation-based optimization algorithm, and does it with only 150 simulation runs. It also outperforms a commercial traffic signal control software. To address Problem (2), we develop a simulation-based optimization algorithm that embeds the transient analytical traffic model, and demonstrate that it outperforms the optimization algorithm that embeds the stationary traffic model. To address Problem (3), we develop an online optimization framework that combines a model-driven method (i.e., the optimization algorithm developed for Problem (1) and a data-driven method (i.e., a supervised classification method). The framework is able to derive signal plans that outperform the signals plans derived by a framework with only the model-driven method and a framework with only the data-driven method most of the time.

1.3 Thesis Structure

Chapter 2 proposes a computationally efficient simulation-based optimization algorithm for large-scale simulation-based urban transportation problems. The algorithm combines the use of the proposed stationary analytical traffic model and a microscopic traffic simulator. The method of Chapter 2 has been published as: Osorio, C. and Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems, Transportation Science 49

(3): 623-636. Chapter 3 proposes a simulation-based optimization algorithm designed for dynamic urban transportation problems. The algorithm combines the use of the proposed transient traffic model and a microscopic traffic simulator. The method of Chapter 3 has been published as: Chong, L. and Osorio, C. (forthcoming). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems, Transportation Science Forthcoming. Chapter 4 proposes a optimization framework for online transportation optimization problems. Chapter 5 summarizes this thesis and presents future research directions. Appendix A is the appendix of Chapter 2. Appendix B is the appendix of Chapter 3. Appendix C is the appendix of Chapter 4.

Chapter 2

A Scalable Metamodel Simulation-based Optimization Algorithm for Large-scale Urban Transportation Problems

This chapter presents a novel computationally efficient optimization algorithm for large-scale simulation-based (SO) urban optimization transportation problems with high-dimensional decision variables. We propose a novel metamodel SO algorithm that combines information from a stationary analytical traffic model and information from a microscopic traffic simulator. The method of this chapter has been published as: Osorio, C. and Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems, Transportation Science 49 (3): 623-636.

2.1 Introduction

The massive amount and variety of mobility data that can now be collected through, for instance, ubiquitous mobile devices, is enhancing our fundamental understanding of individual mobility. For instance, it improves our understanding of the intricate

behavior of travelers, e.g., how they make activity and thereby travel decisions, and how these decisions are motivated by an underlying objective to enhance their well-being.

State-of-the-art high-resolution traffic simulation models, such as microscopic traffic simulation models, embed such disaggregate models of traveler behavior (e.g., departure time choice, multi-modal route choice, access and response to en-route traffic information), and account for behavior heterogeneity. They represent individual vehicles, and can therefore be coupled with vehicle-specific simulators (e.g., propulsion simulators) to yield detailed estimates of the performance of vehicles (e.g., energy consumption or emissions estimates) in networks with complex topologies and complex traffic dynamics. Additionally, microscopic simulators provide a detailed representation of the underlying supply (e.g., variable message signs, public transport priorities).

Microscopic traffic simulators describe in detail the interactions between (i) vehicle performance, (ii) traveler behavior and (iii) the underlying transportation infrastructure, and yield an elaborate description of traffic dynamics in urban networks. They are therefore suitable tools to address transportation problems where a detailed representation of either of these three components should be accounted for.

Microscopic simulators are popular tools used in practice to evaluate the performance of a set of predetermined transportation strategies. Cities such as Toronto, New York, Boston, Stockholm and Hong Kong have used these tools to inform their planning and operations decisions (Traffic Technology International; 2012a,b; Papayannoulis et al.; 2011; Toledo et al.; 2003; Hasan; 1999).

For a given strategy, these simulators can provide accurate and detailed performance estimates. Their use is mostly limited to what-if analysis (also called scenario-based analysis) or sensitivity analysis. That is, they are used to evaluate the performance of a set of predetermined transportation alternatives (e.g., traffic management or network design alternatives), such as in Bullock et al. (2004), Ben-Akiva et al. (2003), Hasan et al. (2002), Stallard and Owen (1998), Gartner and Hou (1992) and Rathi and Lieberman (1989). See further references in Ben-Akiva et al. (2003).

The numerous models of disaggregate traveler behavior, vehicle-performance and supply components lead to detailed performance estimates, yet also to models which are expensive to develop and calibrate, and computationally expensive to evaluate. Thus, an accurate estimation of performance is computationally costly to obtain. Additionally, these simulators derive stochastic nonlinear, and typically nonconvex, performance measures with no closed-form available. For these reasons, the use of these simulators to address optimization problems is a challenge.

Currently, the use of these simulation tools is mostly limited to what-if analysis. With the ubiquity of access to real-time traffic information, and the increasing number of prevailing and interacting traffic control strategies, traffic dynamics of congested networks are becoming more and more intricate. Thus, determining a priori a set of alternatives with good local and network-wide performance is no longer feasible. Thus, there is a need to embed these detailed simulators within optimization frameworks in order to systematically identify alternatives with improved local and network-wide performance. Additionally, given the high cost of developing large-scale simulation tools, transportation projects would benefit from computationally efficient methods that allow the use of simulators to go beyond a what-if analysis.

This chapter proposes a simulation-based optimization (SO) method that allows large-scale urban transportation problems to be addressed with detailed microscopic traffic simulators. We focus on problems where the objective function is derived from the simulator and, thus, no closed-form analytical expression is available. The problems have general (e.g., nonconvex) constraints. Closed-form analytical and differentiable expressions are available for all constraints (i.e., the constraints are not simulation-based).

These urban transportation problems can be formulated as:

$$\min_{x} f(x, z; p) = E[F(x, z; p)] \tag{2.1}$$

$$g(x; p) = 0, \tag{2.2}$$

where the purpose is to minimize the expected value of a given stochastic performance

measure F, x denotes the deterministic continuous decision vector, z denotes other endogenous variables, and p denotes the deterministic exogenous parameters. For instance, in this chapter we use the proposed SO approach to solve a traffic signal control problem where F denotes trip travel time, x represents the green times of the signal phases, z accounts, for instance, for signalized link capacities, route choice decisions, and p accounts, for instance, for the network topology, the total traffic demand, and fixed lane attributes (e.g., length, grade, maximum speed). Constraint (2.2) is a general formulation for any type of constraint (i.e., inequality constraints can be transformed and expressed as equality constraints of the form (2.2)). To summarize, Problem (2.1)- (2.2) considers a simulation-based objective function and general, analytical and differentiable constraints.

The main challenges of addressing Problem (2.1)-(2.2) lie in the objective function, $f(x, z; p)$. The challenges are the following.

- The function f has no known analytical form. We can merely estimate it by running simulation replications of the stochastic traffic simulator.

- An accurate estimation of f is computationally costly to obtain. It involves running numerous simulation replications. In a high-resolution urban traffic simulator, such as the one used in this chapter, running a single replication is costly because it involves simulating the travel behavior of typically tens of thousands of individual travelers. The behavior of a single traveler is defined by hundreds of pre-trip and en-route travel decisions (e.g., route choice, lane-changing), which are each simulated by sampling from stochastic travel behavioral models.

- The function f is typically nonconvex. For example, in Section 2.4 we study a signal control problem where f represents the expected trip travel time of the travelers. In the simulation model, the travel decisions of a given traveler (e.g., route choice) can depend on the state of the network (e.g., congestion patterns), which itself is a consequence of the past decisions of numerous travelers. Hence, the mapping of a signal plan (the decision vector) to network-wide traffic

assignment and to the corresponding expected trip travel time (the objective function) is intricate.

This highlights the general complexity of simulation-based problems across all application fields, as well as the additional challenges that are unique to urban transportation problems.

This chapter proposes a technique that can efficiently address generally constrained large-scale simulation-based urban transportation problems. The performance of the technique is evaluated by considering a network-wide traffic signal control problem. This problem is considered large-scale and complex for derivative-free algorithms, signal control algorithms and simulation-based optimization algorithms.

Additionally, the chapter focuses on SO techniques with good short-term performance, i.e., computationally efficient methods that can identify alternatives with improved performance within a tight computational budget. The computational budget can be defined as a limited number of simulation runs or a limited simulation run time. Such techniques respond to the needs of transportation practitioners by allowing them to address problems in a practical manner.

The remainder of this section reviews past work in this field.

2.1.1 Literature Review

Few SO methods that embed microscopic simulators have been developed (Li et al.; 2010; Stevanovic et al.; 2008; Branke et al.; 2007; Yun and Park; 2006; Hale; 2005; Joshi et al.; 1995). The most common approach is the use of heuristic algorithms and, in particular, the use of genetic algorithms (see Yun and Park (2006) for a review). These methods embed microscopic simulators within general-purpose optimization algorithms. They treat the simulator as a black-box, using no a priori structural information about the underlying transportation problem (e.g., network structure). They therefore require a large number of simulated observations in order to identify transportation strategies (i.e., trial points) with improved performance.

This chapter proposes an SO technique with good short-term performance suitable

for microscopic traffic simulators to be used to address complex high-dimensional problems. In order to derive computationally efficient methods that embed inefficient simulators, information from other more efficient (i.e., tractable) models that provide analytical structural information to the algorithm should be used throughout the optimization process.

In general, methods to address SO problems can be classified as direct-search methods, stochastic gradient methods and metamodel methods. For reviews of SO methods see Hachicha et al. (2010), Barton and Meckesheimer (2006), Fu et al. (2005). This chapter focuses on metamodel methods. For a description of why metamodel techniques are a suitable approach to address complex simulation-based transportation problems, see Osorio and Bierlaire (2013).

Metamodel methods build an analytical approximation of the simulation-based components of the optimization problem (e.g., objective function, constraints). In this chapter, the objective function is simulation-based. Thus, the metamodel provides an analytical approximation of the objective function. By resorting to a metamodel approach, the stochastic response of the simulation is replaced by an analytical response function (the metamodel), such that deterministic optimization techniques can be used. Metamodel techniques use an indirect-gradient approach, i.e., they compute the gradient of the metamodel, which is a deterministic function. Thus, traditional deterministic gradient-based optimization algorithms for generally constrained problems can be used.

Reviews of metamodels are given by Conn et al. (2009b), Barton and Meckesheimer (2006) and Søndergaard (2003). Metamodels can be classified as either physical or functional metamodels (Søndergaard; 2003). Physical metamodels are application or problem-specific metamodels. Their functional form and parameters have a physical interpretation. Functional metamodels are general-purpose (i.e., generic) functions chosen based on their analytical tractability. The most common general-purpose metamodel is the use of low-order polynomials, and particularly of quadratic polynomials (Conn et al.; 2009b; Kleijnen; 2008; Marti; 2008). Other general-purpose metamodels include spline models, radial basis functions and Kriging models (Kleij-

nen et al.; 2010; Wild et al.; 2008; Barton and Meckesheimer; 2006).

The existing metamodels consist of either physical or functional components. Recent work has proposed a metamodel that is a combination of a functional and a physical metamodel (Osorio and Bierlaire; 2013). The functional component ensures asymptotic metamodel properties necessary for convergence analysis (such as full linearity (Conn et al.; 2009a)). The physical component is an analytical and differentiable macroscopic traffic model. It provides a problem-specific analytical approximation of the objective function, unlike the generic approximation provided by the functional component. The physical component therefore yields structural information about the problem at hand, which enables the identification of well performing alternatives (i.e., trial points) with very small samples (i.e., good short-term algorithmic performance). The physical component used here is an analytical differentiable queueing network model. This macroscopic traffic model is less detailed and accurate than the simulator, yet is computationally efficient to evaluate.

This combined use of functional and physical metamodels allows information from the detailed, yet inefficient, microscopic simulator to be combined with analytical information from a more efficient macroscopic model. This leads to an algorithm with a good detail-tractability trade-off and good short-term performance.

This physical and functional metamodel approach has been used to efficiently address complex urban transportation problems, such as signal control problems that account for detailed (also called microscopic) vehicle-specific energy consumption patterns (Osorio and Nanduri; 2015a), emissions patterns (Osorio and Nanduri; 2015b), and reliable signal control problems that used detailed full distributional travel time estimates provided by the simulator to improve both average trip travel times and travel time reliability (Osorio et al.; 2016).

This approach has been successfully used to control networks with approximately 50 roads, yet is not suitable to address problems for much larger scale networks. This chapter builds upon this existing metamodel SO technique (hereafter referred to as the *initial* method), and proposes a metamodel that can efficiently address high-dimensional simulation-based problems.

In Section 2.2 of this chapter, we present the proposed metamodel framework. We then present the traffic signal control problem which is used to evaluate the scalability and short-term performance of this approach (Section 2.3). Empirical results are detailed in Section 2.4, followed by conclusions (Section 2.5).

2.2 Methodology

The main steps of a metamodel SO algorithm are displayed in Figure 2-1. At a given iteration k, there are a set of points that have been simulated prior to iteration k. We call this set of points the current sample. Step 1 determines which point of the current sample is considered to have the best performance. This point is referred to as the *current iterate* and is denoted x_k. In step 2, the parameters, β_k, of the metamodel, m_k, are fitted based on the current sample. For example, in the algorithm used in the case study of this chapter, the vector β_k is obtained as the solution of a least squares problem that minimizes the distance, over the current sample, between metamodel values and simulation-based objective function estimates. Step 3a solves the following problem (or a subproblem of it):

$$\min_x m_k(x, z; p, \beta_k) \tag{2.3}$$

$$g(x; p) = 0. \tag{2.4}$$

This problem differs from Problem (2.1)-(2.2) in that the simulation-based objective function f of (2.1) is replaced with the metamodel function m_k. The latter is an analytical, and often differentiable, function that depends on the decision vector x, on a vector of endogenous variables z, a vector of exogenous parameters, q, and an iteration-specific metamodel parameter vector, β_k. The solution to this problem is called the *trial point.* Step 3b allows to simulate points that may not be solutions to the approximate problem (2.3)-(2.4). These are known as *model improvement points.* The corresponding sampling strategy that defines these points may, for instance, have the goal of improving the properties of the sampled space (e.g., increase the

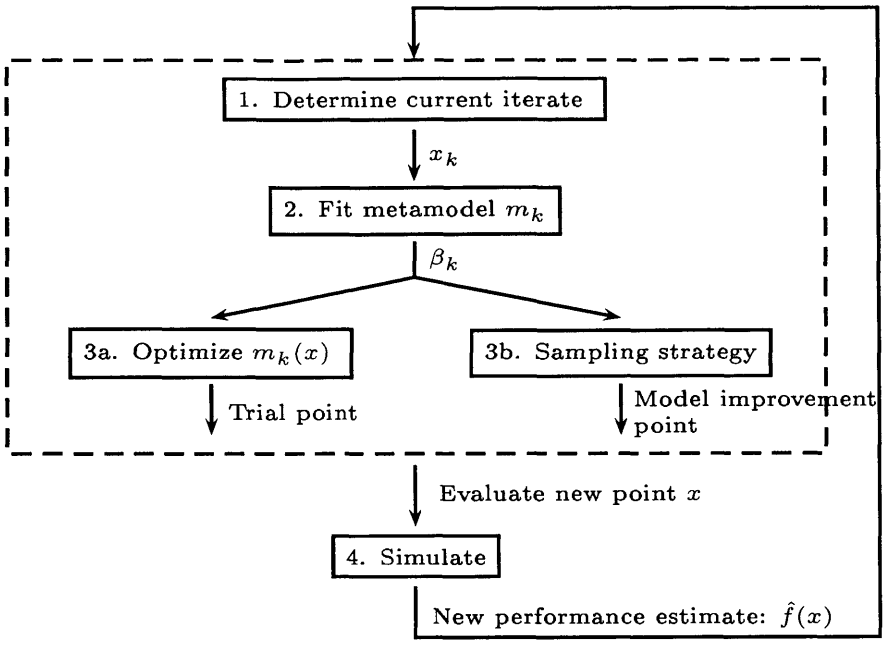


Figure 2-1: Metamodel simulation-based optimization framework (Chong and Osorio; forthcoming)

dimension of the space spanned, sample uniformly, etc.). The new points (trial or model improvement) are simulated in step 4 to obtain an estimate of the objective function, denoted $\hat{f}(x)$. As the iterations advance, more points are sampled, leading to an improved metamodel and to points with improved objective function estimates.

One feature of metamodel methods is that the trial points are derived by solving Problem (2.3)-(2.4), which is analytical and differentiable. Hence, it can be solved with a variety of mainstream solvers. Additionally, the algorithm we use in this chapter (Osorio and Bierlaire; 2013) is a derivative-free algorithm. Hence, it does not require the estimation of first- or second-order derivatives of the SO objective function. Many traditional SO algorithms rely on derivative estimations, which can be computationally costly to obtain in high-dimensional spaces. The use of a derivative-free algorithm is important to achieve computational efficiency.

2.2.1 Metamodel Functional Form

Recall the general form of the urban transportation problems that we address (Problem (2.1)-(2.2)). Since there is no closed-form available for the objective function, f, we use a metamodel to approximate it. The functional form of the metamodel used in this chapter is that proposed by Osorio and Bierlaire (2013). It combines a physical and a functional component. The functional form of the metamodel, at a given iteration k is given by:

$$m_k(x, z; \beta_k, p) = \beta_{k,0} f_A(x, z; p) + \phi(x; \beta_{k,1}, \ldots, \beta_{k,D}), \tag{2.5}$$

where ϕ (the functional component) is a quadratic polynomial in x, and has D coefficients $(\beta_{k,1}, \ldots, \beta_{k,D})$, f_A (the physical component) represents the approximation of the objective function derived by an analytical macroscopic traffic model. It is scaled by the scalar coefficient $\beta_{k,0}$. The parameter vector of the metamodel is represented by $\beta_k = (\beta_{k,0}, \ldots, \beta_{k,D})$.

The metamodel m_k can be interpreted as an analytical and macroscopic approximation of the objective function provided by f_A, which is corrected parametrically by both a scaling factor $\beta_{k,0}$ and a separable error term $\phi(x; \beta_{k,1}, \ldots, \beta_{k,D})$. For details regarding the choice of this functional form, we refer the reader to Osorio and Bierlaire (2013).

The problem solved at a given iteration k of the SO algorithm is of the form:

$$\min_x m_k(x, z; p, \beta_k) \tag{2.6}$$

$$g(x; p) = 0 \tag{2.7}$$

$$h(x, z; p) = 0. \tag{2.8}$$

This problem differs from Problem (2.3)- (2.4) in the Constraint (2.8). This constraint represents the analytical traffic network model used to derive the physical metamodel component (i.e., term f_A of Equation (2.5)). The traffic model used in this chapter is a tractable stationary network model, that is also analytical and

differentiable. It is defined as a system of nonlinear equations.

Problem (2.6)-(2.8) is solved at every iteration of the SO algorithm. Therefore, the development of a computationally efficient SO algorithm requires solving Problem (2.6)-(2.8) efficiently. Hence, the analytical network model (represented by (2.8)) needs to be tractable.

The problem-specific approximation f_A contributes to the computational efficiency of the algorithm in various ways. First, as an analytical and differentiable model, it allows for the use of traditional and computationally efficient algorithms (e.g., gradient-based algorithms) to solve Problem (2.6)-(2.8). Second, it provides an approximation of f in the entire feasible region, i.e., it provides a global approximation. This is in contrast with traditional general-purpose functions (e.g., polynomial functions) that are designed to provide good local approximations of f. Third, the accuracy of this global approximation is independent of the availability of simulation observations. In particular, if few or even no simulation observations are available, this approximation may still provide a suitable approximation of the objective function f. Fourth, if the network model h (Equation (2.8)) is tractable, then Problem (2.6)-(2.8) can be solved efficiently.

In this chapter, we use the same functional component as in Osorio and Bierlaire (2013) (i.e., the quadratic polynomial ϕ). We propose a novel scalable physical component. In Section 2.2.2 we recall the formulation of the physical component of the initial metamodel and describe its limitations. We then present the new formulation of the physical component in Section 2.2.3.

2.2.2 Initial Queueing Network Model

The physical component of the initial metamodel is an urban traffic model based on queueing network theory. It combines ideas from existing traffic models, various national urban transportation norms, and queueing models. The detailed formulation of the model is given in Osorio (2010, Chap. 4) (which is based on the more general queueing network model of Osorio and Bierlaire (2009)). We outline here the main ideas of its formulation.

Each lane of an urban road network is modeled as a queue (and in some cases as a set of queues). In order to account for the limited physical space that a queue of vehicles may occupy we resort to *finite capacity queueing theory*, where there is a finite upper bound on the length of each queue. Each lane is modeled as a finite capacity M/M/1/k queue. The network model analytically approximates the queue interactions among adjacent lanes. Congestion and spillbacks are modeled by what is known in queueing theory as *blocking*. This occurs when a queue is full, and thus blocks arrivals from upstream queues at their current location. This blocking process is described by endogenous variables such as blocking probabilities and unblocking rates. The model consists of a set of nonlinear equations that capture these between-queue interactions.

In the following notation the index i refers to a given queue.

γ_i	external arrival rate;
λ_i	total arrival rate;
μ_i	service rate;
$\tilde{\mu}_i$	unblocking rate;
$\hat{\mu}_i$	effective service rate (accounts for both service and eventual blocking);
ρ_i	traffic intensity;
P_i^f	probability of being blocked at queue i;
k_i	upper bound of the queue length;
N_i	total number of vehicles in queue i;
$P(N_i = k_i)$	probability of queue i being full, also known as the blocking or spillback probability;
p_{ij}	transition probability from queue i to queue j;
$\mathscr{D}_i$	set of downstream queues of queue i.

The queueing network model is formulated as follows.

$$\begin{cases} \lambda_i = \gamma_i + \dfrac{\sum_j p_{ji}\lambda_j(1 - P(N_j = k_j))}{(1 - P(N_i = k_i))}, & \text{(2.9a)} \\ \dfrac{1}{\tilde{\mu}_i} = \displaystyle\sum_{j \in \mathscr{D}_i} \dfrac{\lambda_j(1 - P(N_j = k_j))}{\lambda_i(1 - P(N_i = k_i))\hat{\mu}_j}, & \text{(2.9b)} \\ \dfrac{1}{\hat{\mu}_i} = \dfrac{1}{\mu_i} + P_i^f \dfrac{1}{\tilde{\mu}_i}, & \text{(2.9c)} \\ P(N_i = k_i) = \dfrac{1 - \rho_i}{1 - \rho_i^{k_i+1}} \rho_i^{k_i}, & \text{(2.9d)} \\ P_i^f = \displaystyle\sum_{j \in \mathscr{D}_i} p_{ij} P(N_j = k_j), & \text{(2.9e)} \\ \rho_i = \dfrac{\lambda_i}{\hat{\mu}_i}. & \text{(2.9f)} \end{cases}$$

Equation (2.9a) is a flow conservation equation, it relates flow transmission between upstream and downstream queues. The factor $(1 - P(N_i = k_i))$ represents the probability that queue i is not full (i.e., the queue can receive flow from its upstream queues). If the queue is full, it cannot receive flow from upstream queues, which may lead to spillbacks. Equation (2.9b) defines the rate at which spillbacks at queue i dissipate, $\tilde{\mu}_i$. Equation (2.9c) defines the rate at which queue i dissipates accounting for both spillback and non-spillback states, $\hat{\mu}_i$. It is defined as a function of the service rate of the queue, μ_i. The latter is determined by combining ideas from national transportation norms, and is a function, for instance, of the free flow capacity of the underlying lane. Equation (2.9d) defines the probability that a queue is full, i.e., the spillback probability of the underlying lane. This expression is derived from finite capacity queueing theory (Bocharov et al.; 2004). Equation (2.9e) defines the probability of a vehicle being blocked while at queue i, i.e., the probability that a vehicle at the underlying lane is affected by spillback from a downstream lane. Equation (2.9f) defines the traffic intensity of a queue, it is also derived from traditional finite capacity queueing formulae.

In this model, the exogenous parameters of a given queue are γ_i, μ_i, p_{ij} and k_i. All other parameters are endogenous. When used to solve a signal control problem, the flow capacity of the signalized lanes become endogenous, which makes the corre-

sponding service rates, μ_i, endogenous. In that case, the exogenous parameters are γ_i, p_{ij} and k_i. This is a stationary model with exogenous traffic assignment (the turning probabilities p_{ij} are exogenous). As described in Section 2.5, analytical tractable formulations that describe traffic dynamics are developed in Chapter 3.

As described in Section 2.1.1, this model has been used to solve signal control problems for medium-scale networks. However, it is not sufficiently tractable to address large-scale network problems. For instance, in the case of the Lausanne city network (with over 600 links and 200 intersections), the time needed by a standard nonlinear optimization algorithm to solve the trust region (TR) subproblem (detailed in Section 2.3.2) exceeds 20 minutes. Since this TR subproblem is solved at every iteration of the SO algorithm, it is critical to solve it efficiently.

In this chapter, we propose a more tractable and scalable physical component of the metamodel. It is an approximation of this initial queueing network model. It consists of a simple system of one linear and two nonlinear equations. In particular, as is detailed in Section 2.4.2, the TR subproblem is now solved on average within less than 2 minutes. This significantly enhances the computational efficiency of the SO algorithm, and allows to efficiently address more complex high-dimensional constrained transportation problems.

2.2.3 Tractable Stationary Queueing Network Model

We introduce the following two variables:

- $\hat{\lambda}_i$ effective arrival rate;
- $\hat{\rho}_i$ effective traffic intensity.

These two new variables are defined by:

$$\hat{\lambda}_i = \lambda_i(1 - P(N_i = k_i)) \tag{2.10}$$

$$\hat{\rho}_i = \frac{\hat{\lambda}_i}{\hat{\mu}_i}. \tag{2.11}$$

The tractable stationary queueing network model is given by:

$$\begin{cases} \hat{\lambda}_i = \gamma_i(1 - P(N_i = k_i)) + \sum_j p_{ji}\hat{\lambda}_j & (2.12a) \\ \hat{\rho}_i = \frac{\hat{\lambda}_i}{\mu_i} + \left(\sum_{j\in\mathscr{D}_i} p_{ij}P(N_j = k_j)\right)\left(\sum_{j\in\mathscr{D}_i}\hat{\rho}_j\right) & (2.12b) \\ P(N_i = k_i) = \frac{1-\hat{\rho}_i}{1-(\hat{\rho}_i)^{k_i+1}}(\hat{\rho}_i)^{k_i}. & (2.12c) \end{cases}$$

Equation (2.12a) is obtained directly by inserting Equation (2.10) into Equation (2.9a). Equation (2.12b) is obtained as follows. Multiply Equation (2.9b) and (2.9c), respectively, by $\hat{\lambda}_i$ to obtain:

$$\frac{\hat{\lambda}_i}{\tilde{\mu}_i} = \sum_{j\in\mathscr{D}_i}\frac{\hat{\lambda}_j}{\hat{\mu}_j}, \tag{2.13}$$

$$\hat{\rho}_i = \frac{\hat{\lambda}_i}{\mu_i} + P_i^f\frac{\hat{\lambda}_i}{\tilde{\mu}_i}. \tag{2.14}$$

Insert Equation (2.13) into (2.14) to obtain:

$$\hat{\rho}_i = \frac{\hat{\lambda}_i}{\mu_i} + P_i^f(\sum_{j\in\mathscr{D}_i}\hat{\rho}_j). \tag{2.15}$$

Insert the expression of P_i^f given by Equation (2.9e), and Equation (2.12b) results.

Equation (2.12c) is an approximation of Equation (2.9d) which is obtained by replacing the traffic intensity ρ, by the effective traffic intensity $\hat{\rho}$. That is, we use the expression of the blocking probability of a finite capacity queue, yet approximate the traffic intensity with the effective traffic intensity.

Equation (2.11) defines $\hat{\rho}$ and shows that it may underestimate ρ. For queues with light traffic, we have $\hat{\rho} \approx \rho$, and the two models will yield similar network performance estimates. For congested links, the scalable approximation may underestimate link congestion.

The proposed model consists of three endogenous variables per queue ($\hat{\lambda}_i, \hat{\rho}_i, P(N_i = k_i)$). When using this model to address signal control problems, μ_i also becomes en-

dogenous. This model is defined by one linear and two nonlinear equations. This formulation results in increased computational efficiency, enabling us to address a full city-scale microscopic simulation-based optimization problem.

2.2.4 Example of Functional Form of f_A

As described in Section 2.1.1, one of the advantages of using a physical component in the metamodel is to have problem-specific approximations of the objective function. In this section, we give an example of the functional form of the analytical approximation of the objective function provided by the queueing model, $f_A(x, z; p)$. In Section 2.3, we address a signal control problem, where the objective is to minimize the expected trip travel time. The queueing approximation of this expectation is obtained by applying Little's law (Little; 2011, 1961) to the entire network. It is given by:

$$\frac{\sum_i E[N_i]}{\sum_i \gamma_i (1 - P(N_i = k_i))}, \tag{2.16}$$

where $E[N_i]$ represents the expected number of vehicles in lane i, γ_i is the rate of vehicles entering the network via lane i (i.e., the external arrival rate), and $P(N_i = k_i)$ is the probability that lane i is full (i.e., spillback or blocking probability). The numerator of Equation (2.16) represents the expected number of vehicles in the network, whereas the denominator represents the effective arrival rate to the network. Their ratio yields the expected time in the network.

The expected number of vehicles on lane i, $E[N_i]$, is given by:

$$E[N_i] = \rho_i \left(\frac{1}{1 - \rho_i} - \frac{(k_i + 1)\rho_i^{k_i}}{1 - \rho_i^{k_i+1}} \right). \tag{2.17}$$

This expression is derived in Appendix A.1. In the scalable model proposed in this chapter, ρ_i is approximated by $\hat{\rho}_i$ in Equation (2.17).

2.2.5 SO Algorithm

The SO algorithm used in this chapter is that of Osorio and Bierlaire (2013). It is given in Appendix A.2. It is based on the derivative-free trust region (TR) algorithm proposed by Conn et al. (2009a). For an introduction to trust region (TR) methods, we refer the reader to Conn et al. (2000). They summarize the main steps of a TR method in the *Basic trust region algorithm.* The derivative-free method proposed by Conn et al. (2009a) builds upon the *Basic trust region algorithm* by adding two additional steps: a model improvement step and a criticality step. This algorithm allows for arbitrary metamodels to be used and, unlike traditional TR algorithms, it makes no assumptions on how these metamodels are fitted (interpolation or regression). It is therefore particularly appealing for the simulation-based context where derivatives are costly to estimate and where metamodels fitted via regression are more suitable than their interpolated versions.

At a given iteration k of the SO algorithm, it solves a trust region subproblem and approximates the objective function by the current metamodel m_k (defined in Equation (2.5)). The metamodel parameters (β_k) are fitted via regression based on the simulated observations collected so far. For a detailed description of the algorithm, see Osorio and Bierlaire (2013).

2.3 Traffic Signal Control Problem

This methodology is suitable to address a variety of simulation-based urban transportation optimization problems. In this section, we evaluate the performance of the methodology by considering a large-scale network-wide traffic signal control problem.

2.3.1 Problem Formulation

A detailed review of traffic signal control formulations is given in Appendix A of Osorio (2010). In this chapter, we consider a fixed-time strategy. Fixed-time (also called time of day or pre-timed) strategies are pre-determined based on historical

traffic patterns. They yield one traffic signal setting for the considered time of day. The traffic signal optimization problem is solved offline.

In this chapter, the signal plans of several intersections are determined jointly. For a given intersection and a given time interval (e.g., evening peak period), a fixed-time signal plan is a cyclic (i.e., periodic) plan that is repeated throughout the time interval. The duration of the cycle is the time required to complete one sequence of signals. The cycle times of the intersections controlled in the Lausanne network (used in the case study of this chapter) are 80, 90 or 100 seconds.

A phase is defined as a set of traffic streams that are mutually compatible and that receive identical control. The cycle of a signal plan is divided into a sequence of periods called stages. Each stage consists of a set of mutually compatible phases that all have green. The stage sequence is defined such as to separate conflicting traffic movements at intersections. The cycle may also contain all-red periods, where all streams have red indications, as well as stages with fixed durations (e.g., for safety reasons). The sum of the all-red periods and the fixed periods is called the fixed cycle time.

Cycle times, green splits and offsets are the three main signal timing control variables. The green split corresponds to the ratio of green times (i.e., total duration of a phase) to cycle time. Offsets are defined as the difference in time between the start of cycles for a pair of intersections. Offset settings are especially important in coordinating the signals of adjacent intersections (e.g., to create green waves along arterials or corridors).

In this chapter, cycle times, offsets and all-red durations are kept constant. The stage structure is also given, i.e., the set of lanes associated with each stage as well as the sequence of stages are both known. This is known as a stage-based approach. The decision variables consist of the endogenous green splits of the different intersections.

To formulate this problem we introduce the following notation:

c_i	cycle time of intersection i;
d_i	fixed cycle time of intersection i;
e_n	ratio of fixed green time to cycle time of signalized lane n;
s	saturation flow rate [veh/h];
$x(j)$	green split of phase j;
x_{LB}	vector of minimal green splits;
$\mathscr{I}$	set of intersection indices;
$\mathcal{N}$	set of indices of the signalized lanes;
$\mathscr{P}_I(i)$	set of endogenous phase indices of intersection i;
$\mathscr{P}_N(n)$	set of endogenous phase indices of lane n.

The problem is traditionally formulated as follows:

$$\min_x f(x;p) = E[F(x;p)] \tag{2.18}$$

subject to

$$\sum_{j\in\mathscr{P}_I(i)} x(j) = \frac{c_i - d_i}{c_i},\ \forall i \in \mathscr{I} \tag{2.19}$$

$$x \geq x_{LB}. \tag{2.20}$$

The decision vector x consists of the green splits for each phase. The objective is to minimize the expected trip travel time (Equation (2.18)). The linear constraints (2.19) link the green times of the phases with the available (i.e., non-fixed) cycle time for each intersection. Equation (2.20) ensures lower bounds for the green splits. These bounds are determined based on the prevailing transportation norms.

2.3.2 Trust Region Subproblem

This section presents the trust region (TR) subproblem that is solved at each iteration of the SO algorithm. It is a variation of the signal control problem defined in Section 2.3.1. At a given iteration k, the SO algorithm considers a metamodel $m_k(x, z; \beta_k, p)$, an iterate x_k (point considered to have best performance so far) and a TR radius Δ_k. The TR subproblem is formulated as follows:

$$\min_{x,z} \; m_k = \beta_{k,0} f_A(x, z; p) + \phi(x; \beta_{k,1}, \ldots, \beta_{k,D}) \tag{2.21}$$

subject to

$$\sum_{j \in \mathscr{P}_I(i)} x(j) = \frac{c_i - d_i}{c_i} \; \forall i \in \mathscr{I} \tag{2.22}$$

$$h(x, z; p) = 0 \tag{2.23}$$

$$\mu_n - \sum_{j \in \mathscr{P}_N(n)} x_j s = e_n s, \; \forall n \in \mathcal{N} \tag{2.24}$$

$$\|x - x_k\|_2 \leq \Delta_k \tag{2.25}$$

$$z \geq 0 \tag{2.26}$$

$$x \geq x_{LB}. \tag{2.27}$$

The TR subproblem approximates the objective functions by the metamodel at iteration k, m_k. It contains the constraints of the signal control problem, and includes three additional constraints. Equations (2.22) and (2.27) are the signal control constraints, they correspond to Equations (2.19) and (2.20). The function h of Equation (2.23) represents the queueing network model (Equations (2.12a)-(2.12c)). Equation (2.24) relates the green splits of a phase to the flow capacity of the underlying lanes (i.e., the service rate of the queues). Constraint (2.25) is the trust region constraint. The endogenous variables of the queueing model are subject to positivity constraints (Equation (2.26)). Thus, the TR subproblem consists of a nonlinear objective function subject to nonlinear and linear equalities, a nonlinear inequality and

bound constraints.

2.3.3 Implementation Notes

This problem is solved with the Matlab routine for constrained nonlinear problems, *fmincon*, and its sequential quadratic programming method (Coleman and Li; 1996, 1994). We set the tolerance for relative change in the objective function to 10^{-3} and the tolerance for the maximum constraint violation to 10^{-2}. For further details on the TR subproblem formulation and its implementation, see Osorio and Bierlaire (2013).

We implement the lower bound constraints of Equation (2.27) as nonlinear equations by introducing a new variable v and implementing Equation (2.27) as:

$$x = x_{LB} + v^2. \tag{2.28}$$

We do not enforce the positivity of all endogenous variables (Equation (2.26)) yet check a posteriori that all endogenous variables are positive. In our numerous experiments, we have not encountered a case with a negative value. We insert Equation (2.24) into Equation (2.12b), and implement the two constraints as a single constraint.

For a problem with r endogenous phases, n lanes, o signalized intersections, where each lane is modeled by a single queue (i.e., we have n queues), there are $3n + r$ endogenous variables, which consist of 3 endogenous queueing variables per lane, and the green splits for each phase. There are n linear equations, $2n + o$ nonlinear equations and 1 nonlinear inequality (trust-region constraint).

2.4 Empirical Analysis: Lausanne City Case Study

2.4.1 Lausanne City Network

We evaluate the scalability and short-term algorithmic performance of this framework by solving a large-scale signal control problem. We solve a problem for the entire Swiss

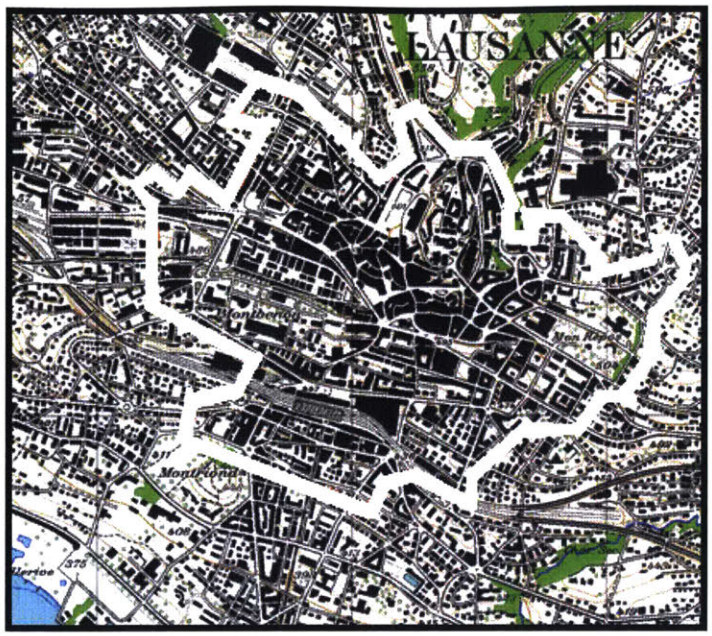


Figure 2-2: Lausanne city road network (adapted from Dumont and Bert (2006))

city of Lausanne. The map is displayed in Figure 2-2, the considered area is delimited in white.

We use a microscopic traffic simulation model of the Lausanne city center developed by Dumont and Bert (2006). It is implemented with the Aimsun simulator (TSS; 2011), and is calibrated for evening peak period demand. Details regarding this Lausanne network are given in Osorio (2010, Chap. 4). In this chapter, the considered demand scenario consists of the first hour of peak period traffic, 5-6 pm.

The road network consists of 603 links and 231 intersections. The signals of 17 intersections are controlled in this problem. The modeled road network is displayed in Figure 2-3, where the 17 intersections are depicted as filled squares. The cycle times of these intersections are 80 seconds (for 2 intersections), 90 seconds (for 13 intersections) and 100 seconds (for 2 intersections). This leads to a total of 99 endogenous phase variables (i.e., the dimension of decision vector is 99).

The queueing model consists of 902 queues. The TR subproblem consists of 2805 endogenous variables with 1821 nonlinear equality constraints, 902 linear equality constraints. The lower bounds of the green splits (Equation (2.20)) are set to 4 seconds according to the Swiss transportation norm (VSS; 1992).

Figure 2-3: Lausanne network model

Performing network-wide signal control of networks with around 70 links and 16 intersections is currently considered large-scale in the field of signal control, as illustrated by recent studies (Aboudolas et al.; 2010, 2007). Thus, the simulation-based signal control problem of this chapter is a challenging large-scale network-wide signal control problem that considers a congested network with a complex topology.

This problem is considered large-scale for existing unconstrained derivative-free algorithms, where the most recent methods are limited to problems with around 200 variables (Conn et al.; 2009b), not to mention the added complexity of constraints and stochasticity. Given the complexity of the underlying simulator, this problem is also considered complex for simulation-based optimization algorithms.

2.4.2 Numerical Results

We compare the performance of the proposed metamodel with a traditional metamodel method that consists only of a functional component, which is a quadratic polynomial with diagonal second derivative matrix (i.e., the metamodel consists of ϕ, defined in Equation (2.5)). In order to compare the two methods, we consider a tight

computational budget, which is defined as a maximum of 150 simulation runs that can be carried out.

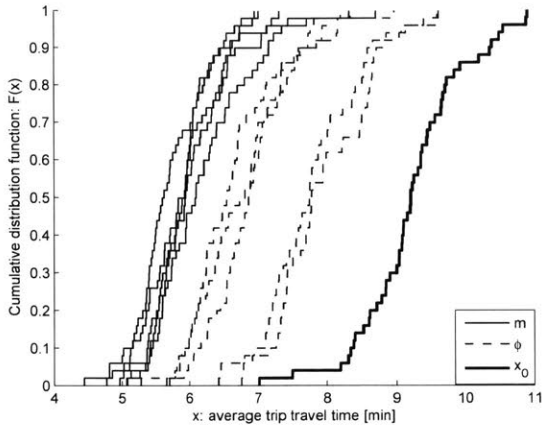
We consider three different initial points (i.e., signal plans). These points are uniformly drawn from the feasible space defined by Equations (2.19) and (2.20). For each initial point, we run the SO algorithm five times, each time allowing for 150 simulation runs. Thus, for each method and each initial point, we derive five "optimal" (or proposed) signal plans. We then use the simulator to evaluate in detail the performance of the proposed signal plans. For each proposed plan signal, we run 50 replications. We compare the empirical cumulative distribution function (cdf) of the average trip travel times obtained from these 50 replications.

Each plot of Figure 2-4 considers a different randomly drawn initial point. Each curve of each plot displays the empirical cdf's of a given signal plan. For each plot, the x-axis displays the average trip travel time (ATTT). For a given x value, the y-axis displays the proportion of simulation replications (out of the 50 replications) that have ATTT values smaller than x. Hence, the more a cdf curve is located to the left, the higher the proportion of small ATTT values; i.e., the better the performance of the corresponding signal plan. The solid thick curve corresponds to the empirical cdf of the initial signal plan (denoted x_0), the dashed curves (resp. solid thin curves) are the empirical cdf's of signal plans proposed by the traditional metamodel, i.e., the polynomial ϕ, (resp. the proposed metamodel, m).

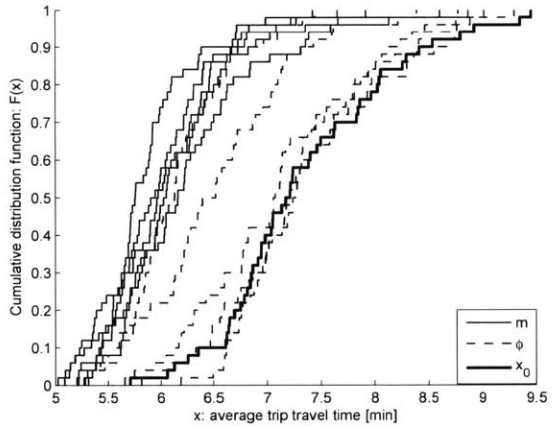
Figure 2-4(a) indicates that all five plans derived by both the proposed metamodel and the traditional metamodel yield improved performance when compared to the initial signal plan. All five plans derived by the proposed metamodel also have better performance compared to those proposed by the traditional metamodel.

Figure 2-4(b) indicates that all five signal plans derived by the proposed metamodel yield improved performance when compared to the initial plan. Four of them outperform all five plans derived by the traditional metamodel. Two of the signal plans derived by the traditional metamodel outperform the initial plan and the other three have similar performance as the initial plan.

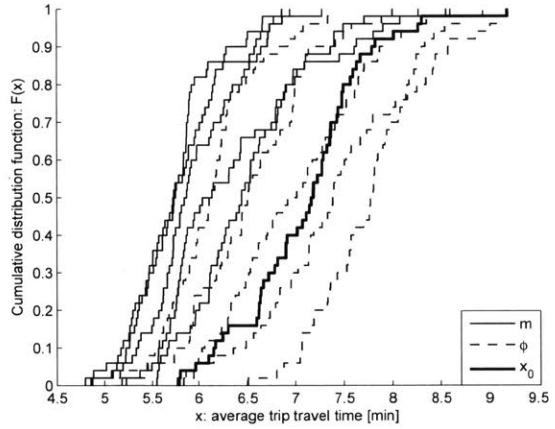
In Figure 2-4(c), all five plans derived by the proposed metamodel yield improve-



(a)



(b)



(c)

Figure 2-4: Empirical cdf's of the average trip travel times considering initial random signal plans and allowing for 150 simulation runs

ment compared to the initial plan, three of them outperform all five signal plans proposed by the traditional metamodel. Two of the signal plans proposed by the traditional metamodel have worse performance than the initial signal plan, one has similar performance and two have improved performance.

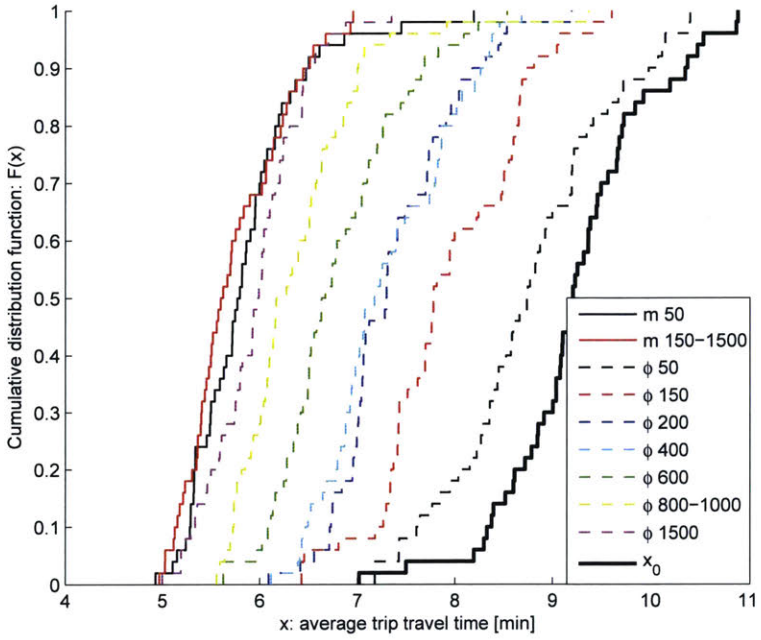
For all three initial points, the proposed method systematically derives signal plans with improved performance when compared to the initial plan, and most often, when compared to the plans obtained from the traditional metamodel. Additionally, the plans derived by the proposed method have good and very similar performance across all SO runs and all initial points, whereas the performance of the plans proposed by the traditional metamodel varies depending on both the initial point and the SO run. This illustrates the robustness of the proposed method to both initial points and to the stochasticity of the simulator.

We evaluate the performance of the proposed approach for larger sample sizes. We run the SO algorithm once, and allow for a total of 1500 simulation runs. We choose two random initial signal plans. We evaluate the performance of the signal plans proposed at sample sizes 50, 150, 200, 400, 600, 800, 1000 and 1500. We evaluate their performance just as before, i.e., for a given proposed plan we run 50 replications of the simulator and plot the empirical cdf (over these 50 replications) of the average trip travel times.

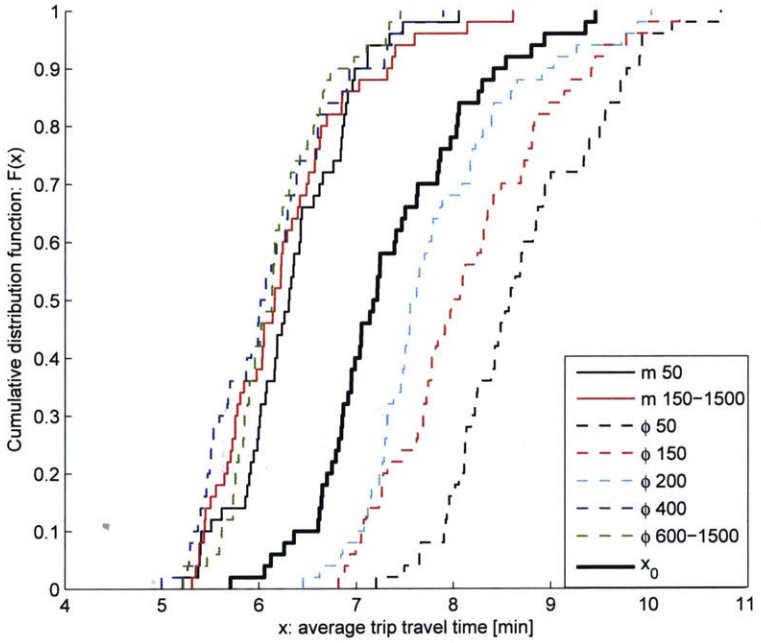
Figure 2-5(a) displays the corresponding cdf's of the initial signal plan used in Figure 2-4(a). The proposed approach identifies a signal plan with excellent performance already at sample size 50 (cdf labeled m 50). The signal plan identified as of sample size 150 remains the best up to sample size 1500. It has slightly improved performance, and in particular reduced variability, compared to that of sample size 50.

The performance of the signal plans proposed by the traditional metamodel (dashed curves) improves as the sample size increases. The traditional metamodel requires a much larger sample size to identify signal plans with good performance.

We carry out a paired t-test to evaluate whether the difference in performance of the signal plans proposed by each method at sample size 1500 is statistically signifi-



(a)



(b)

Figure 2-5: Empirical cdf's of the average trip travel times considering initial random signal plans and allowing for 1500 simulation runs

cant. We assume that the observed average trip travel times arise from a normal distribution with common but unknown variance. The null hypothesis assumes that the expected travel time is the same for both methods, whereas the alternative hypothesis assumes that they differ. The confidence level is 0.05, and there are 49 degrees of freedom. The sample average and sample standard deviation of our proposed signal plan (resp. that proposed by the polynomial metamodel) are 5.73 minutes and 0.51 minutes (resp. 5.95 minutes and 0.47 minutes). The critical value of the test is 1.96. The difference is statistically significant (t-statistic of -2.38, p-value of 0.02).

Thus, at sample size 1500 the proposed method still outperforms its traditional counterpart. That is, the signal plan identified by the proposed method as of sample size 150 outperforms that identified by the traditional method at sample size 1500.

Figure 2-5(b) displays the results considering the initial plan used in Figure 2-4(b). Similarly, the proposed approach identifies a signal plan with an excellent performance even at sample size 50. The signal plan with best performance derived by the proposed metamodel is obtained at sample size 150 and remains the same until sample size 1500. It has similar performance to that of sample size 50.

For sample sizes smaller than 400 the traditional metamodel yields signal plans with worse performance than the initial plan. Their performance significantly improve with increasing sample size until size 400. The performance of the derived signal plans with samples larger than 400 are similar. The signal plans proposed by the traditional metamodel method for sample sizes 600 to 1500 are the same.

We carry out the same paired t-test as before in order to evaluate whether the difference in performance of the signal plans proposed by each method at sample size 1500 is statistically significant. The sample average and sample standard deviation of our proposed signal plan (resp. that proposed by the polynomial metamodel) are 6.25 minutes and 0.73 minutes (resp. 6.16 minutes and 0.50 minutes). The difference is not statistically significant (t-statistic of 0.72, p-value of 0.48).

Figure 2-6 displays two instances of the Lausanne city map. The links are colored based on average link travel times (averaged over the 50 replications). The left (resp. right) map considers the average link travel times for the initial (resp. proposed)

signal plan. Here the proposed plan is that obtained with the initial plan and sample size of 150 of Figure 2-5(a). Green links have average travel times below 40 seconds, yellow links have travel times between 40 and 80 seconds, while red links have travel times greater than 80 seconds. This figure shows how the proposed plan yields city-wide travel time improvements.

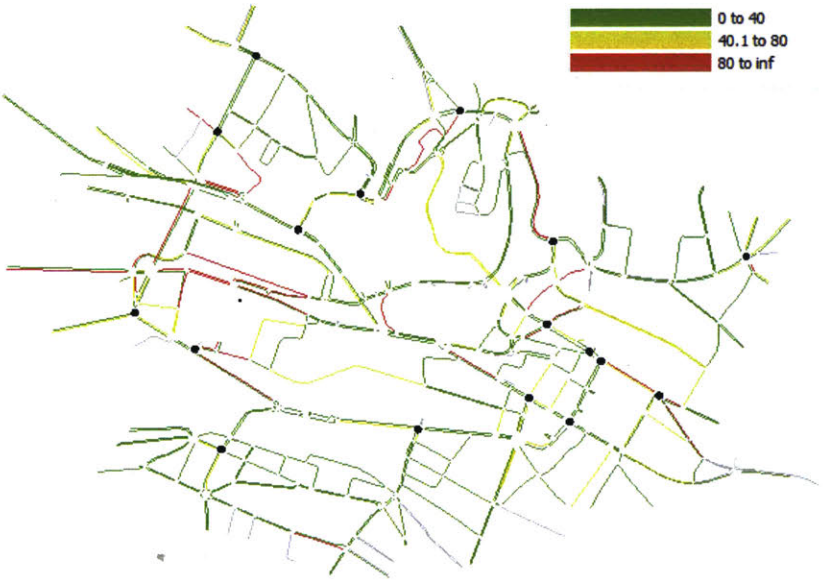
At each iteration of the SO algorithm, the two most computationally expensive tasks are the evaluation of the simulator as well as the solution of the trust-region subproblem (i.e., call of the fmincon routine). We consider the first initial plan (used in Figures 2-4(a) and 2-5(a)), and account for all 5 runs. Figure 2-7 displays the cdf of the simulation runs, and the TR subproblem runs. On average one simulation run takes 1.3 minutes, it takes 1.9 minutes to solve the TR subproblem. The experiments were run on a standard laptop (processor: 2.70GHz and 4 GB of RAM). Thus, the metamodel can be used to efficiently solve the TR subproblem at each iteration of the SO algorithm. Additionally, the structural information that it provides through the queueing network model allows the SO algorithm to identify signal plans with excellent performance under very tight computational budgets.

2.4.3 Comparison with a Signal Plan Derived by Commercial Signal Control Software

In this section, we compare the performance of the signal plans derived by our approach to those derived by the mainstream, commercial, and widely used, traffic signal control software Synchro (Trafficware; 2011, Synchro 8). Synchro is a traffic signal control optimization software based on a macroscopic, deterministic and local traffic model. It is widely used across the US (NYCDOT; 2012; Riniker et al.; 2009; Abdel-Rahim and Dixon; 2007; ATAC; 2003). For details on the split optimization technique within Synchro, we refer the reader to Chapter 14 of Trafficware (2011).

The Synchro version used does not allow for any fixed (i.e., exogenous) phase durations. Hence, we solve a signal control problem without fixed phases. For each intersection we take as cycle time its available (i.e., non-fixed) cycle time, $c_i - d_i$. The

(a)



(b)

Figure 2-6: Average link travel times using the initial signal plan (Figure 2-6(a)) and the signal plan proposed by the SO approach (Figure 2-6(b)). The averages (in seconds) are taken over 50 simulation replications.

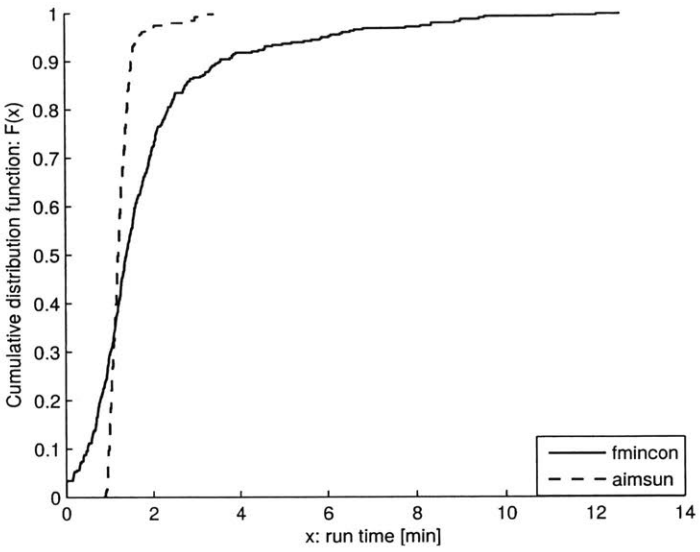


Figure 2-7: Simulation and trust region subproblem run times

problem formulation is given by Equations (2.18)-(2.20) and by replacing the right-hand side of Equation (2.19) by $(c_i - d_i)/(c_i - d_i)$, which equals 1. Synchro and our proposed SO method address this same problem. The corresponding TR subproblem is given by Equations (2.21)-(2.27), and replacing the right-hand side of (2.22) by 1 and the right-hand side of (2.24) by zero.

The Lausanne network is coded in Synchro. All signal plan information needed for Synchro (e.g., phase structure) is obtained from the existing Lausanne signal plan. The minimum splits are set to 4 seconds as in Section 2.4.1. Lane saturation flows (denoted s in Section 2.3.1) are set to 1800 vehicles per hour, following Swiss transportation norms. Synchro also needs, as inputs, estimates of prevailing movement flows. This was also needed when calibrating the analytical queueing model (e.g., to obtain turning probabilities). Hence, we use the same estimates as those provided to the queueing model. These are obtained from the simulator using the existing Lausanne signal plan.

To initialize the proposed SO approach, we consider the same three random initial signal plans as used in Figure 2-4. For each initial plan, we run the SO algorithm once, each time allowing for 150 simulation runs. To evaluate the performance of a

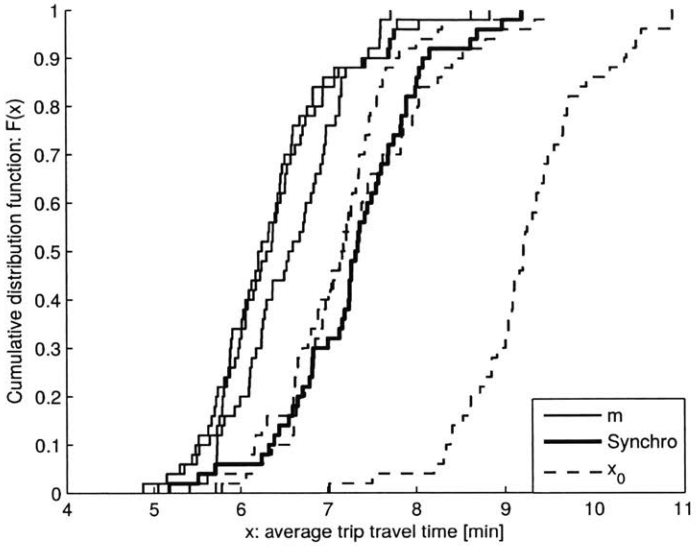


Figure 2-8: Empirical cdf's of the average trip travel times of the signal plans proposed by the SO approach and by Synchro.

plan, we use the simulator and proceed as described in Section 2.4.2.

Figure 2-8 presents the corresponding cdf curves. The three solid thin curves correspond to the plans derived by our proposed metamodel approach (denoted m). The dashed curves correspond to the three random initial signal plans (denoted x_0). The solid thick curve corresponds to the Synchro plan. All three plans derived by the purposed metamodel approach yield improved performance when compared to all three initial plans. All three plans derived by the SO approach also outperform the plan proposed by Synchro. The Synchro plan has similar performance to two of the three randomly drawn signal plans.

2.5 Conclusions

This chapter proposes a metamodel for large-scale simulation-based urban transportation optimization problems. It is a computationally efficient technique that identifies trial points (e.g., signal plans) with improved performance under tight computational budgets. This metamodel SO technique is based on the use of a tractable metamodel

that combines a general-purpose component (a quadratic polynomial) with a physical component (a tractable analytical stationary queueing network model).

We evaluate the performance of this approach by addressing a large-scale network-wide fixed-time signal control problem for the Swiss city of Lausanne. This problem considers a congested network (evening peak period demand) with an intricate topology. We compare the performance of the proposed metamodel to that of a traditional metamodel. The proposed method identifies signal plans that improve the distribution of average trip travel times compared to both the initial signal plans, and most often, to the signal plans derived by the traditional method. This network-wide signal control problem is considered high-dimensional for SO algorithms, for derivative-free algorithms as well as for signal control algorithms. We also compare the performance of the proposed approach to that of a widely-used signal control software, Synchro. All proposed signal plans outperform the plan derived by Synchro.

In this chapter, random uniformly drawn signal plans are used as initial points for the SO algorithm. The results illustrate the robustness of the proposed metamodel method to initial points. This allows practitioners to use the method to address a variety of signal control problems without requiring any field-knowledge to initialize the method.

As part of ongoing research, we are investigating the use of the proposed method to address a variety of generally constrained simulation-based transportation problems, including microscopic model calibration, multi-modal traffic management, and multi-modal network design problems.

We also investigate novel analytical traffic model formulations with increased accuracy. The model used in this chapter is a stationary model. The next step of this work is a time-dependent formulation based on the use of transient finite capacity queueing theory, which will be introduced in Chapter 3.

We are also developing SO algorithms with improved short-term performance by using information from analytical traffic models, such as the queueing network model used in this chapter, to inform both sampling strategies and statistical tests.

Chapter 3

A Metamodel Simulation-based Optimization Algorithm for Large-scale Dynamic Urban Transportation Problems

This chapter presents a novel computationally efficient optimization algorithm that addresses simulation-based dynamic urban transportation problems with time-dependent decision variables. In the previous chapter, we proposed a metamodel SO algorithm that used an analytical stationary traffic model. In this chapter, we propose a metamodel SO algorithm that formulates and uses an analytical transient traffic model. The method of Chapter 3 has been published as: Chong, L. and Osorio, C. (forthcoming). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems, Transportation Science. Forthcoming.

3.1 Introduction

In the field of urban transportation, dynamic optimization problems, i.e., optimization problems with time-dependent decision vectors, have been addressed through the use of analytical dynamic and, mostly deterministic, traffic models. Such models

are based on an aggregate, i.e., low-resolution, description of traffic dynamics. They are computationally efficient to evaluate, yet lack a detailed description of heterogeneous traveler behavior, of vehicle-infrastructure interactions, and thus of intricate traffic dynamics observed in urban areas. A detailed description of these dynamics is provided by a family of high-resolution simulation-based traffic models, known as stochastic microscopic or mesoscopic traffic simulators. Nonetheless, these simulators are computationally inefficient to evaluate. Hence, their use to address optimization problems has been limited. To the best of our knowledge, they have not been used to address dynamic transportation optimization problems; let alone large-scale dynamic problems. This chapter proposes a methodology that enables high-resolution traffic simulators to be used, in a computationally efficient way, to address large-scale dynamic transportation optimization problems.

This chapter focuses on optimization problems of the following form:

$$\min_{x_1,\ldots,x_L} f(x,z;p) = \frac{1}{L}\sum_{\ell=1}^{L} E[F_\ell(x_\ell, z_\ell; p)] \tag{3.1}$$

$$g_\ell(x_\ell; p) = 0 \quad \forall \ell \in \mathcal{L}. \tag{3.2}$$

The time horizon is decomposed into a set of L disjoint time intervals $\mathcal{L}$. Each time interval ℓ considers a continuous decision vector x_ℓ (e.g., traffic signal plan), an objective function defined as the expectation of a network performance function F_ℓ (e.g., trip travel time, network throughput within interval ℓ). The latter depends on a vector of interval-dependent decision variables x_ℓ and endogenous variables z_ℓ (e.g., link travel times, traffic assignment), and a vector of interval-independent exogenous parameters p (e.g., network topology). The decision vector for all time intervals is denoted $x = (x_1, \ldots, x_L)$, similarly we denote $z = (z_1, \ldots, z_L)$. For time interval ℓ, the feasible region is defined by a set of general analytical and differentiable constraints, g_ℓ. This precludes the use of simulation-based constraints (constraints that need to be evaluated via simulation). This is why the function g_ℓ does not depend on the endogenous simulation variables z_ℓ. A discussion on problems with simulation-

based constraints is given in the conclusions of this chapter (Section 3.5). Note that Constraint (3.2) is a general formulation for any type of constraint, i.e., inequality constraints can be transformed and expressed as equality constraints of the form (3.2). To summarize, Problem (3.1)-(3.2) considers a time-dependent decision vector with a simulation-based objective function and general, analytical, differentiable constraints. Hereafter, Problem (3.1)-(3.2) is referred to as a dynamic simulation-based optimization (SO) problem.

The focus of this chapter is to propose computationally efficient algorithms for large-scale dynamic simulation-based problems, i.e., algorithms that can identify solutions with significantly improved objective function values within a tight computational budget (e.g., few simulation runs).

The remainder of this section reviews past work on addressing dynamic SO problems, not limited to transportation applications. We conclude this section by stating the main contributions of this chapter.

3.1.1 Literature Review

The field of supply chain logistics has extensively used detailed stochastic simulators to describe intricate spatial-temporal processes within supply chain networks. Schwartz et al. (2006) and Jung et al. (2004) both consider a dynamic inventory management problem and resort to the use of gradient-based SO algorithms. In both cases, the simulator is seen as a black-box. It is used to obtain objective function and first-order derivative estimates. Nonetheless, no problem-specific analytical structural information is provided to the optimization algorithm. Legato et al. (2008) address a dynamic quay crane scheduling problem at a maritime container terminal. The simulator, a stochastic queueing network model, is also used as a black-box to derive objective function estimates. An approach that addresses a dynamic supply chain problem, and that indeed attempts to exploit problem-specific structure is proposed by Almeder et al. (2009). Although it is not an SO approach, it is worth mentioning because it is also motivated by the ideas of: (i) combining efficient optimization techniques with computationally costly simulation models; and (ii) exploiting problem-specific struc-

tural information. In the framework of Almeder et al. (2009): the following two steps are carried out iteratively: (i) certain parameters of the analytical problems (linear programs and mixed-integer programs) are estimated via simulation; (ii) given the estimated parameters, the analytical problems are solved. The iterations are carried out until the distance between consecutive solutions is below a threshold.

To the best of our knowledge, in past work in the field of dynamic SO, the stochastic simulator has been seen as a black-box. It has been coupled with general-purpose algorithms. This allows for flexibility since the proposed frameworks can be readily extended to address a variety of problems. Nonetheless, the proposed methods are not designed to address problems within tight computational budget. In other words, the SO algorithms used are designed to achieve asymptotic performance guarantees rather than good short-term performance. When using high-resolution computationally expensive simulation models, the simulation run time significantly limits the scale and complexity of the problems that can be addressed. Jung et al. (2004), for instance, clearly state this limitation: "The key limitation of the overall approach lies in the large computing times required to address problems of increasing scope."

In the field of transportation, few SO methods that embed high-resolution simulators have been developed (see Chapter 2 Section 2.1.1 for a review). However, these methods are designed to achieve suitable asymptotic properties (e.g., convergence properties), rather than to identify points with good performance within few simulation runs. In other words, they are not designed to be computationally efficient. In addition, most of these methods focus on problems with time-independent variables. However, in the field of traffic management, optimization problems with time-dependent variables are common, such as time-dependent traffic control optimization problems and dynamic toll optimization problems. Therefore, there is a need to use develop computationally efficient SO algorithms to address these problems.

Contributions

This chapter proposes an SO algorithm for large-scale networks with high-dimensional time-dependent decision variables, i.e., we propose an SO algorithm for large-scale dynamic transportation problems. The proposed approach is suitable to address a variety of transportation problems that can be formulated as large-scale dynamic continuous simulation-based optimization problems with general analytical constraints.

- We propose a framework to address dynamic SO transportation problems of the form (3.1)-(3.2). The framework couples information from the simulator with analytical time-dependent problem-specific structural information. More specifically, a time-dependent analytical traffic model is formulated and used to derive an analytical description of the spatial-temporal congestion patterns observed in the simulator. This analytical information is provided to the SO algorithm. This coupling of information is achieved through the use of metamodel methods. This combination leads to SO algorithms that are computationally efficient, i.e., they can identify solutions with good performance within few simulation runs.

- To the best of our knowledge, this is the first SO algorithm designed for dynamic problems. It is also the first to enable dynamic transportation SO problems to be addressed in a computationally efficient manner. Efficiency is achieved through the formulation of a tractable transient analytical network model.

- The analytical network model is formulated as a simple system of equations. The model complexity is linear in the number of links in the network and is independent of the link space capacities. This makes it particularly suitable for large-scale networks.

- Our past work has developed efficient SO algorithms for problems with time-independent decision variables. Appendix B.1 summarizes the main methods, results and insights of past work. It serves to motivate the ideas of this chapter. This chapter is the first to design an efficient algorithm suitable for SO problems

with time-dependent decision variables. In particular, in past work the analytical traffic models used are stationary models. They provide a description of the spatial propagation of congestion, yet do not describe its temporal propagation. The proposed model is a transient model. It describes both the spatial and the temporal propagation of congestion. More specifically, it approximates the temporal variations of the spillback probabilities of each lane. The use of a transient, rather than a stationary, model is recommended for scenarios where congestion varies substantially within each time period (e.g., congestion build-up or dissipation periods). The case study of this chapter, indicates that providing the SO algorithm with a temporal description of congestion propagation enables it to identify solutions that delay the onset and the propagation of congestion. The proposed analytical model builds upon the stationary model of Chapter 2. A description of their main differences is given in Section 3.2.3.

- The proposed algorithm is used to address a time-dependent traffic signal control problem. This problem controls the signal plans of 17 intersections distributed across a city with over 600 roads. This is considered large-scale for urban traffic signal control problems (Aboudolas et al.; 2010, 2007; Dinopoulou et al.; 2006). This problem is a constrained non-convex problem with a decision vector of dimension 198; this is also considered a challenging and large-scale problem in the field of SO. The case study indicates that the proposed method identifies signal plans that outperform: (i) a signal plan prevailing in the field, (ii) a signal plan derived by a commercial signal control software, and (iii) signal plans derived by the SO method of Chapter 2, which is designed for time-independent problems.

We present the proposed dynamic SO framework in Section 3.2. We then apply the framework to address a traffic signal control problem for the city of Lausanne. The optimization problem is formulated in Section 3.3, the Lausanne city case study results are presented in Section 3.4. The main conclusions are presented in Section 3.5.

3.2 Methodology

This chapter proposes a new metamodel formulation for SO problems with time-dependent variables. This new formulation is then embedded within the SO algorithm of Osorio and Bierlaire (2013) and is used to address a time-dependent signal control problem. In Section 3.2.1, we summarize the main ideas of the metamodel SO algorithm. Section 3.2.2 describes the proposed model. A summary of the methodology is given in Section 3.2.3.

3.2.1 Metamodel SO Framework

The main idea of the metamodel SO algorithm is to approximate the unknown simulation-based objective function (Equation (3.1)) with an analytical function known as the metamodel. The chapter uses the same metamodel framework as described in Chapter 2. As is detailed in Appendix B.1, past metamodel algorithms have performed well for high-dimensional transportation problems with tight computational budgets. In this section, we briefly describe the key points of the metamodel SO algorithm. We use the same notations as Chapter 2 to describe the algorithm. For more details regarding this algorithm, we refer the reader to Chapter 2, Section 2.2.

The metamodel SO algorithm is an iterative method. At a given iteration k, the functional form of the metamodel is given as follows:

$$m_k(x, z; p, \beta_k) = \beta_{k,0} f_A(x, z; p) + \phi(x; \beta_{k,1}, \ldots, \beta_{k,D}). \tag{3.3}$$

As mentioned in Chapter 2, Section 2.2.1, this metamodel consists a problem-specific component f_A and a general-purpose component ϕ. In Chapter 2, f_A represents the approximation of the objective function by the stationary macroscopic traffic network model described in Section 2.2.3. In this chapter, we propose a transient network model to represent the approximation of the objective function f_A.

The problem solved at a given iteration k of the SO algorithm is of the form:

$$\min_{x_1,\ldots,x_L} m_k(x, z; p, \beta_k) \tag{3.4}$$

$$g_\ell(x_\ell; p) = 0 \quad \forall \ell \in \mathcal{L} \tag{3.5}$$

$$h(x, z; p) = 0. \tag{3.6}$$

Metamodel m_k of Equation (3.4) is the analytical approximation of the simulation-based objective function described in Equation (3.1). Constraints (3.5) are equivalent to the analytical constraints of the problem, i.e., Constraints (3.2). Constraint (3.6) represents the proposed traffic model that is used to derive the physical metamodel component (i.e., term f_A of Equation (3.3)).

The proposed traffic model used in this chapter is a transient network model, that is also analytical, differentiable and computationally efficient to evaluate. The transient network model combines ideas from transient queueing theory, queueing network theory and traffic flow theory. It is formulated as a system of nonlinear equations. The model complexity is linear in the number of links in the network and is independent of the individual link space capacities. Hence, it is a scalable model.

The proposed transient network model extends the high-scalable stationary network model we proposed in Chapter 2, Section 2.2.3. In the transient network model, traffic dynamics are described by combining transient queueing theoretic ideas inspired from the works of Morse (1958), Cohen (1982) and Odoni and Roth (1983). These ideas are described in Section 3.2.2. Recently, link models that are both based on transient queueing theory and are fully consistent with traditional deterministic traffic flow theoretic link models have been proposed (e.g., Osorio et al.; 2011; Osorio and Flötteröd; 2014). Their extension to full network models is a topic of ongoing research.

3.2.2 Transient Network Model

The proposed transient model builds upon the stationary model formulated in Chapter 2, Section 2.2.3. The latter model combines ideas from finite capacity queueing network theory, traffic flow theory and various national transportation norms.

The stationary model uses time-independent endogenous variables and parameters. It does not provide any temporal information, and is therefore not suitable to address dynamic optimization problems. In this chapter, we propose a transient network model. This model is then used to approximate the physical component of the metamodel, i.e., f_A of Equation (3.3). The transient metamodel is used in Section 3.4 to address a dynamic SO problem.

This section formulates the transient network model. We discretize the time horizon of interest into a set $\mathcal{L}$ of disjoint equal-length time intervals. In this section, we present the model formulation for a given time interval ℓ, $\ell \in \mathcal{L}$.

Section 3.2.2 defines, for interval ℓ, a set of endogenous queueing variables. Section 3.2.2 and 3.2.2 describe how these variables are used to derive time-dependent spillback probabilities, which describe traffic dynamics throughout the network.

Interval-Specific Queueing Variables

The proposed transient model extends the stationary model proposed in Chapter 2, Section 2.2.3 by accounting for the temporal variations of the spillback probability. For a given time interval ℓ and queue i, we use the following notations to describe

the model:

$\hat{\lambda}_{i,\ell}$	effective arrival rate;
$\hat{\rho}_{i,\ell}$	effective traffic intensity;
$\mu_{i,\ell}$	service rate;
$P_\ell(N_i = k_i)$	stationary spillback probability;
γ_i	external arrival rate;
k_i	upper bound of the queue length;
N_i	total number of vehicles in queue i;
p_{ij}	transition probability from queue i to queue j;
$\mathscr{D}_i$	set of downstream queues of queue i.

In this model, $\hat{\lambda}_{i,\ell}$, $\hat{\rho}_{i,\ell}$, $\mu_{i,\ell}$ and $P_\ell(N_i = k_i)$ are interval-specific variables. They are defined by solving the following system of equations:

$$\begin{cases} \hat{\lambda}_{i,\ell} = \gamma_i(1 - P_\ell(N_i = k_i)) + \sum_j p_{ji}\hat{\lambda}_{j,\ell} & \text{(3.7a)} \\ \hat{\rho}_{i,\ell} = \dfrac{\hat{\lambda}_{i,\ell}}{\mu_{i,\ell}} + \left(\sum_{j\in\mathscr{D}_i} p_{ij}P_\ell(N_j = k_j)\right)\left(\sum_{j\in\mathscr{D}_i}\hat{\rho}_{j,\ell}\right) & \text{(3.7b)} \\ P_\ell(N_i = k_i) = \dfrac{1-\hat{\rho}_{i,\ell}}{1-\hat{\rho}_{i,\ell}^{k_i+1}}\hat{\rho}_{i,\ell}^{k_i}. & \text{(3.7c)} \end{cases}$$

This system of equations is the interval-specific version of the System of Equations (2.12) (in Chapter 2, Section 2.2.3). It assumes that the exogenous parameters (γ_i, p_{ij} and k_i) do not change across time intervals. This assumption can be easily relaxed. By solving the above system of equations, we obtain interval-specific endogenous variables for each queue: $\hat{\lambda}_{i,\ell}$, $\hat{\rho}_{i,\ell}$ and $P_\ell(N_i = k_i)$. These variables may vary from one time interval to the next, yet are assumed constant within a time interval. This assumption significantly reduces model complexity and preserves model tractability.

In the case study of this chapter, the decision vector consists of the green times of the signalized lanes. There is a one-to-one mapping between the total green time of a

lane and the service rate of the corresponding queue (denoted $\mu_{i,\ell}$). Given a specific decision vector value (and hence a specific set of $\mu_{i,\ell}$ values), the above System of Equations (3.7) can be solved simultaneously for all queues, yet independently for each time interval. Therefore, given a decision vector value, a set of L decoupled systems of equations can be solved to obtain the endogenous variables for all time intervals. The formulation of these variables as decoupled systems of equations contributes to the tractability and scalability of the proposed formulation.

Consider a network with a total of n queues. For a given decision vector value and a given time-interval ℓ, the System of Equations (3.7) consists of a total of $3n$ variables, $3n$ equations: n linear (3.7a), n quadratic (3.7b) and n non-quadratic convex (3.7c). The system can therefore be solved efficiently. For any feasible set of demand and supply parameters, i.e., $\{\gamma \geq 0, \mu \geq 0\}$, the system contains at least one solution. In particular, the use of finite capacity queues ensures that for any positive value of the traffic intensity (i.e., the ratio of the expected demand to the expected supply) of each queue, there exists a stationary regime for the network of queues, and hence the stationary probabilities are well-defined. If we had resorted to the use of infinite capacity queues, then the traffic intensities would need to be strictly smaller than 1 to ensure stationarity.

Observations from Existing Transient Queueing Models

The goal is to describe the temporal variations of the spillback probabilities. Such time-dependent probabilities are referred to in queueing theory as transient probabilities. In the field of transportation, models based on transient queueing theory have focused on infinite capacity queues: Heidemann (2001); Peterson et al. (1995) and Odoni and Roth (1983). More broadly, in the field of queueing network theory, research has focused mostly on: (i) networks with infinite capacity queues, and (ii) the analysis of the stationary regime. This is, arguably, because between-queue (i.e., spatial) dependencies are intricate to describe analytically, let alone their temporal variations.

The analytical transient analysis of a single isolated finite capacity queue is presented in the seminal work of Morse (1958) and Cohen (1982). The formulation of the proposed transient network model builds upon ideas from these two works.

Morse (1958) considers an isolated M/M/1/k queue, with fixed arrival rate λ, service rate μ, traffic intensity $\rho = \lambda/\mu$, and a given queue-length distribution at the beginning of a time interval (i.e., initial conditions). The latter is called the initial queue-length distribution. We denote the beginning of the time interval by t_0. Morse (1958, Equation (6.13)) derives an exact closed-form expression for the transient queue-length distribution. More specifically, t time units after t_0, the probability of observing a queue of length m is given by:

$$\begin{cases} P(N = m, t) = P(N = m) + \dots \\ \cdots + \rho^{\frac{m}{2}} \sum_{s=1}^{k} C_s \left[\sin\left(\frac{sm\pi}{k+1}\right) - \sqrt{\rho} \sin\left(\frac{s(m+1)\pi}{k+1}\right) \right] e^{-w_s t} & \text{(3.8a)} \\ w_s = \lambda + \mu - 2\sqrt{\lambda\mu} \cos\left(\frac{s\pi}{k+1}\right). & \text{(3.8b)} \end{cases}$$

The probability that the queue is of length m at time t is denoted $P(N = m, t)$, which is also known as the transient probability. The corresponding stationary probability is denoted $P(N = m)$. The coefficients C_s are determined by solving a linear system of equations that ensure initial boundary conditions:

$$P(N = m, 0) = P^0(N = m), \tag{3.9}$$

where $P^0(N = m)$ denotes the given initial conditions, i.e., the probability that the queue is of length m at time t_0.

The System of Equations (3.8) could be used within a network setting in order to approximate the marginal queue-length distribution of each queue in a network. The main challenge of such an approach is that in order to compute the coefficients C_s, the full queue-length distribution of each queue would need to be computed. In other words, for each queue i, a set of $k_i + 1$ probabilities would need to be computed, and this for every time interval. This would lead to a model complexity that depends on

the space capacity, k_i, of each queue. For instance, in a network with n queues, and a problem with L time intervals, the total number of probabilities to approximate would be $\sum_{i=1}^{n}(k_i+1)L$. Such an approach would not scale well for large-scale urban networks.

The factor $1/w_s$ of Equation (3.8a) is known in queueing theory as the *relaxation time*. It is the time needed for a given performance metric to reach its stationary value. Equation (3.8a) states that the transient queue length distribution converges exponentially to the stationary distribution. Seminal papers that have studied the relaxation time of an isolated infinite capacity queue include Cohen (1982) and Odoni and Roth (1983), where the exponential decay term is written as $e^{-t/\tilde{\tau}}$, and $\tilde{\tau}$ is the relaxation time. Cohen (1982) considers an isolated M/M/1 queue and proposes:

$$\tilde{\tau} = 1/(\mu(1-\sqrt{\rho})^2). \tag{3.10}$$

Odoni and Roth (1983) propose an approximation of $\tilde{\tau}$ for an isolated G/G/1 queue. For an M/M/1 queue, their approximation is similar to that of Cohen (1982) and is given by: $\tilde{\tau} = 2/(2.8\mu(1-\sqrt{\rho})^2)$.

These approximations share the following properties, which will be preserved in our proposed relaxation time approximation.

- The relaxation time increases as congestion increases (for an infinite capacity queue the stationary state is only defined if $\rho < 1$, and increasing congestion corresponds to $\rho \to 1$).
- For a fixed traffic intensity ρ, the relaxation time should be proportional to the time units of the queueing system parameters. In other words, it should be inversely proportional to either the arrival or the service rates. For example, in the above approximations, $\tilde{\tau}$ is proportional to $1/\mu$.

Transient Queueing Model

This section formulates a transient queueing model that preserves the following prop-

erties of the stationary queueing model described in Chapter 2, Section 2.2.3.

- The focus is on the approximation of the transient spillback probabilities. In other words, for each time interval ℓ and each queue i, our objective is to approximate $P_\ell(N_i = k_i, t)$ rather than the full distribution. This leads to a model complexity in the order of nL (instead of $\sum_{i=1}^{n}(k_i+1)L$). The model complexity is linear in the number of queues, and more importantly, is independent of the space capacities.

- The between-queue dependencies are captured through the queueing variables $(\hat{\lambda}, \hat{\rho})$. Given these queueing variables, the spillback probability of a given queue does not depend on any information from other queues. These variables describe, respectively, the expected demand and the ratio of expected demand to expected supply. They capture problem structure, and are therefore referred to as structural variables.

- The structural variables of the queues can be derived by solving a simple system of equations.

Consider time interval ℓ that begins at time t_ℓ and a given queue i. The spillback probability t time units after t_ℓ is approximated by:

$$\begin{cases} P_\ell(N_i = k_i, t) = P_\ell(N_i = k_i) + (P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))e^{-\frac{t}{\tau_{i,\ell}}} & \text{(3.11a)} \\ \tau_{i,\ell} = \dfrac{c\hat{\rho}_{i,\ell}k_i}{\hat{\lambda}_{i,\ell}(1-\sqrt{\hat{\rho}_{i,\ell}})^2}. & \text{(3.11b)} \end{cases}$$

Equation (3.11a) is inspired from Equation (3.8a) in that the transient probability of a queue is defined as the sum of its stationary probability (term $P_\ell(N_i = k_i)$) and a term that decays exponentially with time. The stationary probability is defined by the System of Equations (3.7).

Equation (3.11b) is inspired from Equation (3.10) in that the relaxation time is: (i) directly proportional to $1/(1-\sqrt{\hat{\rho}})^2$, and (ii) proportional to the service rate term given by $\hat{\rho}/\hat{\lambda}$. Equation (3.11b) is inspired from (3.8b) in that the relaxation time depends on the space capacity k. Note that the works of Cohen (1982) and of

Odoni and Roth (1983) consider infinite capacity queues, hence their relaxation time approximations do not depend on space capacity. In Equation (3.11b), the term c is an exogenous scaling parameter, that is fitted based on traffic simulation outputs.

For any set of feasible initial conditions, (i.e., $0 \leq P_\ell(N_i = k_i, t_\ell) \leq 1$) System (3.11) converges asymptotically to $P_\ell(N_i = k_i)$. Convergence is guaranteed for any positive value of the traffic intensity (even for values larger than 1).

3.2.3 Methodology Summary

Let us summarize the proposed methodology. A dynamic extension of the metamodel SO framework of Chapter 2 is used. The metamodel is defined by Equation (3.3). The key to developing a computationally efficient SO algorithm lies in the formulation of an analytical and tractable problem-specific approximation (denoted f_A in Equation (3.3)) of the objective function (denoted f in Equation (3.1)). This chapter proposes a transient queueing network model that yields a tractable approximation of f_A. The model considers, for each time interval ℓ, a set of endogenous queueing model variables defined by the System of Equations (3.7). These variables approximate the between-queue dependencies, e.g., how spillback at a given queue impacts the performance of upstream queues. Given this set of variables, the spillback probability of each queue varies across time, within time interval ℓ, following Equation (3.11). The transient queueing network model is then used to derive the functional form of f_A. An example of the derivation of an expression for f_A is given in Section 3.3.2 for a traffic signal control problem.

An algorithmic summary of the transient network model is given in Algorithm 1. For a network with L time intervals and n queues, the number of endogenous variables is $3nL$. In other words, for each queue and each time interval, the endogenous variables are: $\hat{\lambda}_{i,\ell}$, $\hat{\rho}_{i,\ell}$, and $P_\ell(N_i = k_i)$.

We now summarize the main differences between the proposed transient and the stationary analytical network model of Chapter 2, Section 2.2.3. The stationary model yields stationary (hence, time-independent) lane spillback probabilities, while time-dependent probabilities are derived by the transient queueing model. Thus,

Algorithm 1 Algorithm to evaluate the transient network model

Carry out each of the following steps for all queues i before proceeding to the next step.

Steps:

1. Define the start of each time interval $t_1, t_2, \ldots, t_L$.
2. Define the exogenous parameters $\gamma_i, k_i, p_{ij}, \mu_{i,\ell}$.
3. Define the initial condition of each queue: $P(N_i = k_i, t_1)$.
4. Repeat the following for time intervals $\ell = 1, 2, \ldots, L$
   (a) Solve the System of Equations (3.7) to obtain $\hat{\lambda}_{i,\ell}, \hat{\rho}_{i,\ell}$ and $P_\ell(N_i = k_i)$.
   (b) Compute the spillback probabilities at the end of the time interval according to (3.11) with $t = t_{\ell+1}$.

the proposed model provides a temporal description of congestion propagation. The queueing variables that describe demand and supply (e.g., arrival rates, traffic intensities) are time-independent for the stationary model, they are constant for the entire time horizon. The transient model uses a set of variables for each time period, this allows to describe temporal changes in demand and supply.

The proposed analytical model builds upon the stationary model of Chapter 2, Section 2.2.3. For a network with n lanes and a set of L time intervals, the stationary model consists of a system of $3n$ equations with $3n$ endogenous variables, while the transient model consists of a system of L systems of equations that are solved sequentially and each have a dimension $3n$. The transient model consists of $3nL$ endogenous variables. The complexity of the proposed model scales linearly with the number of time intervals. Thus, it is less tractable than the stationary model. The complexity of both models is linear in n and is independent of the link space capacities. This makes both of them suitable for large-scale network analysis.

3.3 Time-dependent Traffic Signal Control Problem

The proposed method is suitable to address a variety of simulation-based dynamic transportation problems. In this section, we illustrate the computational efficiency of the methodology by considering a large-scale traffic signal control problem with time-dependent decision variables. Section 3.3.1 formulates the traffic control problem. Section 3.3.2 presents the analytical expression for f_A (of Equation (3.3)) for this specific problem. Section 3.3.3 discusses implementation details.

3.3.1 Optimization Problem Formulation

A detailed review of traffic signal control terminology is given in Appendix A of Osorio (2010) or in Lin (2011). We consider fixed-time (also called time of day or pre-timed) signal control plans. They are determined offline. Unlike traffic-responsive strategies, they do not respond to prevailing real-time traffic conditions. Congested networks with complex traffic dynamics (e.g., grid topology, congested multi-modal traffic) often resort to the use of fixed-time plans (Chen et al.; 2015).

We divide the time horizon of interest (e.g., evening peak period) into L time intervals. For each time interval, we determine a fixed-time signal plan. The signal plans for all intersections and all L time intervals are determined jointly.

This chapter focuses on the optimization of green splits, i.e., the decision variables are the green splits of the signal controlled lanes. Cycle lengths and offsets are fixed. All other signal plan variables (e.g., stage structure) are also assumed fixed.

To formulate this problem we introduce the following notation:

c_i	cycle time of intersection i;
d_i	fixed cycle time of intersection i;
x_ℓ	vector of green splits for time interval ℓ;
$x_\ell(j)$	green split of signal phase j for time interval ℓ;
x_{LB}	vector of lower bounds for green splits;
$\mathscr{I}$	set of intersection indices;
$\mathscr{P}_I(i)$	set of endogenous signal phase indices of intersection i.

The problem is formulated as follows:

$$\min_{x_1,\ldots,x_L} f(x,z;p) = \frac{1}{L}\sum_{\ell=1}^{L} E[F_\ell(x_\ell, z_\ell; p)] \tag{3.12}$$

subject to

$$\sum_{j\in\mathscr{P}_I(i)} x_\ell(j) = \frac{c_i - d_i}{c_i},\ \forall i \in \mathscr{I}, \ell \in \mathcal{L} \tag{3.13}$$

$$x_\ell \geq x_{LB},\ \forall \ell \in \mathcal{L}. \tag{3.14}$$

The decision vector x consists of the green splits of all signal phases in all L time intervals. The objective is to minimize the expected trip travel time, where $E[F_\ell(x_\ell, z_\ell; p)]$ represents the expected trip travel time during time interval ℓ.

The linear constraints (3.13) ensure that, for each intersection, the sum of the green times for each signal phase is equal to the available (i.e., non-fixed) cycle time. Equation (3.14) ensures lower bounds for the green splits. These bounds may vary according to the city, the time horizon of interest, and even the intersection.

3.3.2 Derivation of the Analytical Objective Function, f_A

Recall that the transient network model of Section 3.2 is used to derive the analytical approximation (f_A of Equation (3.3)) of the simulation-based objective function (f

of Equation (3.12)). We now derive the analytical expression for f_A for the specific objective function (3.12). More specifically, we derive the analytical approximation of the term $E[F_\ell(x_\ell, z_\ell; p)]$ in Equation (3.12). Let $f_{A,\ell}$ denote this approximation.

For time interval ℓ, we can derive the expected time in the network per user by applying Little's law (Little; 1961) to the entire road network. This leads to:

$$f_{A,\ell} = \frac{\sum_i E_\ell[N_i]}{\frac{1}{t_{\ell+1}-t_\ell}\int_{t_\ell}^{t_{\ell+1}} \sum_i \gamma_i (1 - P_\ell(N_i = k_i, t))dt}, \tag{3.15}$$

where $E_\ell[N_i]$ represents the expected number of vehicles in queue i during time interval ℓ, t_ℓ denotes the start time of time interval ℓ, and $t_{\ell+1}$ denotes the end time of time interval ℓ. The numerator is the expected number of vehicles in the network during time interval ℓ. The denominator is the effective external arrival rate to the network during time interval ℓ. In the denominator, the term within the integral represents the instantaneous effective external arrival rate to queue i at time t. The external arrival rate γ_i is an exogenous parameter, and the transient spillback probability $P_\ell(N_i = k_i, t)$ is given by Equation (3.11a).

We now derive the closed-form expression used to approximate the numerator of Equation (3.15), we then derive a closed-form expression for the denominator. The closed-form expression for $E_\ell[N_i]$ of the numerator is derived as follows. For an isolated queue i of the type M/M/1/k_i, with traffic intensity ρ_i, the stationary expected number of vehicles is given by:

$$E[N_i] = \rho_i\left(\frac{1}{1-\rho_i} - \frac{(k_i+1){\rho_i}^{k_i}}{1-{\rho_i}^{k_i+1}}\right). \tag{3.16}$$

An analytical derivation of Equation (3.16) is given in Appendix A.1. We assume this functional form holds within a given time interval, i.e., we use the following approximation:

$$E_\ell[N_i] = \rho_{i,\ell}\left(\frac{1}{1-\rho_{i,\ell}} - \frac{(k_i+1){\rho_{i,\ell}}^{k_i}}{1-{\rho_{i,\ell}}^{k_i+1}}\right). \tag{3.17}$$

An analytical expression for $\rho_{i,\ell}$ is obtained as follows. The model we proposed in

Chapter 2, Section 2.2.3 is formulated based on the effective traffic intensity $\hat{\rho}_i$, rather than the traffic intensity ρ_i of a queue. The effective traffic intensity is related to the traffic intensity ρ_i as follows: $\rho_i = \hat{\rho}_i/(1 - P(N_i = k_i))$. We therefore approximate the traffic intensity of queue i during time-interval ℓ by:

$$\rho_{i,\ell} = \frac{\hat{\rho}_{i,\ell}}{\frac{1}{t_{\ell+1}-t_\ell}\int_{t_\ell}^{t_{\ell+1}}(1 - P_\ell(N_i = k_i, t))dt}, \tag{3.18}$$

where $\hat{\rho}_{i,\ell}$ is defined by Equation (3.7b). A closed-form expression for the integral in the denominator of Equation (3.18) is obtained as follows. We insert the expression for $P_\ell(N_i = k_i, t)$ given by Equation (3.11a) to obtain:

$$A = \int_{t_\ell}^{t_{\ell+1}} (1 - P_\ell(N_i = k_i, t))\, dt \tag{3.19}$$

$$= \int_{t_\ell}^{t_{\ell+1}} \left(1 - \left[P_\ell(N_i = k_i) + (P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))\, e^{-\frac{t}{\tau_{i,\ell}}}\right]\right) dt \tag{3.20}$$

$$= \int_{t_\ell}^{t_{\ell+1}} (1 - P_\ell(N_i = k_i))\, dt - (P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i)) \int_{t_\ell}^{t_{\ell+1}} e^{-\frac{t}{\tau_{i,\ell}}} dt \tag{3.21}$$

$$= (t_{\ell+1} - t_\ell)(1 - P_\ell(N_i = k_i)) + \ldots$$

$$\cdots + \tau_{i,\ell}(P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))(e^{-\frac{t_{\ell+1}}{\tau_{i,\ell}}} - e^{-\frac{t_\ell}{\tau_{i,\ell}}}). \tag{3.22}$$

In summary, the term $E_\ell[N_i]$ of the numerator of Equation (3.15) is given by Equations (3.17), (3.18) and (3.22).

The denominator of Equation (3.15) can be rewritten by interchanging the summation with the integral to obtain:

$$B = \frac{1}{t_{\ell+1} - t_\ell} \int_{t_\ell}^{t_{\ell+1}} \sum_i \gamma_i (1 - P_\ell(N_i = k_i, t))dt \tag{3.23}$$

$$= \frac{1}{t_{\ell+1} - t_\ell} \sum_i \gamma_i \int_{t_\ell}^{t_{\ell+1}} (1 - P_\ell(N_i = k_i, t))dt. \tag{3.24}$$

A closed-form expression for the integral of Equation (3.24) is given by Equation (3.22).

Note that the analytical expressions derived above allow us to approximate the expected trip travel time (i.e., the objective function) based on the knowledge of spillback probabilities $P_\ell(N_i = k_i, t)$ rather than the knowledge of full queue-length distributions. This contributes to the tractability and scalability of the proposed approach. This leads to a model complexity that is linear in the number of queues and that is independent of the space capacity of each queue.

3.3.3 Implementation Notes

We implement the lower bound constraints (3.14) as nonlinear equality constraints by introducing a new variable v and implementing:

$$x_\ell = x_{LB} + v_\ell^2. \tag{3.25}$$

In addition, we enforce the positivity of the endogenous variables $\hat{\rho}_{i,\ell}$ by introducing a new variable $u_{i,\ell}$ and adding the equalities:

$$\hat{\rho}_{i,\ell} = u_{i,\ell}^2. \tag{3.26}$$

We do not enforce the positivity of the other endogenous variables, rather we check a posteriori that all other endogenous variables are positive. In our numerous experiments, all solutions to the system of equations obtained by the solver have consisted of positive values.

Note that the green splits are related to the service rate of the underlying queue i through the following equation:

$$\mu_{i,\ell} = \left(e_i + \sum_{j \in \mathscr{P}_I(i)} x_\ell(j) \right) s, \tag{3.27}$$

where $\mathscr{P}_I(i)$ represents the set of endogenous phase indices of the lane represented by queue i, e_i is the ratio of fixed green time to cycle time of signalized queue i, and s is the saturation flow rate. We assume a common saturation flow for all signalized

lanes. For each signalized queue, Equation (3.27) is inserted into Equation (3.7b), in order to implement both constraints as a single constraint.

In order to further enhance tractability, for the large-scale case study of Section 3.4, we approximate the arrival rate to each queue (denoted $\hat{\lambda}_{i,\ell}$) as exogenous; i.e., it does not vary with the decision vector values. The exogenous value is obtained by considering the prevailing fixed-time signal plan of the city for the whole time horizon, this yields a set of $\mu_{i,\ell}$ values (through Equation (3.27)). Then the System of Equations (3.7) is solved, and the corresponding $\hat{\lambda}_{i,\ell}$ values obtained are used as fixed values throughout the optimization. This simplification enhances model tractability. Nonetheless, the assumption of arrival patterns to the links independent of the signal plans may lead to a misestimation of the link spillback probabilities. Appendix C derives, as an example, the expression of the derivative of the objective function with regard to an endogenous variable.

For a problem with L time intervals, n lanes (where each lane is modeled as a single queue), where we determine r endogenous signal phases at a total of o signalized intersections per time interval, our implementation leads to a total of $(2n + r)L$ endogenous variables. These consist of two endogenous queueing variables per queue per time interval ($u_{i,\ell}$ and $P_\ell(N_i = k_i)$), and one green split variable ($v_l(j)$) for each signal phase. The corresponding optimization problem (i.e., a trust region subproblem) solved at every iteration of the SO algorithm consists of $(2n + o)L$ nonlinear equality constraints and one nonlinear inequality constraint (which is the trust region constraint). Of the nonlinear equality constraints, $2nL$ correspond to Equations (3.7b) and (3.7c), and oL correspond to Equation (3.13) (the latter becomes nonlinear since v_ℓ is implemented instead of x_ℓ). The trust region (TR) subproblem, that is solved at every iteration, is a variation of the signal control problem formulated in Section 3.3.1. The detailed formulation of the TR subproblem is described in Appendix B.2.

3.4 Empirical Analysis: Lausanne City Case Study

This section addresses a traffic signal control problem for the city of Lausanne, Switzerland. Section 3.4.1 describes the network. Section 3.4.2 benchmarks the proposed transient metamodel SO method against the stationary metamodel SO method proposed in Chapter 2. Sections 3.4.3 and 3.4.4, respectively, compare the performance of a signal plan derived by the proposed method to that of an existing signal plan for the city of Lausanne, and to that of a signal plan derived by a mainstream commercial signal control software.

3.4.1 Lausanne City Network

We evaluate the performance of the proposed SO algorithm by considering a large-scale signal control problem for the entire Swiss city of Lausanne. The city map is displayed in Chapter 2, Figure 2-2, the considered area is delimited in white. We use a microscopic traffic simulation model of the Lausanne city developed by Dumont and Bert (2006). It is implemented with the Aimsun simulator (TSS; 2011), and is calibrated for evening peak period demand. The modeled road network is displayed in Figure 2-3. The road network consists of 603 links and 231 intersections.

We consider the first hour of the evening peak period: 5-6 pm. During this hour, congestion gradually builds up. Hence, it is important to design a signal plan that accounts for this temporal propagation of congestion. We use the proposed algorithm to determine one signal plan for 5-5:30 pm and a second signal plan for 5:30-6 pm. In other words, we decompose the hour into two 30-minute intervals, and determine a signal plan for each of the two intervals.

The signals of 17 intersections are controlled in this case study. These 17 intersections are depicted as filled squares in Figure 2-3. The signal control problem has a total of 198 endogenous signal phase variables (99 signal phases per time interval), i.e., the dimension of the decision vector is 198. The phase variable is defined as the ratio of green time (i.e., the total duration of a phase) to cycle time.

The transient queueing model of this network consists of 902 queues. The trust

Table 3.1: Traffic models used by each of the compared SO methods.

	Microscopic	Macroscopic	
	simulation-based	analytical stationary	analytical transient
Dynamic SO	✓		✓
Stationary SO	✓	✓	

region (TR) subproblem solved at every iteration of the SO algorithm consists of 3806 endogenous variables with 3642 nonlinear equality constraints, and one trust region inequality. The lower bounds of the green splits (Equation (3.14)) are set to 4 seconds according to the Swiss transportation norm (VSS; 1992).

3.4.2 Comparison of the Dynamic SO Method with the Stationary SO Method

In order to benchmark the performance of the dynamic SO (DSO) method, we compare its performance to that of an SO method that has been successfully used to address large-scale SO problems. We benchmark the performance of the dynamic SO method against the performance of the simulation-based stationary metamodel method (SSO) proposed in Chapter 2. Both methods consider a metamodel defined by Equation (3.3). They differ only in the physical component of the metamodel (f_A of Equation (3.3)). The proposed DSO method considers the transient network model formulated in Section 3.2.2. The SSO method considers the stationary network model defined by the System of Equations (2.12) described in Chapter 2, Section 2.2.3. All other algorithmic details and parameters are identical in both methods. The difference between these two methods is also described in Table 3.1. This comparison allows us to evaluate and quantify the added value of using transient analytical information in the metamodel (i.e., the added value of using a time-dependent network model to derive f_A of Equation (3.3)).

For both methods, we consider a tight computational budget, which is defined as a maximum of 100 simulation runs that can be carried out. In other words, the SO algorithm is initialized with no simulation observations available, and it stops

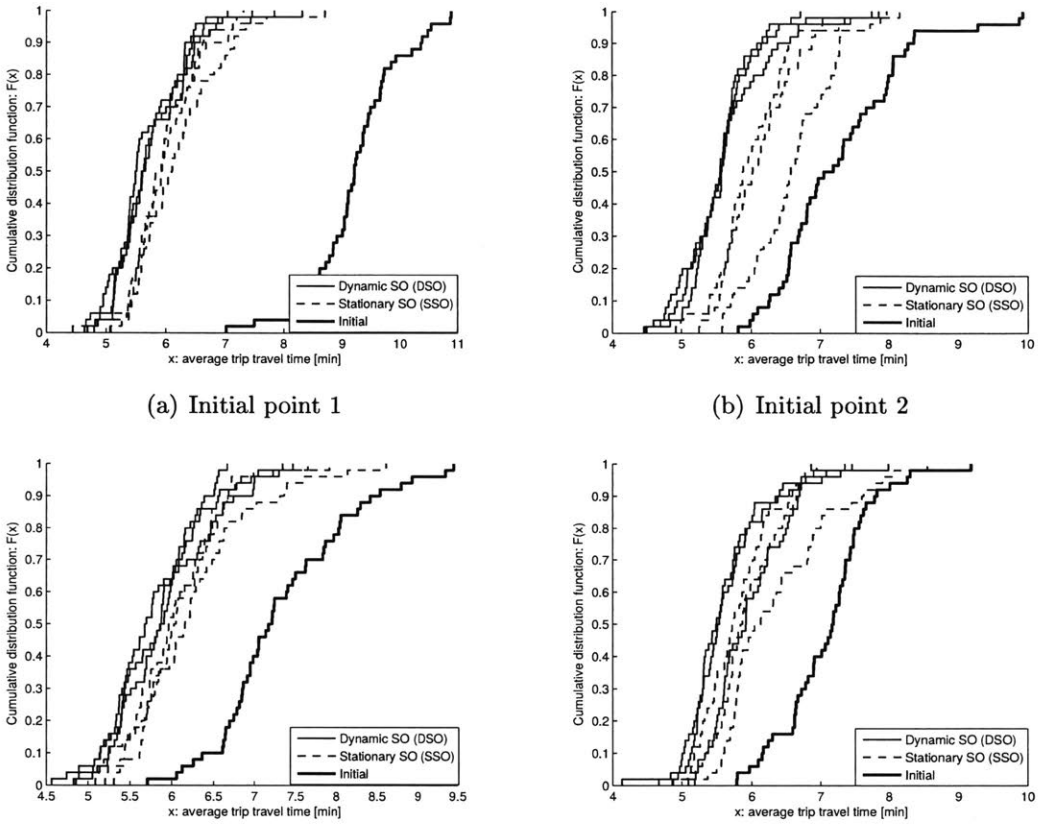
once a total of 100 simulation runs have been carried out. We refer the reader to Chapter 2 for more information about the SSO algorithm, and for a comparison of its performance to that of a traditional SO algorithm.

We consider four different initial points (i.e., initial signal plans) to initialize the SO algorithms. These points are uniformly randomly drawn from the feasible space defined by Equations (3.13) and (3.14). Thus, these points are uniformly sampled, subject to linear equalities and non-negativity constraints. This uniform sampling is carried out according to the code of Stafford (2006). For each initial point, we run each SO method (i.e., SSO and DSO) three times, each time allowing for a total of 100 simulation runs. Thus, for each method and each initial point, we derive three *proposed* signal plans. In order to evaluate the performance of a proposed signal plan, we embed it within the traffic simulator and run 50 simulation replications. We then compare the performance of the proposed signal plans both with statistical tests, and by comparing the cumulative distribution function (cdf) of the objective function realizations (i.e., the average trip travel times) obtained from these 50 simulation replications.

Each plot of Figure 3-1 considers a different initial point. Each curve of each plot displays the cdf of a given signal plan. For each plot, the x-axis displays the average trip travel time (ATTT). For a given x value, the y-axis displays the proportion of simulation replications (out of the 50 replications) that have ATTT values smaller than x. Hence, the more a cdf curve is located to the left, the higher the proportion of small ATTT values; i.e., the better the performance of the corresponding signal plan.

For each plot, the solid thick curve corresponds to the cdf of the initial signal plan, the solid thin curves are the cdf's of signal plans proposed by DSO and the dashed curves are the cdf's of signal plans proposed by SSO.

For all four initial points (Figures 3-1(a)-3-1(d)), all three plans derived by both DSO and SSO yield improved performance when compared to the initial signal plan. For three of the four initial points (Figures 3-1(a)-3-1(c)), all three plans derived by DSO outperform all three plans derived by SSO. For Figure 3-1(d), two out of



(a) Initial point 1

(b) Initial point 2

(c) Initial point 3

(d) Initial point 4

Figure 3-1: Cumulative distribution functions of the average travel times considering different initial signal plans.

the three DSO plans outperform all three SSO plans. The third DSO plan performs similarly to two of the SSO plans. It outperforms the third plan proposed by SSO.

In summary, for all four initial points, DSO systematically derives signal plans with improved performance when compared to the initial plan, and most often, when compared to the plans derived by SSO.

We study the robustness of the DSO solutions to the initial points. Figure 3-2 displays the cdf's of the 12 solutions derived by DSO (solid thin curves) and all 4 initial points (solid thick curves). In other words, all curves of all four plots of Figure 3-1 are displayed here in a single plot in Figure 3-2. The plot shows that: (i) the DSO solutions systematically outperform the initial solutions, (ii) all DSO solutions have similar performance. The DSO plans have good and consistent performance across all

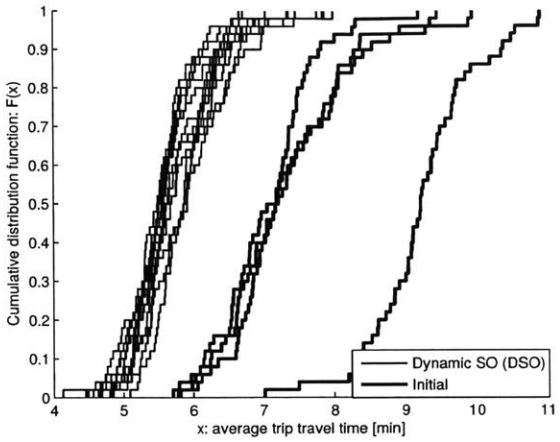


Figure 3-2: Cumulative distribution functions of the average travel times for all 4 initial points and all 12 proposed solutions

SO runs and all initial points. This illustrates the robustness of the proposed method to both the initial points and to the stochasticity of the simulator.

In order to test whether the performance of DSO is statistically significantly better than that of SSO, we carry out, for each initial point, a one-sided paired t-test. We choose a performance metric that accounts for the overall performance of an SO method over all three SO runs. The 50 simulation replications used to derive each of the cdf curves of Figure 3-1 use the same 50 random seeds. Hence, we use a statistic that aggregates the performance of a given SO method for a given random seed. For a given SO method, let X_{ij} denote the average travel time obtained under the j^{th} run ($j \in \{1, 2, 3\}$) and the i^{th} simulation replication seed ($i \in \{1, 2, \ldots, 50\}$). The considered performance metric is defined as:

$$Y_i = \frac{1}{3} \sum_{j=1}^{3} X_{ij}, \ \forall i \in \{1, 2, \ldots, 50\}. \tag{3.28}$$

We treat Y_i as the *average* algorithmic performance of an SO method (DSO or SSO) under replication i.

We use a paired one-sided t-test that assumes that the simulation observations are independent and arise from a normal distribution with common but unknown

Table 3.2: Paired one-sided t-test results that compare the performance of DSO and of SSO

Initial point	T-statistic	P-value	Average	Standard deviation
1	-4.23	$5e-5$	-0.35	0.58
2	-9.31	$1e-12$	-0.65	0.49
3	-3.92	$1e-4$	-0.28	0.51
4	-4.01	$1e-4$	-0.26	0.46

variance. We pair the observations with common random replication seeds. The null hypothesis assumes equal expected trip travel times for both DSO and SSO (i.e., equal expected value of Y for each method). The alternative hypothesis assumes that the expectation of the DSO method is lower than that of the SSO method.

The test significance level is 0.05. It has 49 degrees of freedom. The corresponding critical value is -1.677. Table 3.2 summarizes the test statistics. Each row of the table displays the result of the t-test for a given initial point (i.e., one test for each plot of Figure 3-1). Columns 1 to 5 display, respectively, the initial point index, the t-statistic, the p-value, the average paired difference and the standard deviation of the paired differences.

All t-statistics (Column 2) are smaller than -1.677, hence the null hypothesis of all four tests is rejected. In other words, for each initial point, the signal plans derived by DSO lead to average travel times that are statistically significantly lower than those of the signal plans derived by SSO.

Performance under Increasing Congestion Levels

We now analyze how the performance of the proposed signal plans varies over time. The traffic simulation considers the first hour of peak period traffic (5-6 pm). Over this hour, congestion gradually increases (for more details regarding the temporal evolution of congestion in this network, see Osorio (2010, Chap. 4)). This temporal analysis allows us to understand how the proposed signal plans perform under increasingly congested conditions.

For a given signal plan, we estimate the expected trip travel time in ten minute

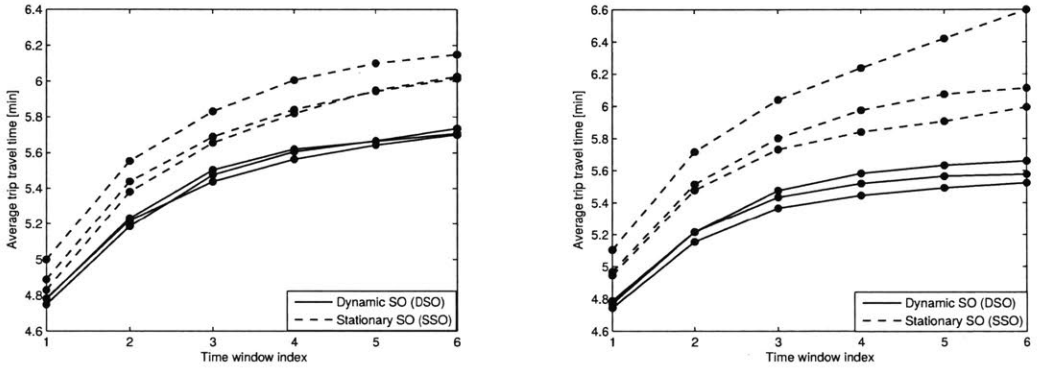
increments. In other words, we consider 6 time windows indexed, respectively, 1 through 6. The corresponding time windows are 5-5:10, 5-5:20, 5-5:30, 5-5:40, 5-5:50 and 5-6 pm.

In Figure 3-3, each plot displays the results considering the same initial plans and the same *proposed* signal plans as in Figure 3-1. The x-axis corresponds to the time window index. The y-axis represents the average trip travel time during the corresponding time window. These are averages obtained over the same 50 simulation replications used in the previous analysis. Each of the solid curves corresponds to one of the three signal plans proposed by DSO. Each of the dashed curves corresponds to one of the three signal plans proposed by SSO. In each curve, the dots represent the average travel time during the corresponding time windows. The curves are interpolated from the dots.

Note that the average travel time estimate for the last time window (time window 6) is the trip travel time averaged over all trips from 5-6 pm. This corresponds to an estimate of the objective function of the optimization problem. This estimate also equals the average of all 50 simulation replications points used to construct a given cdf curve of Figure 3-1.

In Figures 3-3(a), 3-3(b) and 3-3(c), all three plans derived by DSO have better performance compared to the three plans of SSO throughout all six time windows. This holds for two of the three plans derived by DSO in Figure 3-3(d). In Figure 3-3(d), the third plan derived by DSO has worse performance than the three SSO plans for the first three time windows, and worse performance than two of the SSO plans for the remaining three time windows. For all DSO plans of Figures 3-3(a), 3-3(b) and 3-3(c), their performance seems stable for congested conditions (time windows 4-6). Overall, the difference in performance between the DSO plans and the SSO plans seems to increase with increasing levels of congestion. This illustrates again the added value of using an analytical model that is time-dependent, such that it describes this temporal evolution of congestion within the time horizon of interest (5-6 pm).

In order to test the statistical significance of the difference in performance over



(a) Initial point 1

(b) Initial point 2

(c) Initial point 3

(d) Initial point 4

Figure 3-3: Time-dependent average trip travel times for different initial signal plans

Table 3.3: Paired one-sided t-tests that compare the time-dependent performance of DSO and of SSO

(a) T-statistics

Initial point \ Time window index	1	2	3	4	5	6
1	-9.64	-8.95	-5.11	-4.36	-4.36	-4.23
2	-16.31	-13.42	-10.84	-9.56	-9.28	-9.31
3	-9.78	-7.97	-5.37	-4.33	-4.11	-3.92
4	0.38	-1.55	-1.83	-2.59	-3.58	-4.01

(b) P-values

Initial point \ Time window index	1	2	3	4	5	6
1	$3e-13$	$3e-12$	$3e-6$	$3e-5$	$3e-5$	$5e-5$
2	$9e-22$	$2e-18$	$6e-15$	$4e-13$	$1e-12$	$1e-12$
3	$2e-13$	$1e-10$	$1e-6$	$4e-5$	$7e-5$	$1e-4$
4	0.65	0.063	0.036	0.006	$4e-4$	$1e-4$

(c) Average paired differences

Initial point \ Time window index	1	2	3	4	5	6
1	-0.14	-0.25	-0.25	-0.29	-0.34	-0.35
2	-0.24	-0.37	-0.43	-0.50	-0.57	-0.65
3	-0.13	-0.21	-0.21	-0.23	-0.25	-0.28
4	-0.01	-0.04	-0.07	-0.13	-0.21	-0.26

(d) Standard deviation of paired differences

Initial point \ Time window index	1	2	3	4	5	6
1	0.1	0.19	0.35	0.47	0.55	0.58
2	0.1	0.2	0.28	0.37	0.43	0.49
3	0.1	0.19	0.28	0.37	0.44	0.51
4	0.11	0.18	0.25	0.35	0.42	0.46

time, we proceed as before: we carry out for each initial point and each time window a paired one-sided t-test. For each t-test, the metric we use is the *average* algorithmic performance of an SO method (DSO or SSO) during the corresponding time window.

Table 3.3 contains four subtables (a)-(d) that display, respectively, the corresponding t-test statistics, the p-values, the average paired difference and the standard deviation of the paired differences. For a given table, a row corresponds to an initial point, a column corresponds to a time window. As before, the test significance level is 0.05, it has 49 degrees of freedom, and the critical t-value is -1.677.

Table 3.3(b) (or equivalently Table 3.3(a)) shows that for initial point 1, 2 and

3, the null hypothesis of equal expectation for DSO and SSO is rejected for initial points 1, 2 and 3, at all time windows. This means that as congestion increases (i.e., as time advances from 5 pm to 6 pm), the DSO signal plans consistently outperform the SSO plans. For initial point 4, the null hypothesis is rejected for time windows 3-6, and not rejected for time windows 1 and 2. This means that for initial point 4 at low levels of congestion (i.e., the first 20 minutes of the peak hour), the DSO plans do not outperform the SSO plans. Additionally, Table 3.3(c) shows that for a given initial point (i.e., a given row), the average difference increases with the time window index. This shows that the difference in performance between DSO and SSO increases as congestion increases.

Computational Runtime of DSO

The steps of the DSO algorithm that are the most computationally demanding are: (i) evaluating the simulator, and (ii) solving the trust region (TR) subproblem. Details on the formulation and numerical solver used to solve the TR subproblem are given in Appendix B.2. To illustrate the computational runtimes for each of these steps, we consider the three SO runs of the DSO method carried out with initial point 2. Each of the three SO runs allows for 100 simulation evaluations. We use a standard laptop with an Inter(R) Core(TM) i7-2960XM 2.7 GHz processor and 8GB RAM. The average runtime for one simulation replication is 1.2 minutes with a standard deviation of 0.2 minutes. The average time to solve the TR subproblem is 5.5 minutes with a standard deviation of 3.6 minutes. These runtimes are suitable for solving the problem offline. In the future, we would like to improve to runtime of both steps, such that they can be used to address real-time SO problems. For instance, we currently use the standard Matlab routine for nonlinear constrained problems (*fmincon*) to solve the TR subproblem. The use of a standard TR method would reduce the computational runtime for solving the TR subproblem. For real-time SO methods, the main runtime constraint remains the number of simulation runs that can be carried out in real-time.

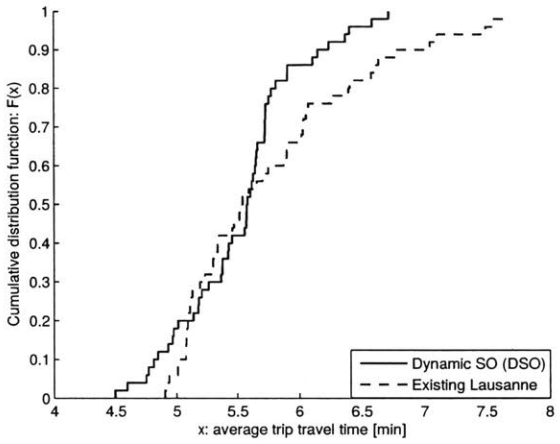


Figure 3-4: Comparison of the performance of a DSO signal plan and an existing signal plan of the city of Lausanne

3.4.3 Comparison with an Existing Signal Plan of the City of Lausanne

We now compare the performance of the best signal plan derived by DSO with that of an existing signal plan for the city of Lausanne. The best DSO signal plan is defined as the one (among the 12 DSO signal plans analyzed in the previous section) with the lowest average travel time over the 50 simulation replications. This corresponds to the left-most cdf curve of Figure 3-1(b), or equivalently the signal plan with the smallest y-value at time window 6 of Figure 3-3(b). Figure 3-4 displays the cdf of this DSO plan (solid line) and of the Lausanne plan (dashed line). The DSO plan outperforms the Lausanne plan. In order to test whether these differences are statistically significant, we carry out a paired one-sided t-test as before. The t-test has, once again, a significance level of 0.05, 49 degrees of freedom, and a critical value of -1.677. The average trip travel time (average over all 50 simulation replications) of the DSO plan is 5.52 minutes, and that of Lausanne signal plan is 5.77 minutes. The average paired difference is 0.25, the corresponding standard deviation is 0.94. This leads to a t-statistic of -1.83, and a p-value of 0.037. The hull hypothesis is rejected. Therefore, the DSO approach can derive signal plans that perform significantly better than the Lausanne plan.

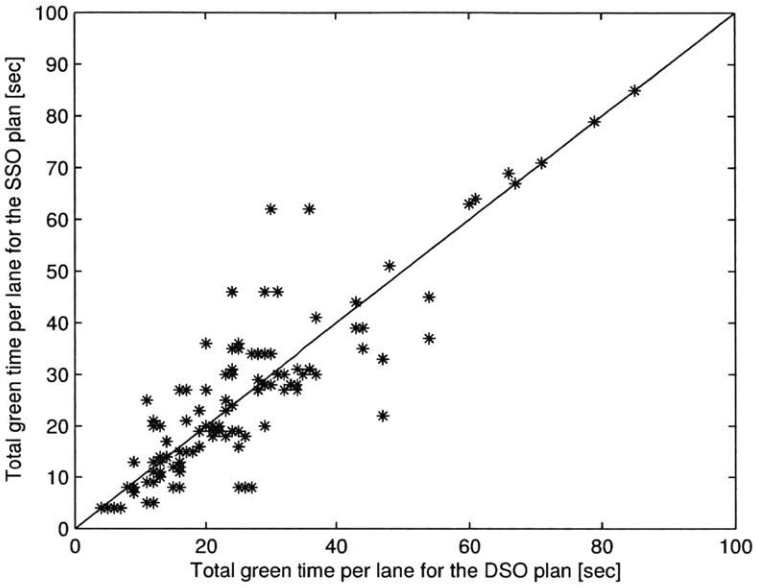
We now compare the values of the signal plans of the best DSO plan, the best SSO plan and the current Lausanne plan. The best DSO plan is defined as that with the lowest average travel time over the 50 simulation replications. This corresponds to the left-most cdf curve of Figure 3-1(b). Similarly, the best SSO plan is the left-most dashed cdf curve in Figure 3-1(d).

The x-axis of Figure 3-5(a) displays, for each signal controlled lane at a time interval, the total green time (in seconds), under the DSO plan. Since the DSO plan yields two different plans for the two intervals, Figure 3-5(a) displays one point for each of these two time intervals. The y-axis displays the total green time under the SSO plan. The diagonal line $y = x$ is also plotted. The points close to the diagonal line indicate lanes that have similar green time values under both plans. Similarly, Figure 3-5(b) displays the green times for the DSO plan (x-axis) and the Lausanne plan (y-axis). The plots indicate that there are many lanes with significantly different green times.

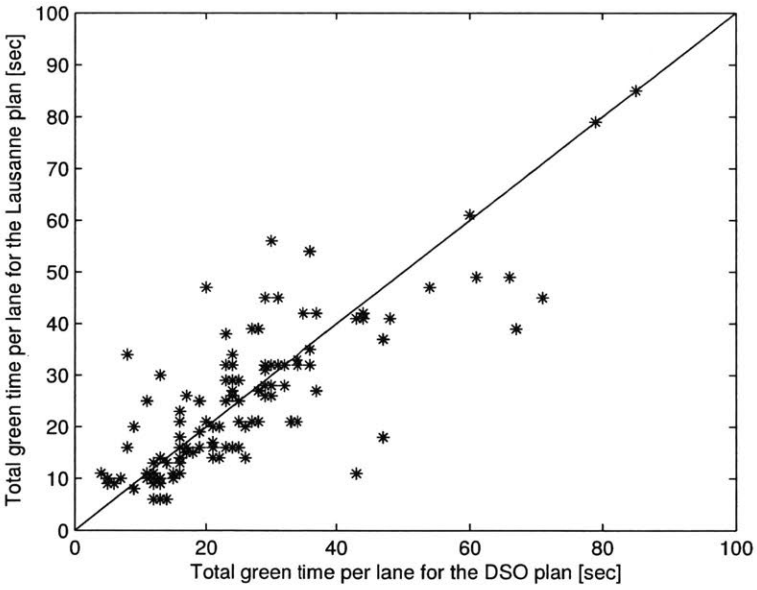
We have also studied the variations of the green times over time for the DSO plan, but have not found any interesting temporal trends.

Figure 3-6(a) displays, for each of these 3 signal plans, the average trip travel time as a function of time. Figure 3-6(b) displays the average link density of the 60 controlled links. For both plots, the x-axis corresponds to a time window index, and for each estimate, the confidence intervals (obtained from the 50 simulation replications) are displayed. The trip travel time metric of Figure 3-6(a) is defined just as that of Figure (3-3). Notice that the objective function corresponds to the average travel time estimated at time interval 6 (i.e., it is the average travel time from 5-6pm). Figure 3-6(b) displays the average link density of the 60 controlled links. For time intervals 1 through 6, this average is computed during time 5:00-5:10, 5:10-5:20, 5:20-5:30, 5:30-5:40, 5:40-5:50 and 5:50-6:00, respectively. This figure illustrates the impact of the signal plans on local (link-level) performance.

Figure 3-6(a) indicates that as congestion increases, so does the difference in performance between the DSO plan and the two other plans. Figure 3-6(b) indicates that the main difference between the DSO plan and the two other plans is that the



(a) Comparison of the best DSO and the best SSO signal plans



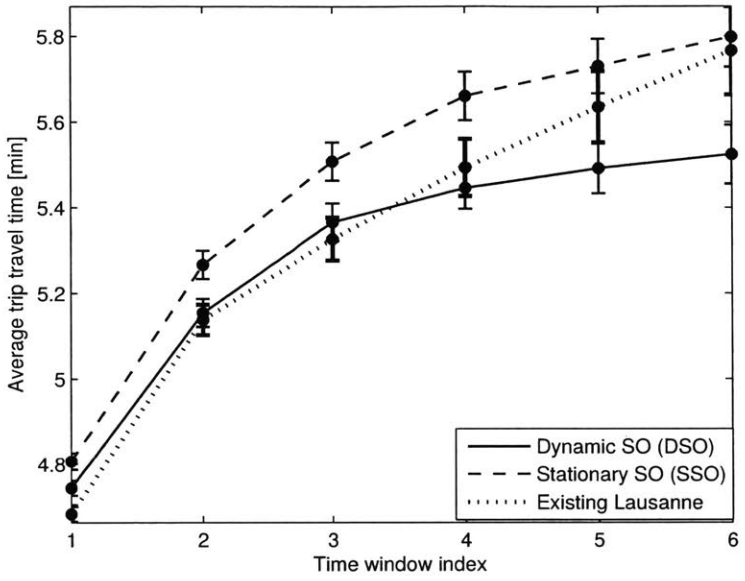
(b) Comparison of the best DSO and the Lausanne plans

Figure 3-5: Total green time (in seconds) per signalized lane for the best DSO, the best SSO and the Lausanne signal plans

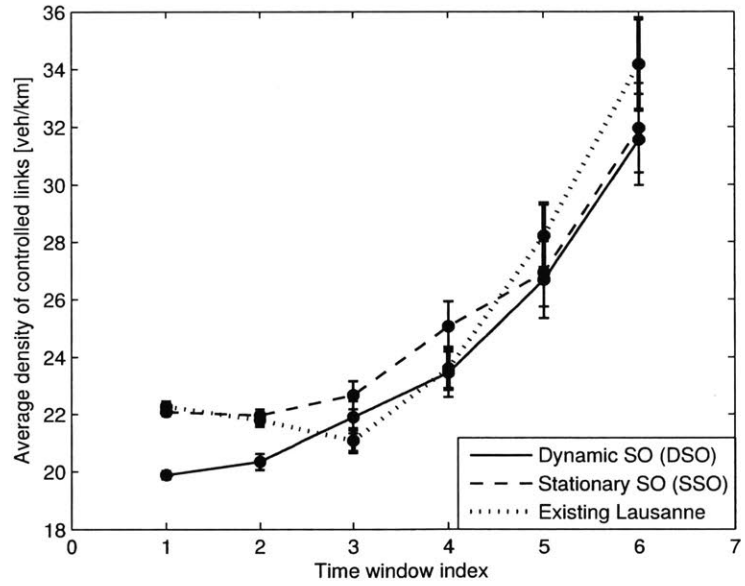
DSO plan leads to significantly lower levels of congestion at the start of the peak-period. As a consequence, it delays the onset and the propagation of congestion. This is observed in Figure 3-6(a), where the travel times at 6pm under the DSO plan are those observed under the Lausanne plan around 5:30pm, and under the SSO plan around 5:40 pm. Figures 3-6(a) and 3-6(b) indicate that as congestion increases, so does the variance of the estimators. This increased variance illustrates one of the challenges of performing SO for congested scenarios, where the objective function estimators tend to have high variance.

3.4.4 Comparison with a Signal Plan Derived by Commercial Signal Control Software

We compare the performance of the best DSO signal plan with that of a signal plan derived with the signal control software Synchro, which is a mainstream, commercial and popular signal control software (Trafficware (2011)). It is widely used across the United States. Major cities, such as New York, rely on it to design their signal plans (NYCDOT; 2012). For details on Synchro's green split optimization technique, we refer the reader to Chapter 14 of Trafficware (2011). Synchro is based on a macroscopic, deterministic and local traffic model. We give Synchro the same input data (e.g., network and traffic data) as for the DSO method. The details on the Synchro input configuration used are given in Chapter 2, Section 2.4.3. As before, the best DSO signal plan is that with the lowest average trip travel time among the 12 plans derived by DSO (i.e., left-most cdf curve of Figure 3-1(b)). To evaluate the performance of the Synchro and the DSO signal plans, we proceed as in Section 3.4.3. Figure 3-7(a) displays the cdf of the average trip travel time of the DSO signal plan (solid curve) and of the Synchro plan (dashed curve). The DSO plan yields a significant improvement in the average trip travel times. The average objective function value, among the 50 simulation replications, is 5.5 minutes for the DSO plan and 7.3 minutes for the Synchro plan. The DSO plan yields a 25% reduction in the trip travel times compared to the Synchro plan.



(a) Time-dependent average trip travel time



(b) Time-dependent average link density of the signal controlled links

Figure 3-6: Time-dependent congestion metrics of the best DSO, the best SSO and the Lausanne plans

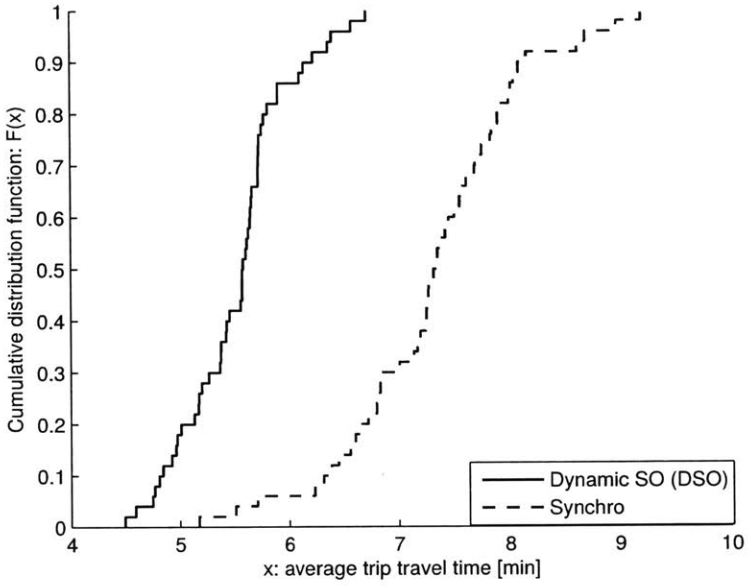
Figure 3-7(b) evaluates the performance of the plans as a function of time. This figure considers the same performance metrics as the plots of Figure 3-3, i.e., the x-axis considers the 5pm-6pm period discretized in 10 minute time increments, and the y-axis displays the average trip travel time. As a result, the best DSO plan outperforms the Synchro plan for all the six time intervals. This figure indicates that, as congestion increases, the DSO plan mitigates the increase in the travel times, while the Synchro plan leads to higher travel times.

3.5 Conclusions

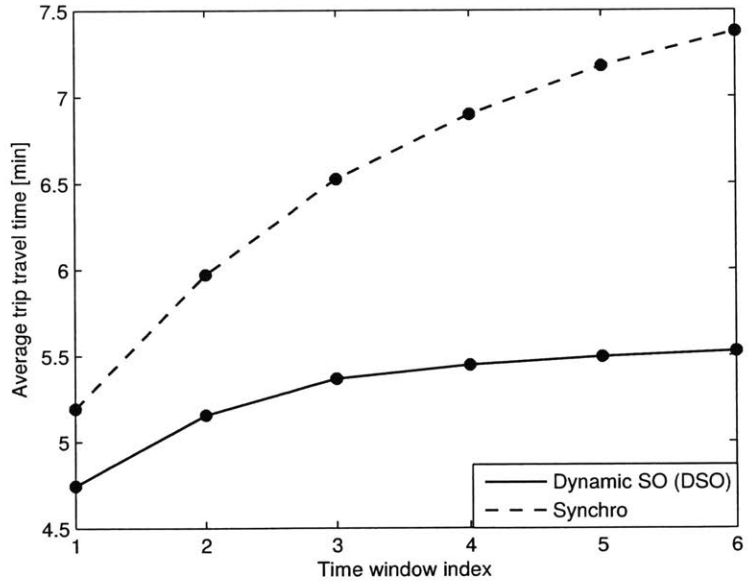
This chapter proposes a novel metamodel method that addresses large-scale simulation-based urban transportation optimization problems with time-dependent decision variables. The proposed metamodel embeds a tractable transient network model that accounts for the time variations of traffic flow and the temporal propagation of congestion in the underlying road network. The transient network model is formulated based on transient queueing theory. The proposed metamodel method is a computationally efficient technique that identifies good solutions (e.g., signal plans) under tight computational budget.

We evaluate the performance of this approach by addressing a large-scale network-wide time-dependent signal control problem for the Swiss city of Lausanne. This problem considers a congested network (evening peak period demand) with an intricate topology. We compare the performance of the proposed dynamic metamodel SO method with that of a stationary metamodel SO method proposed in Chapter 2. The dynamic metamodel SO method identifies signal plans that outperform both the initial signal plans, and most often, the signal plans derived by the stationary metamodel SO method. The analysis of this chapter also illustrates that the best DSO plan outperforms an existing signal plan for the city of Lausanne, as well as a plan derived by Synchro.

The method of this chapter allows practitioners to use a computationally efficient SO method to address a variety of dynamic large-scale transportation problems in a



(a) Comparison of the expected trip travel time of the best DSO signal plan and the signal plan derived by Synchro



(b) Comparison of the time-dependent expected trip travel time of the best DSO signal plan and the signal plan derived by Synchro

Figure 3-7: Comparison of the performance metric of the best DSO signal plan and the signal plan derived by Synchro

computational efficient way. In this chapter, the analytical transient network model is used to approximate the simulation-based objective function. Of current interest is the study of the use of this model to enhance other algorithmic steps, such as sampling strategies and ranking and selection strategies to statistically compare the performance of multiple points.

Discrete SO problems are another family of problems, where these metamodel ideas could prove beneficial. There is a lack of efficient methods for such problems, yet many network design problems, such as bike-sharing and car-sharing facility location problems, are naturally formulated as discrete problems. As part of the ongoing work, we are exploring ideas in this area.

Chapter 4

Combining Problem-specific Model-driven Methods with General Purpose Data-driven methods for Online Large-scale Urban Transportation Problems

Chapter 2 and Chapter 3 focus on large-scale offline simulation-based optimization problems. In this chapter, we focus on online large-scale optimization problems. We propose a novel computationally efficient optimization framework that combines the metamodel method proposed in Chapter 2 and a supervised machine learning method. This framework allows the use of high-resolution yet computationally expensive traffic simulators for addressing large-scale online problems.

4.1 Introduction

Today, real-time traffic information is collected by ubiquitous sensors, such as cell phones, road side units and GPS equipments. This information allows the design of traffic management strategies that are adaptive to real-time traffic conditions. These

strategies can mitigate real-time congestion and provide en-route guidance to road users. In this chapter, we formulate online optimization problems to derive these strategies. In these problems, decision variables (e.g., management strategies) are determined based on real-time data and can be implemented in real-time. To solve this type of problem, we focus on two types of methods: 1) model-driven methods, where problem-specific traffic models are used to derive solutions, and 2) data-driven methods, where solutions are derived by generic models without the need of problem-specific information such as network typology. Model-driven methods can use any type of traffic models (e.g., urban traffic simulators) whereas data-driven methods normally use generic machine learning methods, such as supervised classification or regression methods. In this chapter, we propose a new optimization framework that combines these two methods to address online transportation optimization problems for large-scale networks. In particular, we use a model-driven simulation-based optimization method and a generic supervised classification method to address a traffic responsive signal control problem of the Swiss city of Lausanne.

Past research has either used model-driven methods or data-driven methods to address online transportation optimization problems. Model-driven methods solve these problems using problem-specific traffic models, such as analytical network models (e.g., Aboudolas et al. (2007)) or simulation models (known as traffic simulators), which rely on many network modelling assumptions. On the other hand, data-driven methods do not rely on these modelling assumptions. Rather, the solutions are derived mainly based on data. In other words, data-driven methods do not require any problem-specific information such as the topology, traffic demand or supply information of networks. Model-driven methods and data-driven methods both use real-time data, although for different proposes. As stated in Barros et al. (2015): "while data-driven methods rely on the network's history to predict its evolution, model-driven methods require it in order to calibrate the parameters used in the traffic simulation". In what follows, we briefly review model-driven and data-driven methods in transportation optimization problems.

4.1.1 Literature Review: Model-driven Methods

Model-driven methods use traffic models to derive solutions of optimization problems. These traffic models make many network modelling assumptions to describe real world traffic conditions. As mentioned earlier, analytical traffic models and simulation models are the main types of traffic models. Analytical traffic models are generally low resolution models and are much simpler, whereas simulation-based models (e.g., AIMSUN (TSS; 2011)) are high-resolution models and are generally complex and less computationally efficient to evaluate.

In this review, we focus on model-driven methods used for online (i.e., real-time) traffic management problems such as responsive traffic signal control problems. A review of these methods can be found in Aboudolas et al. (2010), Osorio (2010, Apendix B). Since these methods are used to derive real-time feasible traffic management strategies, they must be computationally efficient. As Aboudolas et al. (2010) summarize, "any real-time feasible traffic signal control strategy design must include some simplification, either in its modelling approach, or in its optimization algorithm, or in its extent of network coverage". In other words, if we want to use these methods to address large-scale problems, we should either use simple and computationally inexpensive analytical models (such as Aboudolas et al. (2010) and Aboudolas et al. (2007)), or use high-resolution models, such as microscopic simulators, with computationally efficient optimization algorithms.

In this chapter, we would like to choose the latter approach: deriving solutions using computationally efficient algorithms with high-resolution traffic simulators since they can describe urban transportation systems in detail. These simulators, including microscopic and mesoscopic simulators, embed various detailed models to take into account many traveler behavior, such as departure time choice and traveller's response to en-route traffic information. Compared with most analytical traffic models, they provide more realistic representations of real traffic conditions but are more computationally expensive.

In our past work, we developed a computationally efficient simulation-based algo-

rithm (see Chapter 2) that uses these high-resolution yet computationally expensive simulators to address a traffic control problem with 99 decision variables for a network that has more than 600 links and 200 intersections. The problem was addressed with 150 simulation runs. As mentioned in Chapter 2, Section 2.4.1, this problem is considered to be large-scale in the field of traffic control.

Although this method is used to address an offline fixed-time traffic signal control problem, we believe the method can also be computationally efficient for online problems. This is because the method identifies good solutions within few simulation runs. Thus, it can potentially be used to address online optimization problems, where simulation budgets are limited by design. Therefore, we use it as the model-driven method in our proposed framework. In our future research, we would like to test the performance of other model-driven methods.

4.1.2 Literature Review: Data-driven Methods

Data-driven methods use models that make no problem-specific structural assumptions and thus can be applied to a variety of problems, such as traffic estimation and prediction problems. A detailed review of data-driven methods for traffic estimation problems can be found in Antoniou et al. (2013). In the field of optimization, however, studies are somewhat limited. Nonetheless, in most of these studies, optimization problems are formulated as estimation problems. In these studies (e.g., Anderson (2015, Chap.6)), standard direct optimization procedures (e.g., gradient-based optimization procedure) are not used to derive solutions, rather, solutions are derived by decision rules that are generalized from good solutions (e.g., good traffic management strategies) of prevalent traffic conditions.

These data-driven methods normally consist of an offline step and an online step. In the offline step, decision rules are generally developed in order to provide the mapping between the data that represents a past condition with a solution (e.g., a traffic management strategy) that works well under the condition. These rules are developed based on historical data, which contains, for instance, historical traffic demand conditions and effective traffic management strategies under these conditions.

Therefore, decision rules are generalizations of good solutions. In the online step, given real-time data, decision rules are implemented in order to generate solutions. Therefore, solutions are not derived from standard direct optimization procedure. Rather, they are estimated by decision rules.

We group data-driven methods into three types: 1) rules-of-thumb, 2) unsupervised learning and 3) supervised learning. Rules-of-thumb type of methods may set decision rules by engineering judgements, without taking into account the impact of the solutions on the objective functions of the underlying problems. This type of method has been used in studies that address traffic control problems, such as in Chiu and Chand (1993) and Nakamiti and Gomide (1996). For unsupervised learning methods, such as Srinivasan et al. (2006), and supervised learning methods, such as Anderson (2015, Chap. 6), Liu et al. (2015), and Papageorgiou et al. (1995), objective functions (or sometimes estimations of objective functions) are used to develop decision rules. In Srinivasan et al. (2006), an unsupervised learning approach (i.e., Q-learning approach) is used to develop a decision rule that learns from its own past decisions: decisions that result in a good objective function value are reinforced and vice versa. For supervised learning methods, decision rules are trained with supervised guidance: training data consists of prevalent traffic conditions and good solutions (e.g., good traffic management strategies) under these conditions, and decision rules are trained to establish the mapping between them. Several typical types of supervised learning approaches include linear decision rules such as Liu et al. (2015), neural networks such as Papageorgiou et al. (1995), classification methods such as Anderson (2015, Chap. 6) and Abbas and Sharma (2006), and look-up tables such as (Chen; 2014, Chap. 5).

It is difficult to compare the performance of these data-driven methods, since they have been used to address different types of transportation optimization problems of different networks with different decision variables. We choose the data-driven method of Anderson (2015, Chap. 6) as the data-driven method in our framework, since it is a scalable method and has been formulated for a traffic responsive signal control problem like the one addressed in this chapter. We would like to test the performance

Table 4.1: Realism, scalability, robustness and computational efficiency of the three models

	Model-driven methods	Data-driven methods
Realism		✓
Scalability	✓	
Robustness	✓	
Computational efficiency		✓

of other data-driven methods in our future research.

4.1.3 Comparison between Model-driven Methods and Data-driven Methods

In this section, we compare the advantages and disadvantages of model-driven methods and data-driven methods. We tabulate our summary in Table 4.1. For each method, we look at four aspects: realism, scalability, robustness and computational efficiency. Scalability is measured by the capability of a method to address problems with high-dimensional data, high-dimensional decision variables, and large-scale networks. Robustness is measured by the performance of a method (i.e., the quality of the solutions derived by the method) under a certain level of data quality, which can be defined as a function of data accuracy and data sparsity (in terms of sparsity in space and time). The robustness is also measured by the performance of a method under different traffic patterns. Computational efficiency is measured by the computational time that a method needs to derive a solution. In what follows, we discuss the advantages and disadvantage of these two types of methods in detail.

Model-driven methods use network modelling assumptions that might provide valuable information of unobservable parts of networks. For instance, they can utilize underlying network structural information to estimate traffic conditions of the unobservable links. Therefore, when data is sparse in space, we believe model-driven methods are more robust. In addition, since model-driven methods can be a simple approximation of real-world traffic conditions, they can be scalable for optimization problems with large-scale networks. For instance, the model-driven metamodel

method proposed in Chapter 2 is scalable for large-scale networks.

However, we believe model-driven methods may not be as realistic as data-driven methods since the network modelling assumptions used by them may not always be accurate. Moreover, compared with data-driven methods, they normally require an online optimization step, which might be computationally expensive. Therefore, we believe this computationally efficiency issue poses a practical constraint on their applicability to online problems.

Data-driven methods are more realistic since they are developed based mainly on data. Therefore, they are expected to perform well when data is complete and accurate. In addition, most data-driven methods do not include a computationally expensive online optimization procedure to derive solutions. Rather, they just implement decisions rules that can generate solutions instantaneously, which make them computationally efficient in general.

However, since decision rules of data-driven methods are trained by historical data, if real-time data cannot be represented well by training data (e.g., real-time data is out of the range of the training data), data-driven methods may not be able to find good solutions. This is known as the extrapolation problem. In other words, data-driven methods may not be robust to the quality of data. In addition, we believe that data-driven models may not be scalable to deal with high-dimensional data. Although standard dimension reduction techniques can be used to reduce the dimensionality of data, the use of these techniques can potentially deteriorate the performance of data-driven methods (in terms of solution quality) , as shown in Xu et al. (2012). Therefore, although data-driven methods have shown some success in problems with small or medium scale networks (e.g., in Liu et al. (2015)), their performance on large-scale network problems has yet to be shown.

We believe there is a need to develop a novel optimization framework that combines the use of these two types of methods. The framework has the potential of taking advantage of each method, overcoming the limitations and achieving a trade-off between realism, robustness, scalability and computational efficiency.

We propose a new optimization framework that embeds a model-driven method

and a data-driven method. The framework allows the use of any model-driven methods and any data-driven methods, as long as they are real-time feasible. In this study, as mentioned earlier, we use the simulation-based metamodel method proposed in Chapter 2 as the model-driven method and a supervised classification method proposed by Anderson (2015, Chap. 6) as the data-driven method.

Contributions

The contribution of this work is the novel online optimization framework that combines the use of a model-driven method and a data-driven method. To the best of our knowledge, this is the first time that these two types of methods are used jointly to address online large-scale transportation optimization problems. In the case study, we use the proposed framework to address a traffic responsive control problem for Lausanne network. We compare the solutions (i.e., signal plans) of the framework with the solutions derived by an optimization framework that only uses the model-driven method and with the solutions derived by an optimization framework that only uses the data-driven method. Our results show that the solutions of the proposed framework outperform those of the two frameworks most of the time, which shows the importance of combining these two methods in addressing online optimization problems.

We present the proposed online optimization framework in Section 4.2. We use the framework to address a traffic responsive control problem, formulated in Section 4.3. In Section 4.4 we illustrate the effectiveness of the framework with a case study of solving a traffic responsive control problem of Lausanne. We also describe in detail the implementation of the framework and the results. The main conclusions are presented in Section 4.5.

4.2 Methodology

4.2.1 Online Framework

In this chapter, the considered online problem consists of a number of equal length disjoint time intervals. We use the proposed framework to derive the solution of each interval. In other words, we run the framework iteratively until the end of the time horizon of interest. We denote δ the length of each interval and t the index of an interval. Interval t starts from time instant $t\delta$ and ends at $(t+1)\delta$.

For an interval t, the problem of our interest can be formulated as follows:

$$\min_{x_t} F(x_t) \tag{4.1}$$

$$g_t(x_t; p) = 0, \tag{4.2}$$

where x_t represents the solution of interval t (e.g., a signal plan). x_t is implemented during interval t, i.e., from $t\delta$ to $(t+1)\delta$. The objective function is a network performance metric F (e.g., average link speed, average link density) during interval t. We assume F can be directly measured from data. For time interval t, the feasible region is defined by a set of general analytical and differentiable constraints, g_t.

The proposed optimization framework uses real-time data to solve Problem (4.1)-(4.2). The set of real-time data, denoted $\mathcal{DA}_t$, includes data from the beginning of the time horizon until the start of interval t. Figure 4-1 displays the steps of this framework.

Figure 4-1 shows that once real-time dataset $\mathcal{DA}_t$ is available, the model-driven method and the data-driven method are launched in parallel, and each method derives its solution independently. Then, their solutions are compared in a solution selection mechanism and the best solution is chosen as the solution of the problem.

In this work, the model-driven method is the simulation-based metamodel optimization method proposed in Chapter 2. The data-driven method is a supervised classification method proposed in Anderson (2015, Chap. 6). The selection mechanism uses simulation to determine which solution (i.e., the solution derived by the

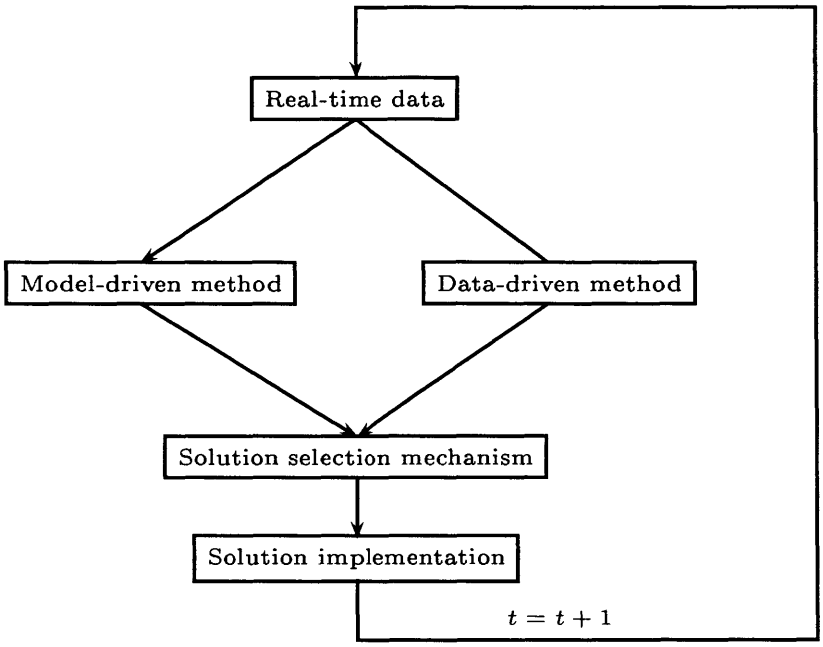


Figure 4-1: The proposed optimization framework for online problems

model-driven or data-driven method) to implement. The rest of this section introduces the model-driven method, the data-driven method, and the solution selection mechanism.

4.2.2 Model-driven Method: Simulation-based Metamodel Method

In this method, the objective function Equation (4.1) is approximated by a microscopic simulator. Thus, Problem (4.1)- (4.2) is formulated as a simulation-based optimization problem with the goal of finding solutions that minimize the simulation-based objective function. In Chapter 2, in order to solve this problem, we developed a metamodel method where the simulation-based objective function is further approximated by an analytical model known as the *metamodel.* In this method, optimization is carried out on the metamodel instead of the simulator. The metamodel is fitted iteratively, leading to an improved metamodel and potentially to points with improved objective function estimates. We refer the reader to Chapter 2, Section 2.2 for more

Algorithm 2 Model-driven method

Given real-time dataset $\mathcal{DA}_t$ and the offline calibrated simulator set $\mathcal{SM}$:

Steps:

1. Use dataset $\mathcal{DA}_t$ to choose simulator $SM(t)$.
2. Run the simulation-based metamodel method of Chapter 2 using simulator $SM(t)$ until the computational budget is depleted.

details regarding the metamodel method.

The metamodel method relies on the use of a microscopic traffic simulator to derive solutions. In the context of addressing online problems, for a given time interval t, a given simulator, denoted $SM(t)$, should be used. To obtain $SM(t)$, we can either use $\mathcal{DA}_t$ to solve an online real-time calibration problem to calibrate the demand and supply parameters of the simulator (e.g., Antoniou (2004)), or we can use $\mathcal{DA}_t$ to choose a simulator from a set of pre-calibrated simulators. In this work, for the sake of simplicity, we use the latter approach, although using the former approach is entirely possible and desirable. We describe our proposed simulator selection method in Appendix C.1.

Algorithm 2 describes the steps of the model-driven method to solve Problem (4.1)- (4.2). In Step 1, we use real-time dataset $\mathcal{DA}_t$ to choose which simulator to use, i.e., $SM(t)$, according to the selection method described in Appendix C.1. In Step 2, we run the metamodel method of Chapter 2 to derive the solution using simulator $SM(t)$.

To represent the computational budget constraint, in the implementation of this algorithm, we allow a maximum number of three metamodel iterations. Once three iterations are finished, the solution, denoted x_t^{model}, is derived. The performance of x_t^{model} will be evaluated in the solution selection mechanism of the proposed framework.

For more details regarding the implementation of this algorithm for the case study of this chapter, we refer the reader to Section 4.4.2 and Section 4.4.4.

4.2.3 Data-driven Method: Supervised Classification Method

The data-driven method used in this work is a supervised classification method proposed in Anderson (2015, Chap. 6). To address Problem (4.1)- (4.2), this method selects a solution from a pre-determined fixed set. The choice depends on real-time data $\mathcal{DA}_t$. This method has an offline step and an online step: the offline step trains a decision rule whereas the online step implements the decision rule to select the solution of the online problem.

In the offline step, a training set is created and then is used to train the decision rule. To create the set, first a fixed-finite set of solutions, denoted $\mathcal{D}$, is built. The set, for instance in Anderson (2015, Chap. 6), consists of existing solutions (i.e., signal plans). Second, historical conditions (from the historic data) are associated with solutions in set $\mathcal{D}$. For each historical traffic condition, the solution with the best performance is identified. The index of the solution is called the label of the traffic condition. Once all traffic conditions are labelled, based on their labels, they are grouped into several classes where traffic conditions with the same label (i.e., the same solution) are grouped into the same class. In classification terms, traffic conditions that share the same solution form a class, and the index of the solution is called the class label. Thus, the training data consists of historical traffic conditions and their class labels. In the training step, the goal is to find a classification model that selects the best solution (i.e., the correct class label) of a given traffic condition. In Anderson (2015, Chap. 6), three classification models are used, i.e., decision trees, random forest and k-nearest neighbors. We refer the reader to Anderson (2015, Chap. 6) for more details.

In addition, in both the offline and online steps, Anderson (2015, Chap. 6) proposes a data preprocessing step to reduce data dimension and regularize data. Therefore, the preprocessed data becomes the input of the classification models. In the offline step, historical data is preprocessed and then used to train classification models; in the online step, real-time data is preprocessed and then used to select the solution. We refer the reader to Appendix C.2 for more details regarding the preprocessing

Algorithm 3 Data-driven method

Given real-time dataset $\mathcal{DA}_t$:

Steps:

1. Preprocess data $\mathcal{DA}_t$.
2. According to a classification model, use the preprocessed data to determine the class label and thus the solution.

procedure.

Algorithm 3 summarizes the online step of Anderson (2015, Chap. 6). It uses a subset of data $\mathcal{DA}_t$, i.e., data of interval $t-1$, to choose the solution. Step 1 is the preprocessing step. Firstly, a standard feature reduction technique, i.e., principle component analysis (PCA) or linear discriminate analysis (LDA), is used to reduce data dimension. Secondly, rounding and scaling techniques are used to regularize and normalize these reduced dimensions. In Step 2, an offline trained classification model is used to determine the class label based on the prepossessed data, and thus, determines the solution.

In the case study of Anderson (2015, Chap. 6), the author uses two feature reduction techniques (i.e., LDA and PCA) to preprocess training data and trains three types of classification methods (i.e., decision trees, random forest and k-nearest neighbors). This creates six classification models. The performance of these models are compared, both in terms of misclassification errors and the performance of their derived solutions (i.e., signal plans).

In the case study of this work, we use the same six models in the offline step. We then compare their performance according to their misclassification rates, which are calculated using the standard cross-validation procedure. In the online step, we use the best three models to derive the proposed solution, denoted x_t^{data}. To do this, we run Algorithm 3 three times, each with one model. This create three class labels, with the label that is chosen by the majority of the three models becoming the label of the solution x_t^{data}. This is known as the *majority vote* in classification terms. If there is a tie between two models, we choose the solution identified by the model with

the better performance. The performance of x_t^{data} will be evaluated in the solution selection mechanism of the optimization framework.

4.2.4 Solution Selection Mechanism

Once the solution of the model-driven method and the solution of data-driven method are derived, this mechanism is launched to compare their performance and chooses the one with a better performance. In this framework, we use the offline calibrated simulator $SM(t)$ to evaluate the performance of the two solutions (x_t^{model} and x_t^{data}). In the case study of this chapter, we implement solutions x_t^{model} and x_t^{data} in $SM(t)$, run simulation replications, collect simulation outputs (i.e., simulated data) of each replication and calculate their corresponding objective function values. We use the average of the objective function values of these replications as the performance metric of a solution. The solution that has a better performance metric is chosen as x_t.

4.3 Traffic Responsive Control Problem

The proposed framework is suitable to address a variety of online transportation optimization problems. In this chapter, we use the framework to address a traffic responsive signal control problem. We formulate this problem in this section.

We consider a time horizon that consists of T disjoint time intervals. For each time interval, we solve a traffic control problem, deriving a signal plan based on real-time data $\mathcal{DA}_t$. The signal plans for the T intervals are derived sequentially. Within the same interval, the signal plan is fixed and periodic.

A detailed review of traffic signal control terminology is given in Appendix A of Osorio (2010). The common decision variables of signal control problems are cycle lengths, green splits and offsets. This chapter focuses only on the optimization of the green splits of a number of signalized control lanes at multiple intersections that are distributed throughout in the network. The other signal plan variables, such as cycle lengths and offsets, are fixed.

To formulate this problem, we introduce the following notation:

c_i	cycle time of intersection i;
d_i	fixed cycle time of intersection i;
x_t	vector of green splits of interval t;
$x_t(j)$	green split of signal phase j of interval t;
x_{LB}	vector of lower bounds of green splits;
$\mathscr{I}$	set of intersection indices;
$\mathscr{P}_I(i)$	set of endogenous signal phase indices of intersection i.

For each interval t, the objective function of the traffic responsive control problem is given as follows:

$$\min_{x_t} F(x_t) \tag{4.3}$$

subject to

$$\sum_{j \in \mathscr{P}_I(i)} x_t(j) = \frac{c_i - d_i}{c_i}, \ \forall i \in \mathscr{I} \tag{4.4}$$

$$x_t \geq x_{LB}. \tag{4.5}$$

The decision vector x_t consists of the green splits of the controlled signal phases during time interval t. The objective function $F(x_t)$ could be any performance metric that can be directly calculated from data. In this work, we use the average of the expected link travel time of the links with sensors during interval t as $F(x_t)$. The expected travel time of a link during interval t is defined as the expected travel time of all the vehicles that have traversed the link during interval t.

4.4 Empirical Analysis: Lausanne City Case Study

In this section, we evaluate the effectiveness of the proposed framework by addressing a traffic responsive control problem of Lausanne. Section 4.4.1 describes the Lausanne signal control problem. The rest of the section describes the implementation details of the framework and the experimental results. Section 4.4.2 describes the offline step of the model-driven method. Section 4.4.3 describes the offline step of the data-driven method. Section 4.4.4 describes the online step of solving the traffic responsive control problem and the corresponding results.

4.4.1 Lausanne City Network

In this case study, we solve a traffic responsive signal control problem of the full city of Lausanne. The network consists of over 600 links and 200 intersections. We control the green splits of 99 signal phases that spread over 17 controlled intersections throughout the network. The city map is displayed in Chapter 2, Figure 2-2, the considered area is delimited in white. The time horizon of this study is from 5:00 pm- 6:30 pm during evening peak hours. We only optimize the signal plans for the last hour, i.e., 5:30 pm -6:30 pm, during which congestion gradually builds up and continues to grow. Hence, there is a need to design signal plans that are adaptive to the growth of congestion. To do this, we divide this hour into four 15 minute intervals. This gives us four intervals to optimize. In other words, for each interval, we solve a traffic control problem with 99 decision variables.

We assume 60 links are deployed with sensors. These links are the inflow links to the 17 intersections. These links consist of roughly ten percent of the links in the network.

4.4.2 Offline Experiment Design: Model-driven Method

Create Offline Simulator Set $\mathcal{SM}$

We create set $\mathcal{SM}$ based on the simulation model of the Lausanne developed by Dumont and Bert (2006). This simulator has only one origin-destination (OD) demand matrix calibrated for the evening peak hour. Therefore, in order to create simulators with different demand scenarios, we create several new synthetic OD demand matrices from the Lausanne OD demand matrix. These synthetic OD demand matrices represent different levels of demand in the Lausanne network, from light traffic to heavy traffic. In the rest of this section, the Lausanne OD demand matrix is referred to as the base OD matrix.

We now describe the technique that generates different OD demand matrices from the base matrix. This technique is similar to the synthetic demand generation technique, which can be found in Balakrishna (2002, Chap. 2), Antoniou (2004, Chap. 4), Liu et al. (2015), Cascetta et al. (1993), and Chen (2014, Chap. 5). The synthetic OD demand matrices used by Balakrishna (2002, Chap. 2) and Antoniou (2004, Chap. 4) are derived from historical data. When historical data is not available, such as the case study of this work, two types of synthetic demand generation techniques are used: scaling (such as in Chen (2014, Chap. 5) and resampling (such as in Liu et al. (2015) and Cascetta et al. (1993)). The scaling technique uses a fixed scalar to multiply a base OD demand matrix to create a new one. In the resampling technique, a new synthetic OD demand matrix is randomly sampled from a known distribution that can be specified from a base OD demand matrix. In the case study, we use both techniques to create six synthetic OD demand matrices from the base OD demand matrix (i.e., the Lausanne OD demand matrix). Five are scaled demand scenarios and one is a randomly sampled demand from a multi-variate normal distribution with expectation equal to that of the Lausanne OD demand matrix and with standard deviation set to ten percent of the expectation. Once the six demand scenarios are defined, they are embedded in six simulators, respectively. Set $\mathcal{SM}$ consists of these six simulators.

4.4.3 Offline Experiment Design: Data-driven Method

As mentioned earlier in Section 4.2.3, the data-driven method uses offline-trained classification models to select solutions online. In this section, we describe the offline steps of the data-driven method. We summarize the main steps in Algorithm 4.

In Algorithm 4, step 1 summarizes the steps to create training data. Step 2 describes the steps to train the data-driven method. Note that the training steps we use for this case study are different from those of Anderson (2015, Chap. 6), because the data type and the solution set $\mathcal{D}$ are different. In what follows, we describe our steps as well as the differences between our approach and Anderson's approach.

Step 1 (a) builds demand scenarios under which training data is collected. In Anderson (2015, Chap. 6), historical data collected from sensors is used. However, since we have do not have historical data, we resort to the use of simulation. The demand scenarios we use here are the same six demand scenarios that are used to create the simulator set $\mathcal{SM}$ in the model-driven method.

Step 1 (b) builds set $\mathcal{D}$ that consists of the signal plans for the method to select from. By design, each plan in set $\mathcal{D}$ needs to be the best plan of at least one demand scenarios. In Anderson (2015, Chap. 6), three existing signal plans form the set, where each plan in the set is indeed the best plan of at least one scenarios. However, in our case study, the best plans under the six demand scenarios are not available to us. Therefore, in this step, we first derive plans that perform well in the six scenarios and then include a subset of these plans in set $\mathcal{D}$. We refer the reader to Appendix C.3 for more details.

Step 1 (c) matches demand scenarios created in Step 1 (a) with the signal plans in set $\mathcal{D}$. The goal of this step is to determine the class label of each demand scenario, which is also the label of the signal plan (in $\mathcal{D}$) that performs the best under the scenario. In Anderson (2015, Chap. 6), this step involves the use of an analytical traffic model, i.e., Webster's method (Webster; 1958). In this work, we use the simulators in set $\mathcal{SM}$ to evaluate the quality of signal plans. Once the match is established, we generate the training data through simulation.

Algorithm 4 Data-driven method offline step implementation

Steps:

1. Create training data
   (a) Create demand scenarios.
   (b) Create the signal plan set $\mathcal{D}$.
   (c) Match demand scenarios to signal plans in $\mathcal{D}$ and create training data.
2. Train the classification models of interest and test their performance.

Step 2 describes the training step. This step includes the standard training procedure of a classification model, which consists of: 1) separating training data into a training set and a cross-validation set, 2) using k-fold cross-validation technique to determine the best parameters of a model. Once a classification model is trained, we evaluate its performance: we use the model to predict the class labels of the data in the cross-validation set and calculate the misclassification rate.

As mentioned in Section 4.2.3, six classification models are considered but only three models with low misclassification rates are used in the online step. They are random forest with LDA (i.e., the LDA technique is used in the preprocessing step and random forest is used as the classification method), k-nearest neighbors with LDA and random forest with PCA. The solution of the traffic responsive control problem is determined by the majority vote of these three models. We refer the reader to Appendix C.4 for more details.

4.4.4 Online Experiments

In this section, we use three frameworks to address the traffic responsive control problem of Lausanne: 1) an optimization framework that uses only model-driven simulation-based metamodel method, denoted SO; 2) an optimization framework that uses only the data-driven method, denoted DA; and 3) our proposed online optimization framework that combines the model-driven method and the data-driven method, denoted SOD. Note that the SO (resp. DA) framework is the SO (resp. DA) branch of the SOD framework (see Figure 4-1).

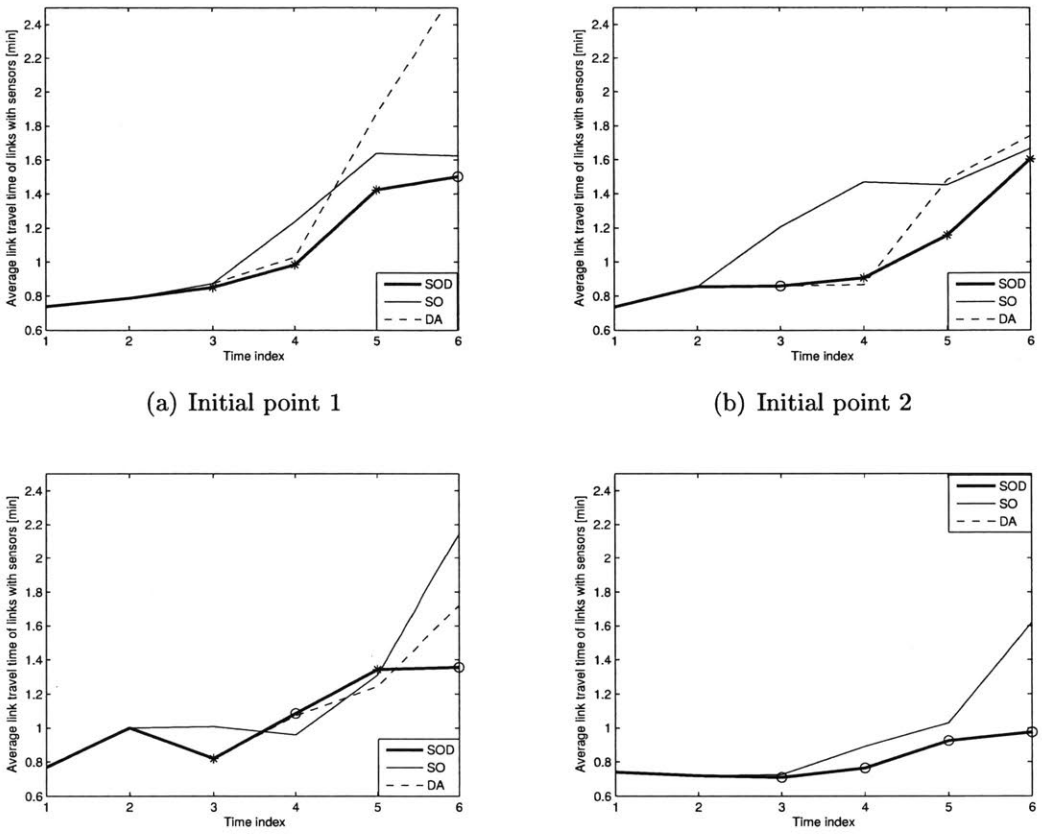
We test the performance of the three frameworks under two demand scenarios: 1) the current Lausanne demand, denoted d_{R1}, and 2) a synthetic Lausanne demand, denoted d_{R2}. We denote SR_1 (resp. SR_2) the simulator that embeds demand scenario d_{R1} (resp. d_{R2}). To create scenario 2), we sample from a multi-variate normal distribution as described previously. Both scenarios are different than the six scenarios used in the offline step. For each demand scenario, we compare the performance of signal plans derived by the three frameworks.

All three frameworks are responsive to real-time data: the model-driven method in SO and SOD uses real-time data to select simulator $SM(t)$ in order to run the metamodel method, and the data-driven method in DA and SOD uses real-time data as the input to the classification methods. However, because real-time data is not available in this study, we generate real-time data by simulation.

Real-time Data Generation

Simulators SR_1 and SR_2 are stochastic simulators: the data generated by them varies across simulation replications. Therefore, for each demand scenario, we can create simulated data from different simulation replications and use them to test the performance of a framework. In the rest of this section, we refer to these data from different replications as different initial points. In this experiment, for each demand scenario, we use four initial points.

To create data for each initial point, we run five simulation replications of the corresponding simulator and use the average of the simulation data from these replications as the real-time data. For instance, to create real-time data of a link statistic (e.g., the expected link travel time), we run the corresponding simulator five times (each with one simulation replication) and take the average of the five link statistics as the real link statistic. To avoid further confusion, we refer to the simulated data generated from these two simulators as the real-time data. In this experiment, real-time data consists of the expected link travel times of links with sensors.



(a) Initial point 1

(b) Initial point 2

(c) Initial point 3

(d) Initial point 4

Figure 4-2: Time-dependent average link travel times of the links with sensors of the four initial points under the Lausanne demand scenario

Experimental Results

Now we compare the performance of the three frameworks: 1) SO, 2)DA and 3) SOD. For each demand scenario, we consider four initial points. For each initial point, we run each framework once. For each run, we collect real-time data of the six fifteen minute intervals from 5:00 pm to 6:30 pm. Note that since we only optimize signal plans for the last four intervals (i.e., 5:30 pm to 6:30 pm), for each initial plan, the differences in the performance of three frameworks are in the last four intervals.

Demand Scenario 1): Lausanne demand

Figure 4-2 plots the performance of the three frameworks for each of the four initials points under demand scenario 1): the Lausanne demand. The x-axis shows

the index of a time interval, i.e., time index 1 represents the time interval between 5:00 pm to 5:15 pm, 2 represents the interval between 5:15 pm and 5:30 pm, etc. The y-axis represents the average link travel time (in minutes) of the links with sensors during the corresponding time interval. In each figure of Figure 4-2, the solid thick curve corresponds to the signal plans derived by SOD, the solid thin curve corresponds to the signal plans derived by SO and the dashed curve corresponds to the signal plans derived by DA. Each curve is obtained by interpolating six data points, where each point represents the objective function value of each time interval. We also mark the choices made by the selection mechanism of SOD on the SOD curve. Whenever SOD chooses the signal plan derived by the model-driven method of an interval, we mark a star at the corresponding interval. Whenever SOD chooses the plan derived by the data-driven method, we mark a circle.

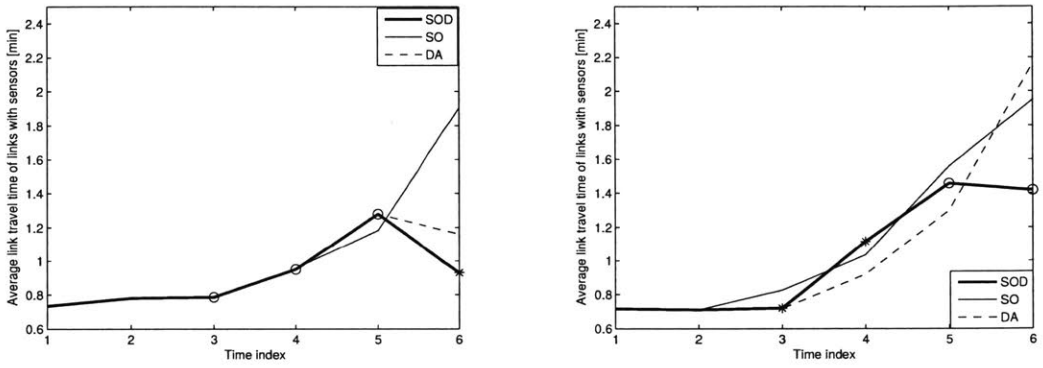
For the four plots of Figure 4-2, SOD performs the best for all intervals in Figure 4-2(a) and Figure 4-2(d). In Figure 4-2(b), SOD performs the best during interval 3, interval 5 and interval 6 and performs slightly worse than the best plan (i.e. the plan derived by DA) during interval 4. In Figure 4-2(c), SOD performs the best in interval 3 and interval 6 but performs the worst in interval 4 and interval 5.

Figure 4-2 shows that SO outperforms DA under some circumstances, and vice versa. In other words, the model-driven method does not always outperform the data-driven method and vice versa. This shows the need of combining these two methods in an optimization framework, which has the potential to outperform each of them.

Note that in Figure 4-2(a), although SOD chooses the plan derived by model-driven method at interval 3, the SOD curve does not overlap with SO curve at interval 3. This is because the model-driven method uses a stochastic simulator $SM(t)$ to derive signal plans, and thus, it is likely that the signal plans derived by model-driven method in the SOD framework and the model-driven method in the SO framework are different. This happens to be the case in the interval 3 of Figure 4-2(a).

Demand Scenario 2): a synthetic Lausanne demand

Figure 4-3 plots the performance of three frameworks of the four runs under demand scenario 2): a synthetic Lausanne demand. In Figure 4-3(d), SOD performs the



(a) Initial point 1

(b) Initial point 2

(c) Initial point 3

(d) Initial point 4

Figure 4-3: Time-dependent average link travel times of the links with sensors of the four initial points under a synthetic Lausanne demand scenario

best in all intervals. In Figure 4-3(a), Figure 4-3(b) and Figure 4-3(c), SOD performs the best most of the time.

4.5 Conclusions

In this chapter, we propose a computationally efficient optimization framework for large-scale online transportation optimization problems, where solutions of each time interval of interest need to be found within limited computational budgets. For each interval, the framework launches in parallel a model-driven method and a data-driven method to derive the solution of the problem. In this work, the model-driven method is a metamodel method proposed in Chapter 2 and the data-driven method is a supervised classification method proposed by Anderson (2015, Chap. 6). Both methods are responsive to real-time data. The framework uses a solution selection mechanism to choose the better solution from the solutions derived by the two methods.

We evaluate the effectiveness of this online framework by addressing a large-scale traffic responsive control for the Swiss city of Lausanne under two demand scenarios. For each scenario, we compare the performance of the proposed optimization framework (SOD) with that of a framework that includes only the metamodel method (SO) and a framework that includes only the data-driven method (DA). The experimental results show that SOD is able to derive signal plans that outperform those derived by SO and DA most of the time. And this is true across different traffic conditions under two different demand scenarios. This illustrates the value of combining both methods in the online framework.

The proposed framework can be used by practitioners to address a variety of online large-scale urban transportation problems. For instance, the framework allows for a broad family of objective functions, such as minimizing network density. In addition, the framework is quite general and can embed any type of real-time feasible model-driven method and data-driven method.

In our future work, we would like to improve the design of the model-driven method and the data-driven method in order to provide better solutions. As for

the model-driven method (i.e., the metamodel method), we would like to develop efficient offline sampling techniques to improve the goodness-of-fit of the metamodel, such that the metamodel method is more likely to identify a good solution within limited computational online budgets. In addition, to resolve the potential issue that the simulator used in the metamodel method may not represent well real-time traffic conditions, we would like to use online calibrated simulators to derive solutions. As for the data-driven method, the supervised classification method that we use does not generate new solutions, which may limit its potential. Therefore, in the future, we would like to update the solution set according to real-time data. Last but not least, we would like to improve the solution selection mechanism of the framework to increase the chance of selecting solutions with better performance.

Chapter 5

Conclusions and Future Research

5.1 Conclusions

This chapter concludes the main contents of this thesis, and also provides directions for future research.

Chapter 2 proposes a simulation-based optimization algorithm, for problems with high-dimensional decision variables, simulation-based stochastic objective functions and general analytical differentiable constraints. The algorithm embeds a novel metamodel method that combines the use of a scalable stationary macroscopic traffic model with a microscopic simulator. It is a computationally efficient algorithm that identifies trial solutions (e.g., signal plans) with improved performance within limited computational budgets. The performance of this algorithm is evaluated by addressing a fixed-time large-scale network wide signal control problem for the Swiss city of Lausanne. The network is considered to be a congested network (evening peak period demand) with an intricate topology. The performance of this algorithm is compared with that of a traditional simulation-based optimization algorithm. The proposed algorithm identifies signal plans that improve the distribution of average travel times compared to the signal plans derived by the traditional algorithm. The performance of the proposed algorithm is also compared to that of a widely-used signal control software, Synchro. All proposed signal plans outperform the plan derived by Synchro.

Chapter 3 proposes a simulation-based optimization algorithm for large-scale dy-

namic problems with high-dimensional time-dependent decision variables, simulation-based objective functions and general analytical differentiable constraints. The algorithm is designed for addressing dynamic simulation-based transportation optimization problems. The algorithm uses a metamodel method that combines the use of an analytical transient traffic model and a microscopic simulator. The proposed algorithm is used to solve a time-dependent signal control problem of the entire city of Lausanne with limited simulation budgets. The performance of the proposed algorithm is compared with the performance of the algorithm proposed in Chapter 2, under the same simulation budget. The latter uses a stationary traffic model as the corresponding metamodel. As a result, the purposed algorithm of this chapter identifies signal plans that improve the distribution of average travel times compared to initial signal plans, and most of the time, outperform the plans derived by the algorithm proposed in Chapter 2. This shows the importance of using a dynamic traffic model in the metamodel method. The best signal plan derived by the proposed algorithm of the chapter outperforms the signal plan derived by the signal control software Synchro, and an existing Lausanne signal plan.

Chapter 4 proposes a novel optimization framework for online transportation optimization problems. The framework consists of a model-driven method, i.e., the simulation-based optimization algorithm proposed in Chapter 2, and a data-driven method, i.e., a supervised classification method proposed by Anderson (2015, Chap. 6). Both methods use real-time data to derive solutions. We use the framework to solve a traffic responsive signal control problem of the entire city of Lausanne under two demand scenarios. We also compare the framework with a framework using only the model-driven method and with a framework using only the data-driven method. As a result, for both demand scenarios, the performance of the proposed framework, in terms of the quality of the derived signal plans, outperforms the framework with only the model-driven method and the framework with only the data-driven method. This shows the added value of combining two types of methods in addressing online optimization problems.

5.2 Future Research Directions

This thesis research allow practitioners to use computational efficient optimization algorithms to address a variety of offline (Chapter 2 and Chapter 3) and online (Chapter 4) problems. In the future, we would like to investigate the use of these algorithms to address a variety of generally constrained transportation problems, such as simulation model calibration, dynamic congestion pricing, and multi-modal traffic management problems.

We believe the proposed optimization algorithms in this thesis have the potential to be improved, both in terms of effectiveness and computational efficiency. For instance, the effectiveness of the online optimization framework of Chapter 4 depends on the accuracy of the selection mechanism, which relies on a microscopic simulator that may not well represent real-time traffic conditions. In the future, we would like to use real-time data to calibrate this simulator online to improve the selection mechanism.

We would also like to improve the offline experimental design to improve the performance of online optimization algorithms. In this work, we believe the performance of the model-driven method is hindered by the limited simulation runs that are allowed online. This leads to a small sample size to fit the metamodel, which may result in the lack-of-fit issue. One way to improve the fit is to include information from offline simulation samples, which can be generated from simulators that are calibrated offline based on historical traffic conditions.

From a methodological perspective, we are interested in developing new simulation-based optimization algorithms with improved short-term performance by using information from analytical traffic models (such as the queueing network models used in this thesis) to inform both sampling strategies, as well as ranking and selection strategies to statistically compare the performance of multiple points.

Last but not least, this thesis only considers problems with analytical constraints. In the future, we would like to extend our algorithms to address problems with simulation-based (i.e., stochastic) constraints. These problems require evaluating

the feasibility of a point via simulation, i.e., numerous simulation replications need to be run in order to test for feasibility. For this reason, problems with simulation-based constraints can be computationally more challenging to address. In the future, we would like to explore the metamodel idea of this thesis to analytically approximate these stochastic constraints and thus potentially save multiple runs to test the feasibility of a solution.

Appendix A

Appendices of Chapter 2

A.1 Derivation of $E[N]$

This appendix serves to describe how the analytical expression of the $E[N]$ is derived by the proposed tractable stationary queueing network model described in 2.2.3. $E[N]$ is introduced in Chapter 2, Section 2.2.4.

In this section we omit the index i that refers to a given queue. $E[N]$ is defined as:

$$E[N] = \sum_{n=0}^{k} nP(N = n). \tag{A.1}$$

The stationary queue-length probabilities for each queue, $P(N = n)$, are given in Bocharov et al. (2004) by:

$$P(N = n) = \frac{1 - \rho}{1 - \rho^{k+1}} \rho^n. \tag{A.2}$$

Inserting Equation (A.2) into (A.1), and then rearranging the terms yields

$$E[N] = \sum_{n=0}^{k} n \frac{1-\rho}{1-\rho^{k+1}} \rho^n, \tag{A.3}$$

$$= \sum_{n=1}^{k} n \frac{1-\rho}{1-\rho^{k+1}} \rho^n, \tag{A.4}$$

$$= \frac{1-\rho}{1-\rho^{k+1}} \sum_{n=1}^{k} n\rho^n, \tag{A.5}$$

$$= \frac{1-\rho}{1-\rho^{k+1}} \rho \sum_{n=1}^{k} n\rho^{n-1}. \tag{A.6}$$

We then derive an expression for the last summation as follows. For a geometric series, such that $\rho \neq 1$, we have:

$$\sum_{n=0}^{k} \rho^n = \frac{\rho^{k+1}-1}{\rho-1}. \tag{A.7}$$

We differentiate this formula with respect to ρ and obtain:

$$\sum_{n=1}^{k} n\rho^{n-1} = \frac{1-\rho^{k+1}}{(1-\rho)^2} - \frac{(k+1)\rho^k}{1-\rho}. \tag{A.8}$$

Inserting the expression of Equation (A.8) into Equation (A.6), and rearranging the terms gives:

$$E[N] = \frac{1-\rho}{1-\rho^{k+1}} \rho \left(\frac{1-\rho^{k+1}}{(1-\rho)^2} - \frac{(k+1)\rho^k}{1-\rho} \right) \tag{A.9}$$

$$= \rho \left(\frac{1}{1-\rho} - \frac{(k+1)\rho^k}{1-\rho^{k+1}} \right). \tag{A.10}$$

A.2 SO Algorithm

This appendix serves to describe the SO algorithm introduced in Chapter 2, Section 2.2.5. This SO algorithm is formulated in detail in Osorio and Bierlaire (2013) and is based on the derivative-free trust region algorithm of Conn et al. (2009a). The

notation used is that of Osorio and Bierlaire (2013).

0. **Initialization.**

   Define for a given iteration k: $m_k(x, z; \beta_k, p)$ as the metamodel (denoted hereafter as $m_k(x)$), x_k as the iterate, Δ_k as the trust region radius, $\nu_k = \beta_k$ as the vector of parameters of m_k, n_k as the total number of simulation runs carried out up until and including iteration k, u_k as the number of successive trial points rejected, ε_k as the measure of stationarity (norm of the derivative of the Lagrangian function of the trust region (TR) subproblem with regards to the endogenous variables) evaluated at x_k.

   The constants $\eta_1, \gamma, \gamma_{inc}, \varepsilon_c, \bar{\tau}, \bar{d}, \bar{u}, \Delta_{max}$ are given such that: $0 < \eta_1 < 1$, $0 < \gamma < 1 < \gamma_{inc}$, $\varepsilon_c > 0$, $0 < \bar{\tau} < 1$, $0 < \bar{d} < \Delta_{max}$, $\bar{u} \in \mathbb{N}^*$. Set the total number of simulation runs permitted (across all points) n_{max}, this determines the computational budget. Set the number of simulation replications per point $\tilde{r}$ (here we use $\tilde{r} = 1$).

   Set $k = 0, n_0 = 1, u_0 = 0$. Determine x_0 and Δ_0 ($\Delta_0 \in (0, \Delta_{max}]$).

   Given the initial point x_0, compute $f_A(x_0)$ (analytical approximation of Equation (2.18)) and $\hat{f}(x_0)$ (simulated estimate of Equation (2.18)), fit an initial model m_0 (i.e., compute ν_0).

1. **Criticality step.** If $\varepsilon_k \leq \varepsilon_c$, then switch to *conservative mode.*

2. **Step calculation.** Compute a step s_k that reduces the model m_k and such that $x_k + s_k$ (the trial point) is in the trust region (i.e. approximately solve the TR subproblem).

3. **Acceptance of the trial point.** Compute $\hat{f}(x_k + s_k)$ and

$$\rho_k = \frac{\hat{f}(x_k) - \hat{f}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

   - If $\rho_k \geq \eta_1$, then accept the trial point: $x_{k+1} = x_k + s_k$, $u_k = 0$.
   - Otherwise, reject the trial point: $x_{k+1} = x_k$, $u_k = u_k + 1$.

Include the new observation in the set of sampled points ($n_k = n_k + \tilde{r}$), and fit the new model m_{k+1}.

4. **Model improvement.** Compute $\tau_{k+1} = \frac{\|\nu_{k+1} - \nu_k\|}{\|\nu_k\|}$. If $\tau_{k+1} < \bar{\tau}$, then improve the model by simulating the performance of a new point x, which is uniformly drawn from the feasible space. Evaluate f_A and $\hat{f}$ at x. Include this new observation in the set of sampled points ($n_k = n_k + \tilde{r}$). Update m_{k+1}.

5. **Trust region radius update.**

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_{inc}\Delta_k, \Delta_{max}\} & \text{if } \rho_k > \eta_1 \\ \max\{\gamma\Delta_k, \bar{d}\} & \text{if } \rho_k \leq \eta_1 \text{ and } u_k \geq \bar{u} \\ \Delta_k & \text{otherwise.} \end{cases}$$

   If $\rho_k \leq \eta_1$ and $u_k \geq \bar{u}$, then set $u_k = 0$.

   If $\Delta_{k+1} \leq \bar{d}$, then switch to *conservative mode.*

Set $n_{k+1} = n_k, u_{k+1} = u_k$, $k = k + 1$.

If $n_k < n_{max}$, then go to Step 1. Otherwise, stop.

Appendix B

Appendices of Chapter 3

B.1 Metamodels for Transportation Problems

This appendix serves to provide a detailed review of past work that uses the simulation-based metamodel algorithms mentioned in Chapter 3, Section 3.2.

In our past work, we have developed efficient SO algorithms for various transportation problems with case studies of Lausanne (Switzerland), Manhattan (NYC, USA) and Berlin (Germany). We summarize here the main insights obtained from past work. The formulation presented in this chapter is motivated by these insights. Table B.1 summarizes the main features of the methods and case studies used for signal control problems. The last row represents the method of Chapter 3. The second and third columns indicate whether the SO method is designed for time-independent or time-dependent problems. More specifically, these methods embed information from analytical traffic models. Columns 2 and 3 indicate whether the analytical models are stationary or transient, respectively. Columns 4-6 indicate whether the case studies have benchmarked the derived signal plans versus signal plans that are: randomly drawn, prevailing in the field or derived from the commercial signal control software Synchro. Columns 7-9 indicate the number of roads, of intersections and of decision variables of the case studies. For papers that include multiple case studies, the table indicates the largest-scale case study. The last column indicates the objective function.

Table B.1: Summary of metamodel SO methods for signal control problems

	Model		Outperforms			Number of			Metamodel of
	Stat.	Trans.	Random	Field	Synchro	Roads	Inter.	Dec. var.	
Osorio and Bierlaire (2013)	✓		✓	✓		48	15	51	Trip travel time
Osorio and Nanduri (2015a)	✓		✓			47	15	51	Fuel consumption and trip travel time
Osorio and Nanduri (2015b)	✓		✓	✓		47	15	51	Pollutant emissions and trip travel time
Osorio et al. (2014)	✓		✓	✓		134	32	64	Queue-lengths
Osorio et al. (2016)	✓		✓			603	231	99	Link travel time variance and expectation
Osorio and Chong (2015) (i.e., Chapter 2)	✓		✓		✓	603	231	99	Trip travel time
Chong and Osorio (forthcoming) (i.e., Chapter 3)		✓	✓	✓	✓	603	231	198	Trip travel time

Column 2 indicates that Chapter 3 is the first to formulate a metamodel with a transient (i.e., time-dependent) traffic model. This table indicates that formulations for a variety of objective functions have been proposed and successfully used for problems that are considered large-scale for both signal control and for SO.

Recall that in a metamodel framework, the objective function needs to be approximated analytically by the metamodel. For a specific objective function, the key to designing a computationally efficient SO algorithm lies in the formulation of an analytical traffic model that is both: (i) a good approximation of the unknown objective function, and (ii) is sufficiently tractable such that Problem (3.4)-(3.6) can be solved efficiently. Column 10 of the table indicates that we have designed algorithms for objective functions that are intricate to approximate analytically (e.g., vehicular emissions, fuel consumption, travel time variability).

All these case studies include analysis with tight computational budgets, where the simulation budget ranges from 50 to 150. Even for such tight budgets, the signal plans identified by the metamodel methods outperform a variety of signal plans: (i) randomly drawn plans, (ii) plans prevailing in the field, and (iii) plans derived by mainstream widely used commercial software. The results of these various case studies consistently indicate that the ability of these SO methods to identify good solutions under tight budgets is due to the combination of simulation observations with analytical traffic model information.

More specifically, the papers tabulated in Table B.1 (excluding Osorio et al. (2014)) have benchmarked the metamodel of Equation (3.3) with a metamodel that

does not include information from the analytical traffic model (i.e., $m = \phi$ in (3.3)). All case studies have shown that by embeding information from analytical traffic models, the following properties are achieved: (i) solutions with better performance are identified, (ii) good quality solutions are identified within significantly fewer simulation runs, and (iii) the algorithm becomes robust to the quality of the initial points. Note that a metamodel approach with $m = \phi$ (i.e., with a local quadratic general-purpose metamodel) is a traditional SO approach. It corresponds to an iterative response-surface methodology, which is broadly and commonly used in the literature and is often not referred to as a metamodel technique. For lower-dimensional problems with larger computational budgets (i.e., easier problems to solve), the traditional approach and the metamodel with analytical traffic information identify solutions with similar performance (see Osorio and Bierlaire (2013), for problems with: (i) 2 controlled intersections, a decision vector of dimension 13 and a budget of 750, and (ii) 9 controlled intersections, a decision vector of dimension 51 and a budget of 3000).

Recent work has investigated what type of structural information provided by the analytical traffic model is key to identifying signal plans that perform well for networks with high levels of congestion and intricate traffic patterns (Osorio et al.; 2014). The latter work considered a Manhattan case study. It indicates that providing the algorithm with an analytical description of between-link interactions or dependencies is critical to identifying signal plans that can contribute to mitigate congestion. In particular, the algorithms are particularly efficient when they are provided with an analytical description of the occurrence and the impact of vehicular spillbacks. The analytical traffic model formulated in Chapter 3 builds upon these insights. It proposes a time-dependent description of spillback probabilities, and it accounts for the impact of spillbacks on the underlying link's flow capacity.

These metamodel ideas have been recently used to efficiently address two other classes of optimization problems. First, metamodels have been formulated for an offline demand calibration problem for the metropolitan area of Berlin (Zhang et al.; 2017). The formulated analytical traffic model is shown to provide an accurate approximation of the unknown simulation-based objective function. The time-

dependent model formulated in Chapter 3 can be used to extend the ideas in Zhang et al. (2017) for online calibration problems. This calibration work also illustrates the use of these metamodel ideas for problems where the decision variables are demand, rather than supply, variables. In particular, in that work the decision variables are coefficients of a route choice model. Second, metamodels have been formulated to design multi-model SO algorithms, where multiple simulators with different computational runtime costs are jointly used (Osorio and Selvam; forthcoming).

This past work highlights that these metamodel ideas have been successfully used to address problems with intricate objective functions, intricate traffic patterns (e.g., Manhattan), and very different demand-supply interactions. This encourages us to design algorithms for real-time problems. The time-dependent formulation proposed in Chapter 3 is a first-step towards this goal.

B.2 Trust Region Subproblem

The appendix serves to describe the formulation of the trust region (TR) problem described in Chapter 3, Section 3.3.3. To formulate the trust region (TR) subproblem

that is solved at each iteration k of the SO algorithm, we use the following notation.

x	vector of decision variables;
x_ℓ	vector of green splits for time interval ℓ;
$x_\ell(j)$	green split of signal phase j for time interval ℓ;
x_{LB}	vector of lower bounds for green splits;
x_k	current iterate at iteration k;
$\mu_{n,\ell}$	service rate of lane n for time interval ℓ;
z	vector of endogenous variables;
p	vector of exdogenous parameters;
β_k	vector of metamodel parameters at iteration k;
Δ_k	trust region radius at iteration k;
c_i	cycle time for intersection i;
d_i	fixed cycle time of intersection i;
e_n	ratio of fixed green time to cycle time of signalized lane n;
s	satuation flow rate;
$\mathcal{N}$	set of indices of the signalized lanes;
$\mathscr{P}_N(n)$	set of endogenous phase indices of lane n;
$\mathscr{I}$	set of intersection indices;
$\mathscr{P}_I(i)$	set of endogenous signal phase indices of intersection i.

The TR subproblem is formulated as follows:

$$\min_x \; m_k(x, z; p, \beta_k) \tag{B.1}$$

subject to

$$\sum_{j \in \mathscr{P}_I(i)} x_\ell(j) = \frac{c_i - d_i}{c_i} \; \forall i \in \mathscr{I}, \ell \in \mathcal{L} \tag{B.2}$$

$$h(x, z; p) = 0 \tag{B.3}$$

$$\mu_{n,\ell} - \sum_{j \in \mathscr{P}_N(n)} x_\ell(j) s = e_n s, \; \forall n \in \mathcal{N}, \ell \in \mathcal{L} \tag{B.4}$$

$$\|x - x_k\|_2 \leq \Delta_k \tag{B.5}$$

$$z \geq 0 \tag{B.6}$$

$$x_\ell \geq x_{LB}, \forall \ell \in \mathcal{L}. \tag{B.7}$$

The objective function is the metamodel $m_k(x, z; \beta_k, p)$. Equations (B.2) and (B.7) are the signal control constraints, they correspond to Equations (3.13) and (3.14). The function h of Equation (B.3) represents the transient network model. It represents Equations (3.7a)-(3.7c). Equation (B.4) associates the green splits of a phase with the flow capacity of the underlying lanes (i.e., the service rate of the queues). Constraint (B.5) is the trust region constraint, where Δ_k is the trust region radius. The endogenous variables of the queueing model are subject to positivity constraints (Equation (B.6)). Thus, the TR subproblem consists of a nonlinear objective function subject to nonlinear equalities, linear equalities, a nonlinear inequality and bound constraints.

The TR subproblem is solved with the Matlab routine for constrained nonlinear problems, *fmincon*, and its interior point programming method (Coleman and Li; 1996, 1994). We set the tolerance for relative change in the objective function to 10^{-3} and the tolerance for the maximum constraint violation to 10^{-3}. For further details on the TR subproblem formulation and its implementation, see Chapter 2, Section 2.3.2.

B.3 Derivative of the Objective Function

This appendix serves to illustrate how the analytical expressions for the objective and constraint functions of Problem (3.12)-(3.14) are derived. The problem is described in Chapter 3, Section 3.3.

The example we provide is for the derivative of the objective function (denoted $f_{A,\ell}$ in Equation (3.15)) with regards to the variable $\hat{\rho}_{i,\ell}$. The analytical form of any other derivative with respect to any of the endogenous variables can be derived following the same logic.

By definition, we have:

$$f_{A,\ell} = \frac{\sum_i E_\ell[N_i]}{\frac{1}{t_\ell - t_{\ell+1}} \int_{t_\ell}^{t_{\ell+1}} \sum_i \gamma_i (1 - P_\ell(N_i = k_i, t)) dt}. \tag{B.8}$$

Let C denote the numerator and D denote the denominator. Following the quotient rule, we obtain:

$$\frac{\partial f_{A,\ell}}{\partial \hat{\rho}_{i,\ell}} = \frac{\frac{\partial C}{\partial \hat{\rho}_{i,\ell}}}{D} - \frac{\frac{\partial D}{\partial \hat{\rho}_{i,\ell}} C}{D^2}. \tag{B.9}$$

We first derive the analytical expression for $\frac{\partial C}{\partial \hat{\rho}_{i,\ell}}$, then we derive that of $\frac{\partial D}{\partial \hat{\rho}_{i,\ell}}$.

Using the chain rule, we have:

$$\frac{\partial C}{\partial \hat{\rho}_{i,\ell}} = \frac{\partial E_\ell[N_i]}{\partial \rho_{i,\ell}} \frac{\partial \rho_{i,\ell}}{\partial \hat{\rho}_{i,\ell}}. \tag{B.10}$$

The analytical form of $E_\ell[N_i]$ and $\rho_{i,l}$ are given in Equations (3.17) and (3.18), respectively.

Equation (3.17) can be rewritten as:

$$E_\ell[N_i] = \frac{\rho_{i,\ell}}{1 - \rho_{i,\ell}} - \frac{(k_i + 1)\rho_{i,\ell}{}^{k_i+1}}{1 - \rho_{i,\ell}{}^{k_i+1}}, \tag{B.11}$$

and its derivative is given by:

$$\frac{\partial E_\ell[N_i]}{\partial \rho_{i,\ell}} = \frac{1}{(1-\rho_{i,\ell})^2} - \frac{(k_i+1)^2 \rho^{k_i}}{(1-\rho_{i,\ell}^{k_i+1})^2}. \tag{B.12}$$

Equation (3.18) can be rewritten as:

$$\rho_{i,\ell} = (t_{\ell+1} - t_\ell)\frac{\hat{\rho}_{i,\ell}}{A}, \tag{B.13}$$

where A is defined by Equation (3.19). This leads to:

$$\frac{\partial \rho_{i,\ell}}{\partial \hat{\rho}_{i,\ell}} = (t_{\ell+1} - t_\ell)(\frac{1}{A} - \frac{\hat{\rho}_{i,\ell}}{A^2}\frac{\partial A}{\partial \hat{\rho}_{i,\ell}}). \tag{B.14}$$

In order to derive an expression for $\frac{\partial A}{\partial \hat{\rho}_{i,\ell}}$, we use the analytical expression of A given by Equation (3.22), this leads to:

$$\begin{aligned}\frac{\partial A}{\partial \hat{\rho}_{i,\ell}} &= \frac{\partial \tau_{i,\ell}}{\partial \hat{\rho}_{i,\ell}}\left[(P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))(e^{-\frac{t_{\ell+1}}{\tau_{i,\ell}}} - e^{-\frac{t_\ell}{\tau_{i,\ell}}})\right] + \dots \\ &\cdots + \tau_{i,\ell}(P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))\left(\frac{\partial e^{-\frac{t_{\ell+1}}{\tau_{i,\ell}}}}{\partial \hat{\rho}_{i,\ell}} - \frac{\partial e^{-\frac{t_\ell}{\tau_{i,\ell}}}}{\partial \hat{\rho}_{i,\ell}}\right) \qquad \text{(B.15)} \\ &= \frac{\partial \tau_{i,\ell}}{\partial \hat{\rho}_{i,\ell}}\left[(P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))(e^{-\frac{t_{\ell+1}}{\tau_{i,\ell}}} - e^{-\frac{t_\ell}{\tau_{i,\ell}}})\right] + \dots \\ &\cdots + (P_\ell(N_i = k_i, t_\ell) - P_\ell(N_i = k_i))\left(\frac{t_{\ell+1}}{\tau_{i,\ell}}e^{-\frac{t_{\ell+1}}{\tau_{i,\ell}}}\frac{\partial \tau_{i,\ell}}{\partial \hat{\rho}_{i,\ell}} - \frac{t_\ell}{\tau_{i,\ell}}e^{-\frac{t_\ell}{\tau_{i,\ell}}}\frac{\partial \tau_{i,\ell}}{\partial \hat{\rho}_{i,\ell}}\right). \qquad \text{(B.16)}\end{aligned}$$

Equation (B.15) is obtained by observing that the derivative of $(t_{\ell+1} - t_\ell)(1 - P_\ell(N_i = k_i))$ with regards to $\hat{\rho}_{i,\ell}$ is zero. This is because the terms $t_{\ell+1}$ and t_ℓ are constant, and because $\frac{\partial P_\ell(N_i = k_i)}{\partial \hat{\rho}_{i,\ell}} = 0$, since both $P_\ell(N_i = k_i)$ and $\hat{\rho}_{i,\ell}$ are both endogenous variables. And because $P_\ell(N_i = k_i, t_\ell)$ is the initial spillback probably at interval ℓ and does not depend on $\hat{\rho}_{i,\ell}$, $\frac{\partial P_\ell(N_i = k_i, t_\ell)}{\partial \hat{\rho}_{i,\ell}} = 0$.

In order to derive $\frac{\partial \tau_{i,\ell}}{\partial \hat{\rho}_{i,\ell}}$, let us recall that $\tau_{i,\ell}$ is defined in Equation (3.11b) by:

$$\tau_{i,\ell} = \frac{ck_i}{\hat{\lambda}_{i,\ell}}\frac{\hat{\rho}_{i,\ell}}{(1 - \sqrt{\hat{\rho}_{i,\ell}})^2}. \tag{B.17}$$

Therefore,

$$\frac{\partial \tau_{i,\ell}}{\partial \hat{\rho_{i,\ell}}} = \frac{ck_i}{\hat{\lambda}_{i,\ell}} \partial \left(\frac{\hat{\rho}_{i,\ell}}{(1-\sqrt{\hat{\rho}_{i,\ell}})^2} \right) / \partial \hat{\rho}_{i,\ell} \tag{B.18}$$

$$= \frac{ck_i}{\hat{\lambda}_{i,\ell}} \left(\frac{1}{(1-\sqrt{\hat{\rho}_{i,\ell}})^2} - \frac{\hat{\rho}_{i,l} \frac{\partial((1-\sqrt{\hat{\rho}_{i,\ell}})^2)}{\partial \hat{\rho}_{i,\ell}}}{(1-\sqrt{\hat{\rho}_{i,\ell}})^4} \right) \tag{B.19}$$

$$= \frac{ck_i}{\hat{\lambda}_{i,\ell}} \left(\frac{1}{(1-\sqrt{\hat{\rho}_{i,\ell}})^2} + \frac{\sqrt{\hat{\rho}_{i,\ell}}}{(1-\sqrt{\hat{\rho}_{i,\ell}})^3} \right). \tag{B.20}$$

Note that Equation (B.18) is derived by observing that $\frac{\partial \hat{\lambda}_{i,\ell}}{\partial \hat{\rho}_{i,\ell}} = 0$, since both variables are endogenous.

Combining Equations (B.10), (B.12), (B.14), (B.16) and (B.20), we obtain the analytical derivative of the numerator of the objective function, $\frac{\partial C}{\partial \hat{\rho}_{i,\ell}}$.

In order to derive the derivative of the denominator D, note that it can be expressed as the following function of A:

$$D = \frac{1}{t_\ell - t_{\ell+1}} \sum_i \gamma_i A. \tag{B.21}$$

Thus,

$$\frac{\partial D}{\partial \hat{\rho}_{i,\ell}} = \frac{\gamma_i}{(t_{\ell+1} - t_\ell)} \frac{\partial A}{\partial \hat{\rho}_{i,\ell}}. \tag{B.22}$$

The expression for $\frac{\partial A}{\partial \hat{\rho}_{i,\ell}}$ is given by Equation (B.16).

Appendix C

Appendices of Chapter 4

C.1 Simulator Selection Method of the Model-driven Method

This appendix serves to describe in detail the simulator selection method of the online optimization framework introduced in Chapter 4, Section 4.2.2.

We denote $\mathcal{SM}$ the simulator set that consists of simulators with different synthetic OD demand matrices. For each simulator, the simulation time horizon is one time interval. The simulator used by the model-driven method for interval t, i.e., $SM(t)$, is selected based on real-time dataset $\mathcal{DA}_t$. The selection method is given as follows:

$$SM(t) = \text{argmin}_{i \in \mathcal{SM}} |PM_i - PM_r|, \tag{C.1}$$

where PM_r represents a user defined performance metric measured from real-time dataset $\mathcal{DA}_t$, PM_i is the corresponding performance metric measured from the simulated data generated by simulator SM_i, and $|PM_i - PM_r|$ represents the absolute difference between PM_i and PM_r. In the case study of Chapter 4, the performance metric is defined as the average link travel time of the links with sensors.

C.2 Preprocessing Procedure of Anderson's Data-driven Method

This appendix serves to describe in detail the preprocessing procedure of the data-driven method of Anderson (2015, Chap. 6). This method is introduced in Chapter 4, Section 4.2.3.

As mentioned earlier, Anderson (2015, Chap. 6) uses only the data of interval $t-1$, denoted DA_{t-1}, to derive the solution of interval t. DA_{t-1} is a subset of $\mathcal{DA}_t$. In this section, DA_{t-1} is named as *features*, following the nomenclature convention of the classification literature.

C.2.1 Preprocessing: Feature Dimension Reduction

In this step, features DA_{t-1} are transformed into reduced features, denoted DA_{t-1}^{rd}. This step is considered as the feature dimension reduction step, where features that have major impacts on final results are kept and features that have minor impacts are excluded. This is done by using a projection matrix that linearly maps features DA_{t-1} to the reduced features DA_{t-1}^{rd}. This process is described as follows:

$$DA_{t-1}^{\text{rd}} = DA_{t-1}\phi, \tag{C.2}$$

where ϕ is known as the *projection matrix*, which produces a linear mapping from original data (original features) to reduced features. Anderson (2015, Chap. 6) uses two types of feature reduction techniques: principle component analysis (PCA) and linear discriminant analysis (LDA). We refer the reader to Anderson (2015, Chap. 6) for a detailed description of these two techniques.

Note that the projection matrix ϕ maps a N dimensional feature to a K dimensional feature (assuming $N > K$), where K is a user-defined dimension. For LDA, K is strictly less than the number of classes. For PCA, the upper bound of K is N. In the case study of the problem where the data-driven method is used to address a

KC-class classification problem, we specify K to be $KC - 1$ for LDA and K to be KC for PCA.

C.2.2 Preprocessing: Rounding and Scaling

In this step, Anderson (2015, Chap. 6) proposes a mapping function to transform the reduced features DA^{rd}_{t-1} to integer features DA^{int}_{t-1}. This function uses normalizing, rounding and bounding techniques to transform continuous features to bounded integer features. This function, denoted as ψ, is given as follows:

$$DA^{\text{int}}(i) = \psi(DA^{\text{rd}}(i)) = \min\{\max\{\text{int}[\frac{M}{b(i)^{\max} - b(i)^{\min}}(DA^{\text{rd}}(i) + b(i)^{\min})], 0\}, M\}, \tag{C.3}$$

where $DA^{\text{int}}(i)$ is the i^{th} dimension of the integer feature DA^{int} and $DA^{\text{rd}}(i)$ is the i^{th} dimension of the reduced feature DA^{rd}. $DA^{\text{int}}(i)$ is transformed from $DA^{\text{rd}}(i)$ through function ψ. $b(i)^{\max}$ and $b(i)^{\min}$ are the maximum and minimum values of the i^{th} reduced feature $DA^{\text{rd}}(i)$ of the training data, M is a user defined integer, and int[.] is the rounding operator (to the nearest integer). We omit the time indices of features DA^{int} and DA^{rd} for the sake of simplicity.

Equation (C.3) ensures that continuous features are transformed into bounded integer features that range from 0 to M. M is a case-specific calibration parameter. In our case study, we use $M = 32$. A detailed description regarding the choice of M can be found in Anderson (2015, Chap. 6).

Note that in Anderson (2015, Chap. 6), reduced features are referred to as computational channel CC, and integer features are referred as plan selection parameters PS.

C.3 Creating the Solution Set $\mathcal{D}$ of the Data-driven Method

This appendix serves to describe the procedure to create the solution set $\mathcal{D}$ of the data-driven method in the case study described in Chapter 4, Section 4.4.3.

In our case study, since signal plans with good performance under the six demand scenarios (i.e., the demand scenarios represented by simulator set $\mathcal{SM}$) are not available, we first generate a number of signal plans that can potentially perform well under these scenarios, named as *candidate plans*, and then choose a subset of these plans to create set $\mathcal{D}$. This is because $\mathcal{D}$ only includes best plans. Therefore, if a candidate plan is not the best plan for any of the scenarios, it should not be included in $\mathcal{D}$.

In this work, we derive candidate plans using the simulation-based algorithm of Chapter 2, although using other optimization methods is entirely possible. Then, we use simulation to determine which candidate plan is the best under each demand scenario. We now describe these steps in detail.

To derive the candidate plan for scenario d_i, denoted sp_i, we run the optimization algorithm proposed in Chapter 2 (only once, with one initial point) using simulator SM_i, which is the simulator that corresponds to demand scenario d_i. We denote $\mathcal{SP}$ the set that consists of all sp_is. For more information on this optimization algorithm, we refer the reader to Chapter 2.

From set $\mathcal{SP}$, we determine signal plans that perform the best under at least one scenario. These plans are then included in set $\mathcal{D}$. To find the best plan for demand scenario d_i, we run simulator SM_i to evaluate the performance of all signal plans in set $\mathcal{SP}$. Denote sp_i^{opt} the best plan of scenario d_i, which is given as:

$$sp_i^{opt} = \text{argmin}_{j \in \mathcal{SP}} F_{SM}(sp_j; d_i) \tag{C.4}$$

where sp_i^{opt} represents the best signal plan for scenario d_i, j represents the index of a signal plan in set $\mathcal{SP}$ and $F_{SM}(sp_i; d_i)$ represents the simulated performance metric

of signal plan sp_j under demand scenario d_i.

The performance metric $F_{SM}(sp_j; d_i)$ should be calculated from data. In this chapter, in order to be consistent with the objective function, we use the simulated average link travel time of the links with sensors as the performance metric.

Once the performance of all signal plans of scenario d_i is obtained, we determine the best signal plan under scenario d_i (i.e., sp_i^{opt}) according to Equation (C.4). In addition, we conduct paired t-tests to determine whether the performance of plan sp_i^{opt} is statistically significantly better than the other plans in set $\mathcal{SP}$. Plan $sp_i^{opt]}$ is then included in set $\mathcal{D}$.

C.4 Classification Model Offline Training Procedure and Results

The appendix serves to describe the offline training procedure and the training results of the data-driven method illustrated in the case study described in Chapter 4, Section 4.4.3.

Once training data is obtained, we first preprocess training data following Equations (C.2) and (C.3). We then train the data-driven model with the preprocessed data.

We use two feature reduction techniques (i.e., LDA nad PCA) in the preprocessing step. This provides two projection matrices ϕ, (see Equation (C.2)) named as ϕ_{LDA} and ϕ_{PCA}, respectively. For each reduction technique, same as Anderson (2015, Chap. 6), we use three classification methods, i.e., decision trees, random forest and k-nearest neighbors. The leads to six classification models (three with LDA projection and three with PCA projection). For a brief description of the three classification models (decision trees, random forest and k-nearest neighbors), we refer the reader to Anderson (2015, Chap. 6).

We then train these six classification models with the preprocessed data $[DA^{\text{int}}, Y]$. Firstly, we use cross-validation procedure to fine-tune parameters of each model and

Table C.1: Misclassification error (percentage)

classification model	LDA	PCA
decision tree	14.0	29.3
random forest	<5	<5
k-nearest neighbors	12.7	31.6

then create six classification models. Secondly, from the six models, we choose three with low misclassification rates as the models to be used online.

We use standard five-fold cross-validation procedure to fine-tune the parameter(s) of each model, i.e., the best leaf size for the decision tree, the best ensemble size for the random forest and the best k value for the k-nearest neighbors. To determine the best parameter(s) of each model, we use the misclassification rate of validation data as the evaluation metric, which is defined as the percentage of wrong predictions in the validation set made by a classification model. As a result, the best parameter value can yield the minimum misclassification rate. Once these parameters are fine-tuned, we use all the training data to build a classification model. The misclassification rate of each model is shown in Table C.1.

Table C.1 shows that random forest method with LDA and random forest method with PCA yield low misclassification rates (less than five percent). When comparing between the two feature reduction techniques (comparing between two columns), LDA is a better feature reduction technique. For instance, the decision tree and k-nearest neighbors models with LDA perform better than their counterparts. This finding is consistent with that of Anderson (2015, Chap. 6).

We use the best three models in the data-driven method: random forest method with LDA, random forest method with PCA and k-nearest neighbors with LDA. As mentioned in Chapter 4, Section 4.4.3, we use the majority vote of the three models to decide the signal plan of a traffic scenario.

Bibliography

Abbas, M. M. and Sharma, A. (2006). Multiobjective plan selection optimization for traffic responsive control, *Journal of transportation engineering* **132**(5): 376–384.

Abdel-Rahim, A. and Dixon, M. (2007). Guidelines for designing and implementing traffic control systems for small- and medium-sized cities in Idaho, *Technical Report N06-18*, Idaho Department of Transportation.

Aboudolas, K., Papageorgiou, M. and Kosmatopoulos, E. (2007). Control and optimization methods for traffic signal control in large-scale congested urban road networks, *American Control Conference*, pp. 3132–3138.

Aboudolas, K., Papageorgiou, M., Kouvelas, A. and Kosmatopoulos, E. (2010). A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks, *Transportation Research Part C: Emerging Technologies* **18**(5): 680 – 694.

Almeder, C., Preusser, M. and Hartl, R. (2009). Simulation and optimization of supply chains: alternative or complementary approaches?, *OR spectrum* **31**(1): 95–119.

Anderson, L. A. (2015). Data-driven methods for improved estimation and control of an urban arterial traffic network.

Antoniou, C. (2004). *On-line calibration of dynamic traffic assignment*, PhD thesis, Massachusetts Institute of Technology.

Antoniou, C., Koutsopoulos, H. N. and Yannis, G. (2013). Dynamic data-driven local traffic state estimation and prediction, *Transportation Research Part C* **34**: 89–107.

ATAC (2003). Signal coordination strategies final report, *Technical report*, Advanced Traffic Analysis Center, Upper Great Plains Transportation Institute, North Dakota State University.

Balakrishna, R. (2002). *Calibration of demand simulator in a dynamic traffic assignment system*, Master's thesis, Massachusetts Institute of Technology.

Barros, J., Araujo, M. and Rossetti, R. J. (2015). Short-term real-time traffic prediction methods: a survey, *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*, IEEE, pp. 132–139.

Barton, R. and Meckesheimer, M. (2006). Metamodel-based simulation optimization, *in* S. G. Henderson and B. L. Nelson (eds), *Handbooks in operations research and management science: Simulation*, Vol. 13, Elsevier, Amsterdam, chapter 18, pp. 535–574.

Ben-Akiva, M., Cuneo, D., Hasan, M., Jha, M. and Yang, Q. (2003). Evaluation of freeway control using a microscopic simulation laboratory, *Transportation research Part C* **11**(1): 29–50.

Bocharov, P., D'Apice, C., Pechinkin, A. and Salerno, S. (2004). *Queueing theory*, Modern Probability and Statistics, Brill Academic Publishers, Zeist, The Netherlands, chapter 3, pp. 96–98.

Branke, J., Goldate, P. and Prothmann, H. (2007). Actuated traffic signal optimization using evolutionary algorithms, *Proceedings of the 6th European Congress and Exhibition on Intelligent Transport Systems and Services.*

Bullock, D., Johnson, B., Wells, R., Kyte, M. and Li, Z. (2004). Hardware-in-the-loop simulation, *Transportation Research Part C* **12**(1): 73 – 89.

Cascetta, E., Inaudi, D. and Marquis, G. (1993). Dynamic estimators of origin-destination matrices using traffic counts, *Transportation Science* **27**(4): 363–373.

Chen, X. (2014). *Traffic signal control in congested urban networks: simulation-based optimization approach*, PhD thesis, University of Coimbra.

Chen, X., Osorio, C., Marsico, M., Talas, M., Gao, J. and Zhang, S. (2015). Simulation-based adaptive traffic signal control algorithm, *Proceedings of the Transportation Research Board Annual Meeting*, Washington DC, USA.

Chiu, S. and Chand, S. (1993). Adaptive traffic signal control using fuzzy logic, *Fuzzy Systems, 1993., Second IEEE International Conference on*, IEEE, pp. 1371–1376.

Chong, L. and Osorio, C. (forthcoming). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems, *Transportation Science* . Forthcoming. Available at: http://web.mit.edu/osorioc/www/papers/osoChoDynSOsubmitted.pdf .

Cohen, J. W. (1982). *The single server queue*, Applied mathematics and mechanics, North-Holland Publishing Company, The Netherlands, chapter III.6.

Coleman, T. F. and Li, Y. (1994). On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds, *Mathematical Programming* **67**(2): 189–224.

Coleman, T. F. and Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds, *SIAM Journal on Optimization* **6**: 418–445.

Conn, A. R., Gould, N. I. M. and Toint, P. L. (2000). *Trust-region methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, USA.

Conn, A. R., Scheinberg, K. and Vicente, L. N. (2009a). Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points, *SIAM Journal on Optimization* **20**(1): 387–415.

Conn, A. R., Scheinberg, K. and Vicente, L. N. (2009b). *Introduction to derivative-free optimization*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, PA, USA.

Dinopoulou, V., Diakaki, C. and Papageorgiou, M. (2006). Applications of the urban traffic control strategy tuc, *European Journal of Operational Research* **175**(3): 1652–1665.

Dumont, A. and Bert, E. (2006). Simulation de l'agglomération Lausannoise SIMLO, *Technical report*, Laboratoire des voies de circulation, ENAC, Ecole Polytechnique Fédérale de Lausanne.

Fu, M. C., Glover, F. W. and April, J. (2005). Simulation optimization: a review, new developments, and applications, *in* M. E. Kuhl, N. M. Steiger, F. B. Armstrong and J. A. Joines (eds), *Proceedings of the 2005 Winter Simulation Conference*, Piscataway, New Jersey, USA, pp. 83–95.

Gartner, N. and Hou, D. (1992). Comparative evaluation of alternative traffic control strategies, *Transportation Research Record* (1360).

Hachicha, W., Ammeri, A., Masmoudi, F. and Chachoub, H. (2010). A comprehensive literature classification of simulation optimisation methods, *Proceedings of the International Conference on Multiple Objective Programming and Goal Programming MOPGP10*, Sousse, Tunisia.

Hale, D. (2005). Traffic network study tool - TRANSYT-7F, *Technical report*, McTrans Center in the University of Florida, Gainesville, Florida.

Hasan, M. (1999). *Evaluation of ramp control algorithms using a microscopic traffic simulation laboratory, MITSIM*, Master's thesis, Massachusetts Institute of Technology.

Hasan, M., Jha, M. and Ben-Akiva, M. (2002). Evaluation of ramp control algorithms using microscopic traffic simulation, *Transportation Research Part C* **10**(3): 229–256.

Heidemann, D. (2001). A queueing theory model of nonstationary traffic flow, *Transportation Science* **35**(4): 405–412.

Joshi, S., Rathi, A. and Tew, J. (1995). An improved response surface methodology algorithm with an application to traffic signal optimization for urban networks, *in* C. Alexopoulos, K. Kang, W. R. Lilegdon and D. Goldsman (eds), *Proceedings of the Winter Simulation Conference*, Arlington, VA, USA, pp. 1104–1109.

Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis, G. V. and Eversdyk, D. (2004). A simulation based optimization approach to supply chain management under demand uncertainty, *Computers and Chemical Engineering* **28**: 2087–2106.

Kleijnen, J. P. C. (2008). Response surface methodology for constrained simulation optimization: An overview, *Simulation Modelling Practice and Theory* **16**(1): 50–64.

Kleijnen, J. P. C., van Beers, W. and van Nieuwenhuyse, I. (2010). Constrained optimization in expensive simulation: Novel approach, *European Journal of Operational Research* **202**(1): 164–174.

Legato, P., Mazza, R. M. and Trunfio, R. (2008). Simulation-based optimization for the quay crane scheduling problem, *in* S. Mason, R. Hill, L. Moench and O. Rose (eds), *Proceedings of the Winter Simulation Conference*, pp. 2717–2725.

Li, P., Abbas, M., Pasupathy, R. and Head, L. (2010). Simulation-based optimization of maximum green setting under retrospective approximation framework, *Transportation Research Record* **2192**: 1–10.

Lin, S. (2011). *Efficient model predictive control for large-scale urban traffic networks*, PhD thesis, Delft University of Technology.

Little, J. (1961). A proof for the queuing formula: L= λ W, *Operations Research* **9**(3): 383–387.

Little, J. (2011). Little's law as viewed on its 50th anniversary, *Operations Research* **59**(3): 536–549.

Liu, H., Han, K., Gayah, V. V., Friesz, T. L. and Yao, T. (2015). Data-driven linear decision rule approach for distributionally robust optimization of on-line signal control, *Transportation Research Part C* **59**: 260–277.

Marti, K. (2008). *Stochastic optimization methods*, Springer, Berlin, Germany.

Morse, P. (1958). *Queues, inventories and maintenance; the analysis of operational systems with variable demand and supply*, Wiley, New York, USA.

Nakamiti, G. and Gomide, F. (1996). Fuzzy sets in distributed traffic control, *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, Vol. 3, IEEE, pp. 1617–1623.

NYCDOT (2012). Downtown Flushing,mobility and safety improvemnt project, *Technical report*, New York City Department of Transportation.

Odoni, A. and Roth, E. (1983). An empirical investigation of the transient behavior of stationary queueing systems, *Operations Research* **31**(3): 432–455.

Osorio, C. (2010). *Mitigating network congestion: analytical models, optimization methods and their applications*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne.

Osorio, C. and Bierlaire, M. (2009). An analytic finite capacity queueing network model capturing the propagation of congestion and blocking, *European Journal of Operational Research* **196**(3): 996–1007.

Osorio, C. and Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems, *Operations Research* **61**(6): 1333–1345.

Osorio, C., Chen, X., Marsico, M., Talas, M., Gao, J. and Zhang, S. (2014). Reducing gridlock probabilities via simulation-based signal control, *Proceedings of the International Symposium on Transport Simulation (ISTS)*. Under review for journal publication, version available at: http://web.mit.edu/osorioc/www/papers/osoChenNYCDOTOfflineSO.pdf .

Osorio, C., Chen, X. and Santos, B. F. (2016). Simulation-based travel time reliable signal control. Submitted. Available at: http://web.mit.edu/osorioc/www/papers/osoCheSanReliableSO.pdf .

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

Osorio, C. and Flötteröd, G. (2014). Capturing dependency among link boundaries in a stochastic dynamic network loading model, *Transportation Science* **49**(2): 420–431.

Osorio, C., Flötteröd, G. and Bierlaire, M. (2011). Dynamic network loading: a stochastic differentiable model that derives link state distributions, *Transportation Research Part B* **45**(9): 1410–1423.

Osorio, C. and Nanduri, K. (2015a). Energy-efficient urban traffic management: a microscopic simulation-based approach, *Transportation Science* **49**(3): 637–651.

Osorio, C. and Nanduri, K. (2015b). Urban transportation emissions mitigation: Coupling high-resolution vehicular emissions and traffic models for traffic signal optimization, *Transportation Research Part B* **81**: 520–538.

Osorio, C. and Selvam, K. (forthcoming). Simulation-based optimization: achieving computational efficiency through the use of multiple simulators, *Transportation Science* . Forthcoming. Available at: http://web.mit.edu/osorioc/www/papers/osoSelMultiModel.pdf .

Papageorgiou, M., Messmer, A., Azema, J. and Drewanz, D. (1995). A neural network approach to freeway network traffic control, *Control Engineering Practice* **3**(12): 1719–1726.

Papayannoulis, V., Marsico, M., Maguire, T., Strasser, J. and Scalici, S. (2011). An integrated travel demand, mesoscopic and miroscopic modeling platform to assess traffic operations for Manhattan, New York, *Proceedings of the Transportation Research Board (TRB) Conference*, Washington DC, USA.

Park, B., Yun, I. and Ahn, K. (2009). Stochastic optimization for sustainable traffic signal control, *International Journal of Sustainable Transportation* **3**(4): 263–284.

Peterson, M. D., Bertsimas, D. J. and Odoni, A. R. (1995). Decomposition algorithms for analyzing transient phenomena in multiclass queueing networks in air transportation, *Operations Research* **43**(6): 995–1011.

Rathi, A. K. and Lieberman, E. B. (1989). Effectiveness of traffic restraint for a congested urban network: a simulation study, *Transportation Research Record* **1232**: 95–102.

Riniker, K., Eisenach, P. and Hannan, T. (2009). City of Winchester, VA traffic signal upgrade project, *Technical report.*

Schrank, D., , Lomax, T. and Bak, J. (2015). TTI's 2015 urban mobility report, *Texas A&M Transportation Institute. The Texas A&M University System* .

Schwartz, J. D., Wang, W. and Rivera, D. E. (2006). Simulation-based optimization of process control policies for inventory management in supply chains, *Automatica* **42**: 1311–1320.

Søndergaard, J. (2003). *Optimization using surrogate models - by the Space Mapping technique*, PhD thesis, Technical University of Denmark.

Srinivasan, D., Choy, M. C. and Cheu, R. L. (2006). Neural networks for real-time traffic signal control, *IEEE Transactions on Intelligent Transportation Systems* **7**(3): 261–272.

Stafford, R. (2006). *The Theory Behind the 'randfixedsum' Function.* http://www.mathworks.com/matlabcentral/fileexchange/9700 .

Stallard, C. and Owen, L. (1998). Evaluating adaptive signal control using CORSIM, *in* D. Medeiros, E. Watson, J. Carson and M. Manivannan (eds), *Proceedings of the 1998 Winter Simulation Conference*, Washington, D.C., USA, pp. 1147–1154.

Stevanovic, A., Stevanovic, J., Zhang, K. and Batterman, S. (2009). Optimizing traffic control to reduce fuel consumption and vehicular emissions. integrated approach with VISSIM, CMEM, and VISGAOST, *Transportation Research Record* **2128**: 105–113.

Stevanovic, J., Stevanovic, A., Martin, P. T. and Bauer, T. (2008). Stochastic optimization of traffic control and transit priority settings in VISSIM, *Transportation Research Part C* **16**(3): 332 – 349.

Toledo, T., Koutsopoulos, H. N., Davol, A., Ben-Akiva, M. E., Burghout, W., Andreasson, I., Johansson, T. and Lundin, C. (2003). Calibration and validation of microscopic traffic simulation tools: Stockholm case study, *Transportation Research Record* **1831**: 65–75.

Traffic Technology International (2012a). *Admirable Solution*, Traffic Technology International. February/March.

Traffic Technology International (2012b). *In the Frame*, Traffic Technology International. April/May.

Trafficware (2011). *Synchro Studio 8 User Guide*, Trafficware, Sugar Land, TX.

TSS (2011). *AIMSUN 6.1 Microsimulator UserâĂŹs Manual*, Transport Simulation Systems.

VSS (1992). *Norme Suisse SN 640837 Installations de feux de circulation; temps transitoires et temps minimaux*, Union des professionnels suisses de la route, VSS, Zurich.

Webster, F. V. (1958). Traffic signal settings, *Technical Report 39*, Road Research Laboratory.

Wild, S. M., Regis, R. G. and Shoemaker, C. A. (2008). ORBIT: Optimization by radial basis function interpolation in trust-regions, *SIAM Journal on Scientific Computing* **30**: 3197–3219.

Xu, H., Caramanis, C. and Mannor, S. (2012). Sparse algorithms are not stable: A no-free-lunch theorem, *IEEE transactions on pattern analysis and machine intelligence* **34**(1): 187–193.

Yun, I. and Park, B. (2006). Application of stochastic optimization method for an urban corridor, *in* L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol and R. M. Fujimoto (eds), *Proceedings of the Winter Simulation Conference*, Monterey, CA, USA, pp. 1493–1499.

Zhang, C., Osorio, C. and Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators, *Transportation Research Part B* **97**: 214–239.