Multi-Model Simulation-Based Optimization applied to Urban Transportation

by

Krishna Kumar Selvam

Dual Degree in Civil Engineering
Indian Institute of Technology Madras, 2012

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

Master of Science in Transportation

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Author .....
Department of Civil and Environmental Engineering
May 21, 2014

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

Accepted by .....
Heidi M. Nepf
Chair, Departmental Committee for Graduate Students

Multi-Model Simulation-Based Optimization applied to Urban Transportation

by

Krishna Kumar Selvam

Submitted to the Department of Civil and Environmental Engineering
on May 21, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science in Transportation

Abstract

Transportation agencies often resort to the use of traffic simulation models to evaluate the impacts of changes in network design or network operations. They often have multiple traffic simulation tools that cover the network area where changes are to be made. Nonetheless, these multiple simulators may differ in their modeling assumptions (e.g., macroscopic versus microscopic), in their reliability (e.g., quality of their calibration) as well as in their modeling scale (e.g., city-scale model versus regional-scale model).

The choice of which simulation model to rely on, let alone of how to combine their use, is intricate. A larger-scale model may, for instance, capture more accurately the local-global interactions; yet may do so at a greater computational cost. This thesis proposes a methodology that enables the simultaneous use of multiple traffic simulation models.

We propose a simulation-based optimization algorithm that embeds information from simulation models with different levels of accuracy and with different levels of computational efficiency. The algorithm combines the use of high-accuracy low-efficiency models with low-accuracy high-efficiency models. This combination leads to an algorithm that can identify points (e.g., network designs, traffic management strategies) with good performance at a reduced computational cost.

We evaluate the performance of the algorithm with a traffic signal control problem on a small network, as well a large-scale city network. We show that the proposed algorithm identifies signal plans with excellent performance, i.e., with reduced average trip travel times, while doing so with a reduction in the computational cost.

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

Acknowledgments

I would like to express the deepest gratitude to my thesis supervisor, Prof. Carolina Osorio for her support and guidance through the past two years. She has been a wonderful mentor and I thank her for the challenging and stimulating research experience. Above all, she has been genuinely supportive of my pursuits outside of my thesis work, both academic and otherwise, and I cannot thank her enough for this.

It has been a wonderful learning experience working with Prof. Cynthia Barnhart on the vehicle sharing project. The manner in which she maintains the balance between her extraordinarily successful career and personal life is something I will always aspire to.

I would like to thank Prof. Karthik Srinivasan at IIT Madras who introduced me to research in Transportation. His course on Transportation Network Analysis is what got me excited about research in the first place, and I am grateful to him for that. I am also grateful to have worked on projects with Prof. Gitakrishnan Ramadurai and Prof. Saravanan at IITM.

I would like to thank Professors Amedeo Odoni, Richard Larson, Peter Belobaba, Moshe Ben-Akiva, Babak Ayazifar, Roy Welsch, Pablo Parrilo, Marta Gonzalez, Abel Sanchez and John Williams for providing me a great learning experience through their courses at MIT.

I am indebted to the New England University Transportation Centers Program, Ford and the Universities and Grants Programs (Federal Highway Administration) for funding my studies and research over the last two years.

I cannot thank my colleague and friend, Linsen Chong, enough for helping me out at several instances and playing the role of the lab senior to perfection. I am glad to have met Kenneth Loh - watching him slog it out during his last few weeks at MIT is something that will inspire me for life. I have learnt a great deal by working on assignments and course projects with motivated friends like Matthew, Kenneth, Setareh, Franco and Yashovardhan. Working with Ta on the Ford vehicle sharing

project also taught me a great deal about paying attention to the details while being efficient at the same time. I am fortunate to have worked on the Informs competition with Setareh, one of the most motivated people I have ever met.

I am thankful to Harshavardhan and Varun for having been amazing role models right from my undergraduate days and to my colleagues from the ITS lab Yichen, Tianli, Yin, Haijeng, Peiguang and others for their great company.

I would like to thank my friends Ankur, Himani, Parnika, Rasha, Rinal, Rushabh and Sanket for being great buddies and making sure I didn't miss my family here. I have grown immensely as a person by interacting with each one of them, and will always cherish the great times we've shared. I'm also glad to have met the Hews street residents and my other friends from Sangam, THRA and Tang.

Last, but by no means the least, I would like to thank my family for their constant motivation and encouragement. Without their hardwork and sacrifice, most of the good things in my life wouldn't have been possible. I am also thankful to my extended family and cousins whom I love dearly.

Contents

List of Figures

Chapter 1

Introduction and Literature Review

1.1 Introduction

Managing urban vehicular traffic is an important and complex problem that cities all over the world are interested in solving, and traffic simulation models have become ubiquitous tools to achieve this. There are a number of commercially available traffic simulation models, each of which is customized to suit a specific aspect of transportation planning and operations. Such models can be broadly classified in terms of how traffic is represented as microscopic or mesoscopic or macroscopic. A detailed review of the different types of simulation models can be found in Ratrout and Rahman (2009). In this work, we focus on deriving traffic signal plans that reduce the average user travel time using microscopic traffic simulation models within an optimization framework.

1.2 Problem Statement

Let $E[f(x;p)]$ represent the objective that needs to be optimized, x represent the decision vector that can be controlled by a transportation professional, and p represent the vector of exogenous parameters that describe the system of interest.For instance,

x could represent traffic signal plans or the design of a vehicle sharing network. The network topology, for instance is a parameter that is beyond the control of a transportation professional in most cases, and could be represented by p. The objective, $E[f(x;p)]$ could be chosen as the expected value of the average delay experienced by users, pollution level etc. as evaluated using the simulation model. The problem can be formulated without loss of generality as follows:

$$\underset{x}{\text{minimize}} \quad E[f(x;p)] \tag{1.1}$$

$$\text{subject to} \quad \mathcal{Z}(x;p) = 0 \tag{1.2}$$

$$x \in \mathbb{R}^N \tag{1.3}$$

As we will describe in later sections, we use a stochastic microsimulation model and hence are interested in the expected value of $f(x;p)$. Here, $\mathcal{Z}(x;p) = 0$ represents the set of constraints that we need to consider (for instance, the sum of the green time allocated to different phases should sum up to the cycle time).

Microsimulation models attempt to model traffic dynamics by modeling the behavior of individual vehicles and are hence computationally intensive. This makes the use of such models in iterative optimization algorithms that involve hundreds of iterations time consuming and highly inefficient. Our goal is to therefore use microsimulation models of varying fidelity within a simulation-based optimization (SO) framework to address this inefficiency. Our approach combines the use of high-accuracy low-efficiency models with low-accuracy high-efficiency models. This combination leads to an algorithm that can identify points (e.g., network designs, traffic management strategies) with good performance at a reduced computational cost.

1.3 Literature Review

In this section, we review how the the transportation community has dealt with issues associated with the computation time of traffic simulation models using hybrid

models. We follow that up with a review of surrogate-based optimization models , and how we propose to apply one such technique within a simulation-based optimization framework.

1.3.1 Microsimulation models

Traffic microsimulation models describe the behavior of individual vehicles in a network, the interaction between different vehicles as well as the interaction between the individual vehicles and the transportation infrastructure (i.e., traffic signal design, width of an intersection). This behavior and the resulting interactions are captured using a set of models that govern every action of a vehicle. For instance, a car-following model governs the acceleration and deceleration patterns of a car, while there are other models that account for lane-changes. Similarly, a route choice model determines how the users decide on the path to their destination, as well as if and when they change their route along the way based on new information regarding the network conditions.

Since each of these models are applied to every vehicle in the network, it results in significantly larger computation times as compared to, for instance, a macroscopic traffic model in which vehicles are aggregated and treated as flows. Another feature of these models, since they try to model human behavior, is that they are stochastic in nature. Therefore, two runs of the same microsimulation model with no change in the inputs or model parameters could lead to very different outcomes. Hence, some applications might require averaging multiple evaluations of the same point using a traffic microsimulator.

Thus, in order to overcome the limitations of large computation times associated with microsimulation models, several attempts have been made to combine their use with models of lower fidelity (macroscopic or mesoscopic). The major hybrid modeling attempts made in the past two decades are summarized in Table 1.1. We describe each of these works in brief, and summarize the key issues associated with the hybrid approach to traffic simulation.

Table 1.1: Hybrid Traffic Models

Model	**Microscopic model**	**Mesoscopic model**	**Macroscopic model**	**Models overlap**	**Models overlap and have feedback**	**Used in an optimization framework**
Bunch et al. (1999)	-	INTEGRATION	EMME/2	Yes	Yes	No
Van Vliet and Hall (1997)	-	SATURN	SATURN	Yes	Yes	No
Montero et al. (1998)	AIMSUN	-	EMME/2	Yes	No	No
Rousseau et al. (2008)	Vissim	-	CUBE, Visum	Yes	No	No
Oh et al. (2000)	PARAMICS	-	DYNASMART	Yes	Yes	No
TransModeler	Inbuilt feature	Inbuilt feature	Inbuilt feature	$-^a$	$-^a$	No
Sewall et al. (2011)	Treiber et al. (2000)	-	ARZ	No	No	No
Bourrel and Lesort (2003)	Newell Optimal Velocity model	-	Strada	No	No	No
Magne et al. (2000)	SITRA-B+	-	SIMRES	No	No	No
Horowitz (2004)	SmartAHS	SmartCAP	-	No	No	No
Burghout (2004)	MITSIMLab	Mezzo	-	Partial overlap^b	Yes	No
Selvam (2014) (This work)	AIMSUN	-	-	No	No	Yes

^{a} these details regarding TransModeler aren't publicly available.
^{b} we consider the virtual links in Burghout (2004) as a partial overlap.

1.3.2 Bunch et al. (1999)

One of the earliest works in using transportation models of different modeling complexities for the same application is reported in Bunch et al. (1999). The authors use a macroscopic planning model (EMME/2) to model a large region. However, they are interested in studying only a subarea of this region, and use a meso scale (mesoscopic) simulation model (INTEGRATION) which covers this subarea in greater detail. The primary reason they chose this approach is to save on computation effort, as is evident from their comment "In theory, one could model the entire region using only a simulation model, but this is not yet practical for desktop PCs and current software".

As a first step, the macroscopic travel demand model is built for the Seattle region. A subarea of the Seattle region is modeled separately using a mesoscopic simulation model. Both the models are validated to be representative of the baseline 1990 time period. The authors study the impact of six different congestion mitigation strategies that are proposed to tackle an expected increase in congestion in the Seattle I-5 North Corridor. These strategies include measures like construction of new roads or lane miles, conventional signal installations, transit improvements, traditional demand management measures, advanced traveler information systems, advanced traffic management systems and advanced public transportation systems (Bunch et al., 1999).

The scenarios that would result from adopting each of the six strategies that are coded into both the macroscopic and the mesoscopic models for the year, 2020 to perform a what-if analysis. Each scenario is first evaluated using the macroscopic model, from which relevant performance measures are recorded. The flows observed in the macroscopic model at the interface of the subarea and the larger region are used to define a demand matrix for the mesoscopic model covering the subarea. The mesoscopic model is then run for each of the scenarios to evaluate the effectiveness of ITS applications using various measures of effectiveness like delay reduction, throughput, coefficient of trip time variation, expected number of stops per vehicle kilometer traveled etc.

The authors note that the differences in assumptions between the macroscopic and the mesoscopic models. While the macroscopic model allows for link volumes greater than the link capacities, the mesoscopic model by design cannot have link volumes greater than the corresponding link capacities. Also, the mesoscopic model explicitly models queuing in links while the macroscopic model uses speed-flow relationships, and does not model the queuing. Hence, absolute values of the link costs are not sent as feedback to the macroscopic simulation model since the links within the subarea and those outside of it have different link costs due to differences in the modeling assumptions. Instead, the relative change in the queuing delays in each of the six strategies with respect to that of the baseline scenario is sent as feedback from the mesoscopic simulation model to the macroscopic demand model. The authors do not describe how this feedback is used by the macroscopic model.

1.3.3 Van Vliet and Hall (1997)

Another early study of simultaneous use of models that have different levels of detail is presented in the user manual of SATURN 9.3 (Van Vliet and Hall, 1997). Users of the software have the option of using a macroscopic static traffic assignment model for the route choice in a large network and a detailed mesoscopic simulation model to model a subset of this large network. The authors prescribe an iterative procedure, in which the macroscopic and mesoscopic model are run sequentially, one after another.

In the macroscopic traffic assignment model, speed flow relationships are used to determine the travel time on links as a function of flow. The link travel times are used to find a set of path flows that are in stochastic user equilibrium. On the other hand, in the mesoscopic simulation model, users also have the option to define the travel time either based on speed-flow relationships or simply use the free-flow travel time. The authors expect the free-flow travel time option to be used more commonly.

The resultant path flows of the macroscopic traffic assignment model are translated into link flows and turning flows, and given as input to the simulation model. The simulation model is then run to determine the delays, capacities and queues for individual turning moments. It also identifies new cost-flow relationships and junc-

tion delays, that are used as inputs to the next iteration of the traffic assignment . Thus, the assignment and simulation model are repeatedly run one after the other, until the percentage variation between link flows of successive assignment runs are less than a threshold.

1.3.4 Montero et al. (1998)

An example of a macroscopic traffic assignment model (EMME/2) being used along with a microscopic simulation model (AIMSUN/2) to evaluate the design of road networks is presented in Montero et al. (1998). The traffic assignment in a large regional network is modeled using EMME/2 and a subarea is chosen for detailed analysis using microsimulation. A total of five different proposed infrastructure projects are evaluated in this exercise.

The EMME/2 model uses volume delay functions to compute a static user equilibrium while the AIMSUN/2 microsimulation model simulates individual vehicles based on behavioral rules (gap acceptance, lane changing etc.). The first step in the integrated framework involves running once the traffic assignment model, which is followed by running the microsimulation model. In a procedure that is similar to the one used in Bunch et al. (1999), the flows from the traffic assignment are used to generate the demand matrix for the microsimulation model. However, in this case there is no feedback from the microsimulation model to the traffic assignment model. The results of the microsumulation model alone are used for evaluation of the performance of different road network designs.

1.3.5 Rousseau et al. (2008)

In Rousseau et al. (2008) , the authors propose a three tier structure to build a high resolution microsimulation model of a subnetwork starting with a macroscopic model of a larger network. The three models used in this framework are :

- Regional model (software: CUBE)
- Subarea macroscopic model with high level of network detail (software: Visum)

- Microscopic model for the subarea (software: Vissim)

The authors propose the following approach to the integration of these three models: The first step is to to identify the subarea boundary (for instance, the downtown area) within the regional model, following which the subarea network is cut out and the demand matrix for this area network is computed. The path flows from the traffic assignment in the regional model are used to generate the demand matrix of the subarea. The subarea network is then refined, by using a street network of higher resolution as compared to the one used in the original model from which it was cut out. This high resolution street network includes side streets and important driveways that are not modeled in the original regional model. It is not clear whether this new data models the existing roads in more detail. Following this, traffic assignment is run once again for the refined subarea network. Intersection data (i.e. geometry and signal control) is then added to the refined network. While the macroscopic model is calibrated using only the link volumes on principal highways, the subarea macroscopic model is calibrated using additional data in the form of turn volume counts along with the data used to calibrate the regional model.

The path flows and the network topology from the refined subarea network are directly exported into a microsimulation software. The microsimulation model has different assumptions compared to the macroscopic ones used so far, including car following, lane changing and gap acceptance models - all these models are then calibrated to come up with a realistic microsimulation model. The next logical step would be to provide feedback from the microscopic model to the macroscopic model. The authors comment that this step is often skipped by practitioners since it would add an additional calibration cycle to an already complex methodology.

1.3.6 Oh et al. (2000)

In Oh et al. (2000), the authors use a microscopic simulation model PARAMICS to describe the movements of individual vehicles. A macroscopic traffic assignment (DYNASMART) model of the full network used in the microsimulation model is used

for route guidance, as described below.

The authors observe that the microscopic simulation is highly sensitive to even minor variations in the road geometry (changing curvature, width etc.). Hence an accurate description of this geometry in the microsimulation model requires the creation of a large number of short links that could otherwise be modeled as a single link in a macroscopic model(as the macroscopic model is not as sensitive to minor changes in the road geometry). As a result, the route choice models that come with the microscopic simulation model PARAMICS do not scale efficiently to large-scale networks due to the detailed network descriptions used in microscopic models. Therefore, for large networks, modeling route choice decisions of drivers as a response to real time information provided by Intelligent Transportation Systems becomes practically impossible.

Hence, the authors create an abstract network that models the important features of the full network used in the microsimulation model, and ignore minor changes in geometry and road characteristics. In the abstract network, the number of nodes isn't significantly larger than the actual number of decision nodes (i.e. intersections in the actual road network where users have to choose from multiple paths to their destination). This abstract network is accurate enough to model the path dynamics using DYNASMART, as it uses macroscopic flows on idealized links.

In the integrated framework, the route choice module in PARAMICS is effectively disabled and replaced by the route choice model from DYNASMART.The microsimulation of vehicle movements in the PARAMICS model is used to compute aggregate link costs and sent to the DYNASMART model at every time step. Every time a vehicle in the PARAMICS model reaches a node where it has the option of choosing from more than one link to go to, a decision routine is run on the DYNASMART model, the result of which informs the vehicle in the PARAMICS model. Thus, the full network is modeled in both DYNASMART and PARAMICS, with the former being used for its efficient route choice models and the latter for its microsimulation capabilities. Although the authors dont present a quantitative comparison of the computation time between the inbuilt route choice model and their proposed approach to

making route choice decisions, they conclude that for large networks, their proposal of using a macroscopic model to make route choice decisions would be advantageous.

1.3.7 TransModeler

The traffic simulation package TransModeler (version 3.0) allows users to describe different sections of the same network as microscopic, mesoscopic or macroscopic models. In the microscopic segments, vehicles are modeled similar to other microsimulation models, using car following and lane changing models. In the mesoscopic section, groups of vehicles are modeled as traffic cells, and speed density relationships are used to determine their movements. The macroscopic model is based on a link performance function similar to the one in static assignment models. The different models are run simultaneously, and the models that run fastest wait for the others at every time step (Burghout, 2004). The details of how the different models interact with each other are not provided in the website that describes these features.

1.3.8 Sewall et al. (2011)

In Sewall et al. (2011) the authors propose a model that has features similar to TransModeler, except that they allow the users to choose between a microscopic and a macroscopic model for different parts of the network and do not provide the mesoscopic option. That is, the road network consists of mutually exclusive regions modeled using either a microscopic or macroscopic model. They use the 'Aw-Rascle-Zhang' (ARZ) model for the macroscopic regions (Aw and Rascle, 2000). For the microscopic regions, they use an extended version of the agent-based simulation model from Treiber et al. (2000). Their version includes behavioral models for 'lane changing, inhomogeneous driver models and vehicle response to traffic signals, intersections and variable speed limits' (Sewall et al., 2011), in addition to the car following model from Treiber et al. (2000).

For a timestep Δt , the computations are first performed for the continuum regions covered by the macroscpoic model; this involves solving partial differental equations.

At the interface between the two regions, the macroscopic densities are then disaggregated into discrete cars, using a procedure called as Poisson instantiation. This process assumes that the spatial distribution of vehicles in a lane follows a Poisson distribution. Now computations for an equal timestep Δt are carried out for the agent-based regions. The next step is to aggregate the discrete cars flowing from the agent-based regions to the continuum regions into densities and flows. While the authors delve into great detail on the aggregation and disaggregation procedures at the interface, they don't explicitly model the a ssignment of traffic in the network. Their model seems to simply track the dynamics of the vehicles on the network after the routes for different origin destination pairs are given as an input.

1.3.9 Bourrel and Lesort (2003)

The hybrid model Hystra (Bourrel and Lesort, 2003) uses a macroscopic and a microscopic model that are both based on the LWR (Lighthill-Whitham-Richards) traffic flow theory. The macroscopic model is the Strada model (Buisson et al., 1996). The authors, in an attempt use a microscopic model that is also based on the LWR model choose the Newell optimal velocity model (Newell, 1961). The authors impose the condition that the time step used in the microscopic model is much smaller than the same in the macroscopic model.

In the hybridization scheme proposed in Hystra, the road network consists of mutually exclusive regions modeled using either a microscopic or macroscopic model. In order to accurately transfer information from the macroscopic model to the microscopic model and vice versa at the boundary at every time step, the authors use a 'transition cell' at the interface to split up the transition process.

In a transition cell connecting a macroscopic segment to a microscopic segment, the flow from the macroscopic model is dis-aggregated into vehicles in the microscopic region. The arrival times of these newly created vehicles are distributed uniformly within the corresponding time period. Similarly, in a transition cell connecting a microscopic segment to a macroscopic segment, the vehicles exiting the microscopic segment are aggregated into macroscopic parameters (like flow and density). The

macroscopic segment has an upper limit on the number of vehicles that are allowed to enter within a time step. This condition is satisfied in the aggregation step by imposing a minimum exit gap for vehicles that leave a microscopic segment and enter a macroscopic one. The vehicles that try to exit with a smaller gap are delayed.

This model is tested on a single lane that consists of a microscopic region sandwiched between two macrosocopic regions. The vehicles first enter the macroscopic region and then transition into the microscopic region through a transition cell where the macroscopic flow is disaggregated into individual vehicles. Finally, the vehicles re-enter the macroscopic region through a transition cell that aggregates individual vehicles into macroscopic flow. Based on the observations from this test, authors conclude that their model correctly translates the boundary conditions between the two types of model used.

1.3.10 Magne et al. (2000)

The MICMAC model in Magne et al. (2000) combines the microscopic model SITRA-B+ and the macroscopic model SIMRES. The SIMSRES model is based on the METANET model from Messner and Papageorgiou (1990), in which the links are discretized into cells for which quantities like flow, density and concentration are calculated for each of the time steps. The authors state that both the microscopic and macroscopic models in this framework need to satisfy the boundary conditions on the flow-density function and its slope for certain critical values of the density. The authors therefore use a modified version of the SITRA-B+ model as the original SITRA-B+ model does not satisfy all of the aforementioned constraints.

The road network in the MICMAC model consists of mutually exclusive regions modeled using either the microscopic or the macroscopic model (similar to Hystra). The disaggregation of flow from a macroscopic cell to a microscopic cell is performed using a Poisson distribution for the time gaps. Similarly, the aggreagation process (from microscopic to macroscopic segments) of discrete vehicles into traffic flow variables like flow, density and speed involves averaging these parameters from the microscopic model. The macroscopic timestep is much larger than the microscopic one,

and hence, in order to synchronize the two models, every timestep of the macroscopic model includes multiple updates to the microscopic regions.

This model is tested on a single link with three lanes that consists of a macroscopic segment followed by a microscopic segment. The results are compared to a model in which the entire link is modeled as a macroscopic segment. While the results of the experiment with the original SITRA-B+ model shows deviation from the full-macroscopic model, the use of the modified SITRA-B+ model produced results that are similar to the full-macroscopic model. One of the drawbacks of this model for our purpose is that they do not describe if or how the travel times from regions modelled using different scales are aggregated to model the route-choice of users. They have only published results for simple cases where there is only one path (as in the case of the experiment with the single link) from the origin to the destination.

1.3.11 Horowitz (2004)

The authors developed an interface between the SmartCAP mesoscopic model (Broucke et al., 1996) and the SmartAHS microsimulation model (Deshpande et al., 1997) to model Automated Highway Systems (AHS). In an Automated Highway System, vehicles are not controlled by the drivers, but by algorithms that group vehicles into platoons. Such an automated system is expected to reduce delays and eliminate driver errors. SmartCAP uses a conservation model of traffic density along with traffic flow behavior models of car following, lane changing etc. This model is specifically built for modeling AHS, and aggregates traffic quantities by modeling the highway system as a series of segments that are approximately 0.25 miles long. SmartAHS is similar to most microsimulation models and describes the movement of each and every vehicle in the network. Different parts of the network are modeled using either of the two models.

Similar to MICMAC and HYSTRA, the authors propose aggregation and disaggregation schemes. More details on the aggregation and disaggregation procedure are available in Horowitz (2004).

1.3.12 Burghout (2004)

This hybrid model, MiMe, integrates the microscopic simulation model MITSIMLab (Ben-Akiva et al., 1997) with Mezzo (Burghout, 2004), a mesoscopic simulation model. This model also allows the user to model different regions of a network using either the microscopic or the mesoscopic model.

The author proposes the use of mesoscopic 'virtual links', which are essentially abstractions of the subpaths in the microscopic areas of the network. That is, for each pair of entry and exit nodes in the subnetworks that are microsimulated, there exists a virtual link for each used path. This consistent representation allows for pre-trip route choices to be made using the mesoscopic model. On the other hand, the presence of microscopic virtual links in the mesoscopic region allows for changing route choice decisions en-route, after a vehicle has entered the microscopic region. The author also proposes a novel method for generating initial velocities in the disaggregation phase (i.e converting flows from upstream mesoscopic segments into individual vehicles in the downstream microscopic links)

1.3.13 Discussion

Most of the hybrid models described so far involve the use of a higher fidelity microsimulation model with a macroscopic or a mesoscopic model (the two exceptions being Bunch et al. (1999) and Van Vliet and Hall (1997) where a mesoscopic model is used as the higher fidelity model along with a low fidelity macroscopic model).Subnetworks of interest are modeled with the high fidelity model while the rest of the network is modeled using the low fidelity model. Besides the issue of data requirements for detailed representation of large networks, a major reason for not modeling a large region using only the high fidelity (microscopic) model is to reduce the computation time.

The hybrid models can broadly be classified into two categories. One category allows the user to model different regions of the same network using different models(Transmodeler, Sewall et al. (2011), Bourrel and Lesort (2003),Magne et al. (2000), Horowitz (2004), Burghout (2004)) . Much of the past work in this category focuses

on the issues of aggregation and disaggregation at the boundaries of the different types of models to ensure consistency, since the representation of traffic is different in different kinds of models. For instance, a lot of attention has been paid to the conversion of flow and density from an upstream macroscopic link into discrete vehicles in a downstream microscopic link. Similarly, the aggregation of vehicles into cumulative metrics of flows and densities when they move from a microscopic region to a downstream macroscopic segment is another important topic that most of these models focus on. The main drawback of these hybrid models for our purpose is that they do not discuss the impact of hybridization on route choice and traffic assignment. The aggregation of travel times from models that have different modeling assumptions is a key issue that affects traffic assignment, and it is omitted in all these models. For a signal plan design problem, this is an important issue since the traffic assignment is highly sensitive to changes in the signal plan. The notable exception among the models discussed so far is Burghout (2004) which uses a meso-micro combination and takes into account several aspects of assignment like pre-trip route choices and en-route assignment. However, this hybrid model only works with a combination of mesoscopic and microscopic simulation. It cannot be extended to a macroscopic model that is not simulation based (for instance, a travel demand model).

The other type of hybrid models allow for the use of two separate models of the same region. For instance Bunch et al. (1999), Van Vliet and Hall (1997) , Montero et al. (1998) and Rousseau et al. (2008) model a large region using the low fidelity (often macroscopic) model. Only a subnetwork of interest is then separately modeled using a high fidelity (microscopic/mesoscopic) model. This means that there are regions of the network that are modeled using both a high fidelity and a lower fidelity model, unlike the first kind where each region is modeled using exactly one model.

Typically, the traffic assignment performed in the macroscopic model is used to generate the OD matrix for the high fidelity model that covers a smaller area. The high fidelity model is then run using this OD matrix and the results are analyzed. In a few of these models, there is feedback of representative metrics of delays and travel times from the high fidelity model to the macroscopic model. Several models do not

even attempt to provide feedback since the traffic assignment in the larger region becomes unstable (i.e. the path flows fluctuate as a result of the feedback) when delays and travel times from the microscopic/mesoscopic assignment are considered. Even those that provide feedback do not explicitly use travel times, and have to work around this instability in order to achieve convergence of path flows.

It is also important to note that the second category of hybrid models described above is a natural fit for variable fidelity optimization since two different models are used to describe the same region of interest. Hence, there are two models, either of which can be used to obtain an estimate of the objective function. This property of overlapping models is very useful to us as it ensures that each run of the subarea model need not be accompanied by a run of the larger model as part of a feedback loop.

1.4 Surrogate Models

One approach to overcome the computational barrier in large-scale optimization problems that use simulation models is to use surrogate-based optimization, the state of the art of which is reviewed in Robinson (2007), as well as Forrester and Keane (2009). In these methods, the computationally expensive model is not used to evaluate every iterate.In certain iterations, a less expensive model of lower fidelity is used instead. The lower fidelity model takes less computation time at the cost of being not as accurate as the high-fidelity model. For instance in an iterative optimization algorithm, the less expensive model would be used in a majority of the iterations, while computationally expensive high fidelity model would be used at a much lesser frequency. Surrogate models can be roughly classified into three categories (Eldred and Dunlavy (2006) and Robinson (2007)). It needs to be noted that this is only a rough classification, and several models could fall in both the first and the second categories.

The first category of data fit surrogates includes statistical models built using samples of $f(x;p)$ at one or more points. Queipo et al. (2005) provide a detailed review

of the different variations of this class of methods, both in terms of the functional form (polynomial regression, Kriging, radial basis functions) as well as the design of experiments (Latin Hypercube Sampling(LHS), orthogonal arrays, optimal LHS etc).

The second category of surrogate models are reduced order models in which the original high-dimensional system is projected down to a lower dimensional space. These are built using techniques like principal component analysis, spectral decomposition and the like. Since these are still physics based models, they have better predictive qualities than data fit surrogates (Eldred and Dunlavy, 2006).

The third category of surrogate models have a hierarchical structure. These are also called variable fidelity (or) multi-fidelity models, in which g(x;p) is a lower fidelity physics based model. It could be of lower fidelity due simpler modeling assumptions (for instance a coarser grid of the finite element mesh in the case of structural optimization), or simply the higher fidelity model with relaxed tolerance for convergence (Robinson, 2007). Several applications of this procedure can be found in Carter (1986), Alexandrov et al. (1999), Sun et al. (2010) , and Huang et al. (2006). This is the class of surrogate models to which our work belongs.

1.4.1 Surrogate-based optimization and transportation

We consider the situation where a transportation agency has access to two microscopic simulation models that cover the network of interest and that both have the same modeling assumptions (i.e. same demand and supply models). Let R denote the larger-scale simulation model (R stands for regional) and C denote the smaller scale simulation model (R stands for city-center). We assume that the subnetwork of interest is the full network of C . This subnetwork and the rest of the region is entirely modeled in R. This is a scenario that one can easily encounter in practice, where R is an available large-scale model, and C is a smaller model extracted from R, and calibrated based on R outputs. Model R is assumed to lead to more accurate estimates of both local and global performance, yet it is significantly more expensive to evaluate as indicated in Table 1.2. The primary focus of our work is to propose a methodology that enables to choose one of the two models (the efficient inaccurate

model C or the inefficient accurate model R) to evaluate the performance of a given point.

Table 1.2: Description of the simulation models used

Model	**Region covered**	**Computational cost**	**Accuracy**
C	subnetwork	low	low
R	full network	high	high

The framework described in Osorio and Bierlaire (2013) combined the use of an analytical traffic model with a stochastic microscopic traffic simulator. This general idea has been used to address a variety of simulation-based signal control problems including large-scale problems (Osorio and Chong, 2014), emissions and fuel-efficient problems (Osorio and Nanduri, 2014b,a), as well as travel time reliability problems (Chen et al., 2013). The proposed work uses the simulation-based optimization algorithm in Osorio and Bierlaire (2013) and extends it in order to allow for the use of multiple traffic simulators.

Figure 1-1 describes the basic outline of this SO algorithm. The first step is to define a deterministic and analytical metamodel that tries to mimic the stochastic simulator. This metamodel is optimized to obtain a new trial point. The new trial point is then evaluated using the simulator, and the result is used to improve the accuracy of the metamodel. Hence, as the algorithm progresses, the metamodel becomes more representative of the objective function, and is expected to lead to points with improved performance.

Our work embeds a multi-fidelity optimization technique as part of this simulation-based optimization framework. Every time a trial point is to be evaluated using the simulator, we pick between the two available ones, based our prediction of the accuracy of the estimate obtained using C with respect to that using R.

To summarize, multi-fidelity optimization techniques have been applied widely in optimizing structural design, especially in the field of aircraft design to overcome

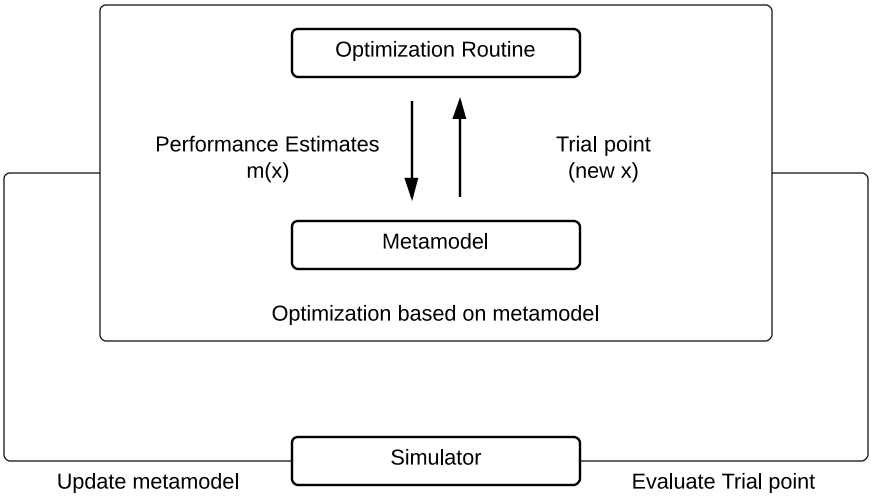


Figure 1-1: The SO algorithm used in Osorio and Bierlaire (2013), adapted from Alexandrov et al. (1999)

computational challenges. The use of simulation-based optimization for optimizing transportation designs is a new field, and our work embeds a multi-fidelity optimization technique as part of this simulation-based optimization framework in order to tackle a large-scale design problem. While we adopt a framework similar to traditional multi-fidelity optimization techniques, we propose to use structural information from the problem to identify with greater accuracy the trial points for which the lower fidelity model is a reasonable substitute for the high fidelity model.

Chapter 2

Methodology

2.1 Introduction

Let the objective function be denoted by $E[f^R(x)]$. In our case this is the average user travel time from origin to destination evaluated using model R. The aim is to derive a transportation strategy (e.g., a signal control plan or a network design alternative, hereafter called a point) that provides an improvement in $E[T_{sub}]_R$. We assume we have a fixed simulation run time budget, hereafter referred to as the computational budget. The objective is to identify a point with improved performance within this budget using model R and model C within a simulation-based optimization framework. There are three possible approaches:

- Use only model R
- Use only model C
- Use a combination of model R and model C

If there are no constraints on the computational budget, the first technique will definitely lead to a transportation strategy with improved performance. However, if the computational budget is limited, then this method might not work since model R takes longer to execute, and we might not have sufficient number of runs of model R to obtain a metamodel that is accurate enough to bring an improvement in performance. Therefore, this strategy will evaluate the performance of only a few points within the

computational budget. The second strategy enables the largest number of simulation runs within the budget yet may not lead to a signal plan with improved performance when evaluated with SR, as every run of SC will yield an estimate of $E[T_{sub}]_C$, which may be different from $E[T_{sub}]_R$, the performance measure we want to optimize. This difference is due to the fact that the demand-supply modeled in model C might not be an accurate representation of the traffic assignment in model R.

The third strategy is the one proposed in this thesis. It attempts to reach a trade-off between the two other strategies. We consider a traditional fixed-time signal control problem. We assume we calibrate model C based on the outputs of SR and do so for a given signal plan (e.g., calibration of behavioral parameters, of origin-destination (OD) matrix). This is done once, before starting the optimization algorithm. Extensions of this framework may calibrate model C iteratively as more observations from model R are collected throughout the optimization. At each iteration of the SO algorithm, the main decision to be made is which simulation model to call (model R or model C).

Note: In the remainder of the thesis, $E[T_{sub}]_R$ will be referred to as $E[f^R(x)]$. Similarly $E[T_{sub}]_C$ will be referred to as $E[f^C(x)]$.

2.2 The simulation-based optimization framework

We propose a modified version of the framework proposed in Osorio and Bierlaire (2013). The proposed methodology integrates multiple simulation models within the basic SO framework. The details are given in this section.

2.2.1 Problem definition

Notation:

r_i	ratio of all red time to cycle time in intersection i
b_i	available cycle ratio of intersection i ($b_i = 1 - r_i$)
x_L	vector of minimum green splits for each phase

$x(j)$	green split of phase j (i.e. the green time of phase j divided by the cycle time of its corresponding intersection)
n_c	length of the decision vector x, where $x = [x(1), x(2), ..., x(n_c)]$
s	saturation flow rate [vehicles/hour]
μ_ℓ	capacity of lane ℓ [vehicles/hour]
$\mathcal{I}$	set of intersection indices
$\mathcal{L}$	set of indices of the signalized lanes
$\mathcal{P}_I(i)$	set of phase indices of intersection i
$\mathcal{P}_L(l)$	set of phase indices of lane l
$f^C(x)$	user travel time in the subnetwork evaluated for the signal plan x using model C (i.e. the objective function evaluated using model C)
$f^R(x)$	user travel time in the subnetwork evaluated for the signal plan x using model R (i.e. the objective function evaluated using model R).

$$\underset{x}{\text{minimize}} \quad E[f^R(x)] \tag{2.1}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{P}_I(i)} x(j) = b_i, \forall i \in \mathcal{I} \tag{2.2}$$

$$x \geq x_L \tag{2.3}$$

The objective function is the expected value of the user travel time in the subnetwork evaluated for the signal plan x using model R. Equation (2.2) is a relation between the green times of different phases of an intersection with the available cycle time. The second constraint (Equation (2.3)) ensures that minimum green times are allocated to each of the phases.

2.2.2 The modified SO algorithm

Notation:

x_k	current iterate at iteration k

$\hat{f}^C(x_k)$	estimate of user travel time in the subnetwork corresponding to the signal plan x_k from one run of model C
$\hat{f}^R(x_k)$	estimate of user travel time in the subnetwork corresponding to the signal plan x_k from one run of model R
β_k	the metamodel parameters of iteration k
u_k	the number of successive trial points rejected
m_k	metamodel in iteration k
Δ_k	trust region radius in iteration k
d_k	an indicator of the decision taken at iteration k. $d_k = 0$ indicates model R was chosen, and $d_k = 1$ indicates model C was chosen in this iteration
n_k	sample size used in fitting the metamodel in iteration k
τ_k	relative change in the parameters of the metamodel between successive iterations
ω_k	improvement in the simulated values between successive iterations
ξ_k	relative improvement in the simulated values between successive iterations with respect to the improvement in the metamodel predictions

The different steps in the framework are given below:

- Step 0: Initialization. Set
  - an initial point x_0 (Section 3.2),
  - an upper bound for the trust region radius, $\Delta_{max} > 0$,
  - an initial trust region radius $\Delta_0 \in (0, \Delta_{max}]$,
  - the maximum number of function evaluations n_{max},
  - the parameters $\eta_1, \nu, \nu_{inc}, \bar{\tau}, \bar{u}$ such that
    - $0 < \eta_1 < 1$
    - $0 < \nu < 1 < \nu_{inc}$
    - $0 < \bar{\tau} < 1$
    - $\bar{u} \in \mathbb{N}$
  - the threshold $\delta > 0$, used while choosing between the simulation models R and C.
  - Evaluate $\hat{f}_R(x_0)$, fit an initial model m_0, and compute β_0

The numerical values of different parameters used are given in Appendix C.

- Step 1: Step calculation

  Solve the trust region subproblem to compute a step s_k that sufficiently reduces the metamodel $m_k(x)$. The problem formulation is the same as the one used in Osorio and Bierlaire (2013), the details of which are provided here.

$$\underset{x}{\text{minimize}} \quad m_k(x) \tag{2.4}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{P}_I(i)} x(j) = b_i, \forall i \in \mathcal{I} \tag{2.5}$$

$$||x - x_k||_2 \leq \Delta_k \tag{2.6}$$

$$x \geq x_L \tag{2.7}$$

  The objective is to minimize the metamodel, by choosing an x that lies within the trust region, while ensuring that minimum green times are allotted to each of the phases.

- Step 2: Selection of simulation model

  Choose the simulation model with which to evaluate the new trial point $x_k + s_k$. Refer to Section 2.3 for details on how the choice is made. At the end of this step,

$$d_k = \begin{cases} 0 & \text{if model } R \text{ is chosen} \\ 1 & \text{if model } C \text{ is chosen} \end{cases} \tag{2.8}$$

- Step 3: Acceptance of trial point

  Evaluate the point $x_k + s_k$ using the chosen simulation model to obtain $\hat{f}_{d_k}(x_k + s_k)$. If d_k is set as '0' in step 2, the signal plan $x_k + s_k$ is coded into model R, which is then run to obtain $\hat{f}_0(x_k + s_k)$. If d_k is set as '1', the signal plan $x_k + s_k$

is coded into model C, which is then run to obtain $\hat{f}_1(x_k + s_k)$. Compute

$$\xi_k = \begin{cases} \frac{\hat{f}^R(x_k) - \hat{f}^R(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)} & \text{if } d_k = 0 \\ \frac{\hat{f}^C(x_k) - \hat{f}^C(x_k+s_k)}{m_k(x_k) - m_k(x_k+s_k)} & \text{if } d_k = 1 \end{cases} \tag{2.9}$$

$$\text{and } \omega_k = \begin{cases} \hat{f}^R(x_k) - \hat{f}^R(x_k + s_k) & \text{if } d_k = 0 \\ \hat{f}^C(x_k) - \hat{f}^C(x_k + s_k) & \text{if } d_k = 1 \end{cases} \tag{2.10}$$

There are two possibilities:

- $d_k = d_{k-1}$. That is, the simulation model used in iteration k is the same as the one used in iteration $k-1$. In this case, we would have already evaluated x_k using the model chosen by d_k in iteration $k-1$.
- $d_k \neq d_{k-1}$. In this case, we also need to evaluate x_k using the model chosen by d_k as it is needed in the computation of ξ_k and ω_k. Set $n_k = n_k + 1$.

If $\xi_k \geq \eta_1$ and $\omega_k > 0$, then accept the trial point. Set $x_{k+1} = x_k + s_k$ and $u_k = 0$. Else, reject the trial point. Set $x_{k+1} = x_k$ and $u_k = u_k + 1$.

Include the new observation in the sample set ($n_k = n_k + 1$) and fit the new model m_{k+1} (thereby obtaining the metamodel parameter vector β_{k+1}). We use a quadratic metamodel as described in section 2.2.3.

- Step 4: Model improvement

Compute

$$\tau_{k+1} = \frac{||\beta_{k+1} - \beta_k||}{\beta_k} \tag{2.11}$$

If $\tau_{k+1} < \bar{\tau}$, then improve the model by evaluating a point x, which is chosen randomly in the feasible region. The simulation model R is used for evaluating

this point. Include this point x in the sample set(Set $n_k = n_k + 1$.), and update m_{k+1}.

- Step 5: Trust region radius update

  If $\rho_k > \eta_1$, then increase the trust region radius:

$$\Delta_{k+1} = \min\{\nu_{inc}\Delta_k, \Delta_{max}\} \tag{2.12}$$

  Otherwise

  – If $u_k > \bar{u}$, then reduce the trust region radius:

$$\Delta_{k+1} = \max(\nu\Delta_k, \Delta_{min}) \tag{2.13}$$

  – Else if $u_k \leq \bar{u}$, then $\Delta_{k+1} = \Delta_k$.

Set $n_{k+1} = n_k$, $u_{k+1} = u_k$ and $k = k + 1$. If $n_k < n_{max}$ go to Step 1; otherwise, stop.

2.2.3 Fitting the metamodel

The quadratic metamodel used in the SO algorithm is given below.

$$m_k(x, \beta_k) = \beta_{1,k} + \sum_{i=1}^{n_c} \beta_{1+i,k} x(i) + \sum_{i=1}^{n_c} \beta_{n_c+1+i,k} x(i)^2 \tag{2.14}$$

The parameters of the metamodel, $\beta_k = (\beta_{i,k})_{i=1,2,\ldots,2n_c+1}$ are estimated using a weighted least squares method, as shown in Equation (2.15) . Let $\mathcal{A}_k$ correspond to the set of points evaluated with model R and $\mathcal{N}_k$ correspond to the same for model C, at iteration k.

$$\underset{\beta_k}{\text{minimize}} \sum_{j=1}^{|\mathcal{A}_k|} \{\psi_j(\hat{f}^R(x_j) - m_k(x_j, \beta_k))\}^2 + \tfrac{1}{10} \sum_{j=1}^{|\mathcal{N}_k|} \{\psi_j(\hat{f}^C(x_j) - m_k(x_j, \beta_k))\}^2 + \sum_{j=1}^{2n_c+1} (\psi_0\beta_j)^2 \tag{2.15}$$

The first squared term in Equation (2.15) corresponds to the weighted distance between the simulations from model R of points from $\mathcal{A}_k$ and the metamodel predictions , while the second squared term is the weighted distance between the simulations from model R of points from $\mathcal{N}_k$ and the metamodel predictions. In this metamodel, errors corresponding to the estimates obtained from model C are weighted down by a factor of 10, as compared to the errors corresponding to the estimates obtained from model R, in view of the inaccuracy of model C with respect to model R; hence we multiply the second term by $\frac{1}{10}$.

The weights ψ_j denote the significance of each of the points in $\mathcal{A}_k$ and $\mathcal{N}_k$ with respect to the current iterate x_k. We use the inverse distance weight function, with Euclidean distance, leading to the following definition of the weights:

$$\psi_j = \frac{1}{1 + ||x_k - x_j||_2} \tag{2.16}$$

This definition of weights is identical to that used by Osorio (2010) for the estimation of metamodel parameters. Such a definition gives greater importance to points near the current iterate x_k.

The third term arises due to a set of $2n_c + 1$ artificial (or augmented) data points which we add to the data used to fit the metamodel. These points ensure that the number of data points used to fit metamodel is greater than or equal to the number of parameters (i.e., $|\beta_k|$) we are fitting. This ensures that the least squares matrix is of full rank, and hence the uniqueness of the parameters. This minimization problem is solved using the Matlab routine *lsqlin*. Since these are artificial points, their impact on the parameters is reduced by using a small value of 0.1 for the fixed weight ψ_0 . The details of the parameters used in the *lsqlin* routine are given in Appendix D.

2.3 Choice of simulation model

Since the computation time of model C is much lesser than that of model R, we choose model C to evaluate all x when we can expect that

$$|\hat{f}^R(x) - \hat{f}^C(x)| < \delta, \tag{2.17}$$

where δ is a parameter that is tuned a priori. That is, if we can predict that model C will return a similar value of the objective function to model R for the signal plan corresponding to x, then we choose the less expensive model C.

The value of $|f^R(x) - f^C(x)|$ shall be denoted as $e(x)$. Our methodology is as follows.

- For a given signal plan x_k, approximate the value of $e(x_k)$. Let the approximation be denoted as $\hat{e}(x_k)$.
- If $\hat{e}(x_k) < \delta$ set $d_k = 1$ (i.e. choose model C). Otherwise, set $d_k = 0$. (i.e. choose model R).

In order to compute $\hat{e}(x)$, we use the analytical model of Osorio and Bierlaire (2009) to describe the congestion in the network and a multinomial logit model to describe the route choice behavior of users in the network. The results of the traffic assignment are a key input to computing $\hat{e}(x)$.

2.3.1 Modeling congestion

We build upon the analytical queuing model presented in Osorio (2010, Chap. 4), in which each lane of a road section is modeled as an M/M/1/ℓ queue, where ℓ is the space capacity of the queue. This space capacity is an upper bound on the queue-length, and is used to capture the propagation of congestion. Given the network structure, the arrival rates and the turning probabilities, this model can be used to obtain the delays and queue length distributions of all the queues in the network.

For a given queue i, the following notation is used.

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;
ρ_i	traffic intensity;
P_i^f	probability of being blocked after service at queue i;
ℓ_i	space capacity;
N_i	number of vehicles in queue i;
$P(N_i = \ell_i)$	probability of queue i being full;
p_{ij}	transition probability from queue i to queue j;
$\mathcal{D}_i$	set of downstream queues of queue i
$E[T_i]$	travel cost of queue i;

$E[N_i]$	number of vehicles in queue i;
l^{veh}	average vehicle length;
$v^{freeflow}$	free flow speed;

For a given road network represented as a queueing network, the marginal queue-length distributions of each queue are obtained by simultaneously solving for all queues the following system of equations.

$$\lambda_i = \gamma_i + \frac{\sum_j p_{ji}\lambda_j P(N_j < \ell_j)}{P(N_i < \ell_i)} \tag{2.18}$$

$$\frac{1}{\hat{\mu}_i} = \frac{1}{\mu_i} + P_i^f \frac{1}{\tilde{\mu}_i} \tag{2.19}$$

$$\frac{1}{\tilde{\mu}_i} = \sum_{j \in \mathcal{D}_i} \frac{\lambda_j P(N_j < \ell_j)}{\lambda_i P(N_i < \ell_i)\hat{\mu}_j} \tag{2.20}$$

$$P(N_i = \ell_i) = \frac{1 - \rho_i}{1 - \rho_i^{\ell_i + 1}} \rho_i^{\ell_i} \tag{2.21}$$

$$P_i^f = \sum_j p_{ij} P(N_j = \ell_j) \tag{2.22}$$

$$\rho_i = \frac{\lambda_i}{\hat{\mu}_i} \tag{2.23}$$

We briefly describe the interpretation of these equations. Equation (2.18) describes the conservation of flow between upstream and downstream queues. For queue i, its total arrival rate, λ_i, is related to its external arrival rate, γ_i and to the arrivals arising from upstream queues (second-term in the right-hand side of the equation). Equation (2.19) describes the service process of a vehicle, which is composed of two phases. First, the vehicle undergoes an initial service. The queue has an underlying service rate, μ_i, that is determined by its underlying supply (e.g., flow capacity of the downstream intersection). After receiving service, a vehicle that is at queue i and is ready to proceed to queue j may do so if queue j is not full. If queue j is full (i.e., if there is a spillback at queue j), then the vehicle is forced to remain at queue i. This is known in queueing theory as blocking. This occurs with probability P_i^f and this second service is referred to as blocking time, the expected blocked time is given by $1/\tilde{\mu}_i$. Equation (2.20) describes the expected blocking time, which is a function of the effective service rate of downstream queue j, $\hat{\mu}_j$. Equation (2.21) describes the probability that queue i is full, it is also known as the blocking probability. In vehicular traffic this represents the spillback probability. The expression of Equation (2.21) is obtained by assuming that queue i is an M/M/1/ℓ queue (e.g., Bocharov et al., 2004). Equation (2.22) describes the probability that a vehicle at queue i gets blocked (i.e., that it cannot proceed downstream of queue i due to downstream spillbacks). Equation (2.23) describes the traffic intensity, which is a ratio of demand to supply.

The following algebraic manipulations are performed on the above equations in order to make the Jacobian of these equations simpler to compute.

Equation (2.18) is multiplied by $P(N_i <= \ell_i)$ and $\lambda_i(P(N_i <= \ell_i))$ is denoted as λ_i^{eff}. Equation (2.19) is multiplied by λ_i^{eff}; $\frac{\lambda_i^{eff}}{\tilde{\mu}_i}$ is denoted as $\tilde{\rho}_i$ and $\frac{\lambda_i^{eff}}{\hat{\mu}_i}$ is denoted as ρ_i^{eff}. Equation (2.20) is multiplied by λ_i^{eff} . Equations (2.21) and (2.22) are left unchanged. In the right hand side of Equation (2.23) , $\frac{\lambda_i}{\hat{\mu}_i}$ is replaced by $\rho_i^{eff} \frac{1}{1-P(N_i=\ell_i)}$

as the two are equivalent.

The modified set of equations is given below:

$$\lambda_i^{eff} = \gamma_i(1 - P(N_i = \ell_i)) + \sum_{j=1}^{n} p_{ji}\lambda_j^{eff} \tag{2.24}$$

$$\rho_i^{eff} = \frac{\lambda_i^{eff}}{\mu_i} + P_i^f \tilde{\rho}_i \tag{2.25}$$

$$\tilde{\rho}_i = \sum_{j \in D_i} \rho_i^{eff} \tag{2.26}$$

$$P(N_i = \ell_i) = \frac{1 - \rho_i}{1 - \rho_i^{\ell_i+1}} \rho_i^{\ell_i} \tag{2.27}$$

$$P_i^f = \sum_{j=1}^{n} p_{ij} P(N_j = \ell_j) \tag{2.28}$$

$$\rho_i = \rho_i^{eff} \frac{1}{1 - P(N_i = \ell_i)} \tag{2.29}$$

The queue length and travel time on different queues are computed as follows:

$$E[N_i] = \frac{\rho_i}{1 - \rho_i} - \frac{(\ell_i + 1)\rho_i^{\ell_i+1}}{1 - \rho_i^{\ell_i+1}} \tag{2.30}$$

$$E[T_i] = \frac{E[N_i]}{\lambda_i^{eff}} + \frac{l^{veh}(\ell_i - E[N_i])}{v^{freeflow}} \tag{2.31}$$

The expected value of the queue length is defined as a function of traffic intensity in Equation (2.30), and the average delay experienced by a vehicle in queue i is obtained by applying Little's Law for that queue, and is equal to $\frac{E[N_i]}{\lambda_i^{eff}}$. A derivation of Equation (2.30) can be found in Osorio and Chong (2014).

The total travel time experienced by a vehicle in a lane is approximated by the sum of the free flow travel until the beginning of the physical vehicular queue and the delay before it exits the queue, as described in Equation (2.31).The second term in Equation (2.31) is an approximation of the travel time to reach the physical queue of vehicles: the numerator approximates the available road-space length not occupied by a stationary vehicular queue, the denominator is the roads free-flow speed. The

constants l^{veh} and $v^{freeflow}$ are assigned values of 4 meters and 60 kilometers per hour respectively. We assume that the free flow speed is the same on all queues, since the simulation model we use also has the same maximum speed on all links.

The main limitation of the model of Osorio (2010) for the purpose of our work is that it assumes exogenous turning probabilities, p_{ij} and arrival rates γ_i. In this work, the purpose of the analytical model is to approximate how subnetwork boundary conditions may change due to changes in supply. More specifically, we want to approximate how the OD matrix of the subnetwork changes due to changes in the subnetwork signal plans. Hence, accounting for endogenous assignment is necessary. Hence, we consider the turning probabilities and arrival rates as endogenous and derive them using a multinomial logit route choice model.

2.3.2 Modeling route choice

The simulation model we use models the route choice behavior according to a stochastic user equilibrium, and we use a similar route choice model in the analytical formulation. The microsimulation models that we use enumerate the first k_s shortest paths between every origin-destination (OD) pair, and then assign flow on these paths in such a way that the probability of a path being chosen from among the different alternatives between that OD pair decreases with increasing travel time. The details are given below.

Notation:

d_s	demand corresponding to OD pair s
c_t	travel time on path t
y_t	expected flow on path t
r_{ti}	proportion of flow on path t that goes through queue i
a_{ti}	indicates whether path t contains queue i
a^*_{ti}	indicates whether the first link of path t contains queue i
z_{ij}	indicates whether queue i and queue j are parallel queues within the same

	link (i.e., parallel lanes)
l_{st}	probability that a vehicle travelling the OD pair s takes path t
θ	route choice probability parameter
$\mathcal{S}$	set of OD pairs
$\mathcal{Q}$	set of queue indices
$\mathcal{T}$	set of paths indices
$\mathcal{P}_s$	set of paths of OD pair s
$\mathcal{G}_{ij}$	set of paths that consecutively go through queues i and j
$\mathcal{H}_i$	set of paths that go through queue i

$$c_t = \sum_{i \in \mathcal{Q}} r_{ti} E[T_i] \quad \forall t \in \mathcal{T} \tag{2.32}$$

$$l_{st} = \frac{e^{-\theta c_t}}{\sum\limits_{j \in \mathcal{P}_s} e^{-\theta c_j}} \quad \forall s \in \mathcal{S}, \quad \forall t \in \mathcal{P}_s \tag{2.33}$$

$$y_t = \sum_{s \in \mathcal{S}} d_s l_{st} \quad \forall t \in \mathcal{T} \tag{2.34}$$

$$\gamma_i = \sum_{t \in \mathcal{T}} r_{ti} a^*_{ti} y_t \quad \forall i \in \mathcal{Q} \tag{2.35}$$

$$p_{ij} = \frac{\sum\limits_{t \in \mathcal{G}_{ij}} f_t}{\sum\limits_{u \in \mathcal{H}_i} f_u} \quad \forall i \in \mathcal{Q}, \quad \forall j \in \mathcal{Q} \tag{2.36}$$

The indicator variables are defined as follows.

$$a_{ti} = \begin{cases} 1 & \text{if queue } i \text{ is part of path } t, \\ 0 & \text{otherwise.} \end{cases} \tag{2.37}$$

$$a^*_{ti} = \begin{cases} 1 & \text{if queue } i \text{ is part of the first link (road) of path } t, \\ 0 & \text{otherwise.} \end{cases} \tag{2.38}$$

$$z_{ij} = \begin{cases} 1 & \text{if queue } i \text{ and queue } j \text{ are parallel lanes in the same link,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.39}$$

$$r_{ti} = \frac{a_{ti}}{\sum_{j \in \mathcal{Q}} a_{tj} z_{ij}} \qquad \forall t \in \mathcal{T}, \forall i \in \mathcal{Q} \tag{2.40}$$

The route cost is defined by Equation (2.32) as the expected travel time for route t. The route travel time is a function of queue travel time $E[T_i]$, which is given by Equation (2.31). The path choice probability is given by the multinomial logit expression of Equation (2.33). The deterministic component of the utility function for a given route t is defined as a function of a single (exogenous) parameter θ and the costs of that particular route. Equation (2.34) defines the flow along a path t as a function of the total demand of a given OD-pair s, denoted d_s, and the probability of choosing path t for OD-pair s, denoted l_{st}. Note that the OD-pair demand d_s of the full R network is exogenous, and obtained from the OD matrix of the R model. Equation (2.35) gives the expression for the external arrival rate of queue i, γ_i. Equation (2.36) defines the probability of turning from queue i to queue j as the ratio of the total flow along paths that have queues i and j as consecutive queues and of the total flow that goes through queue i. In the model of Osorio (2010), this rate is exogenous. In this work, since we account for endogenous assignment, the external arrival rates of a given queue depend on the (endogenous) path choice probabilities, and hence are themselves endogenous.

We use the network shown in Figure 2-1 to illustrate how the parameters a_{ti}, a^*_{ti} and z_{ij} are defined. In this network, there are two paths between the origin O and destination D. The first path (on top) consists of two links, shown in different colors. The first link of this path has a complicated geometry - it is two lanes wide at the beginning and narrows down to a single lane. Each lane is modeled as an $M/M/1/\ell$ queue - this network has a total of 6 queues.

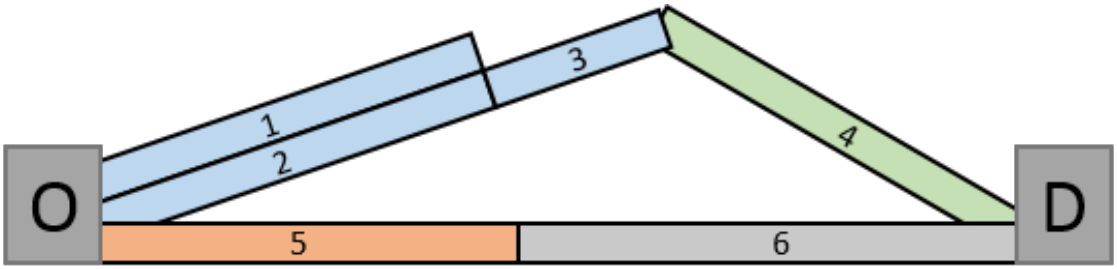


Figure 2-1: An example network

For this network, $a_{1i} = 1, \forall i \in \{1, 2, 3, 4\}$ while $a_{1i} = 0, \forall i \in \{5, 6\}$. Similarly, considering the second path, $a_{2i} = 1, \forall i \in \{5, 6\}$ while $a_{1i} = 0, \forall i \in \{1, 2, 3, 4\}$. $a^*_{11} = 1$, $a^*_{12} = 1$ and $a^*_{1i} = 0, \forall i \in \{3, 4, 5, 6\}$. Similarly, considering the second path, $a^*_{25} = 1$, while $a^*_{2i} = 0, \forall i \in \{1, 2, 4, 6\}$. Also, the only non-zero values of z_{ij} are $z_{12} = z_{21} = 1$. Note that z_{13}, for instance is 0, since these queues are not parallel to each other, although they belong to the same link.

In this network, for instance, $r_{11} = r_{12} = 0.5$, while $r_{13} = r_{14} = 0$. Also, $r_{1i} = 0, \forall r = 5, 6$. Thus, in order to compute the cost of the first path, using Equation (2.32) we get:

$$c_1 = 0.5E[T_1] + 0.5E[T_2] + E[T_3] + E[T_4] \tag{2.41}$$

Thus, the cost of a path is taken to be the sum of the costs on individual queues, weighted by the fraction of flow that passes through each queue that is part of the path. The definition of indicator variables a_{ij} and a^*_{ij} allows us to directly compute path costs from the queues, without having to ever consider the links which these queues are part of.

2.3.3 Solution Procedure

When Equations (2.24) through (2.36) are solved simultaneously (Appendix B), we can obtain all the information regarding the traffic assignment of any network. This model takes the signal plan, x, the network topology and the different OD demands as input. The behavioral parameter θ is assumed to be known apriori. While the

service rates of the non-signalized queues do not change with the signal plan, and can be extracted from the simulation model R, the signal plan defines the service rates μ_i's for all signalized queues as shown in (2.42). (Note: $\mathcal{L}$ was defined earlier as the set of indices of the signalized lanes (i.e. queues), and $\mathcal{P}_L(\ell)$ was defined as the set of phase indices of lane (i.e. queue) ℓ)

$$\mu_\ell = \sum_{j \in \mathcal{P}_L(\ell)} x(j)s, \forall \ell \in \mathcal{L} \tag{2.42}$$

For each OD pair, the first 3 shortest paths are computed. These shortest paths are computed based on the length of the links in the network. We assume that all the flow between an OD pair takes goes through one of these paths. In case 3 paths aren't available between an OD pair (for example, the network shown in Figure 2-1), we distribute the flow among how many ever paths are available. While the simulation models allow a vehicles to choose from all possible paths between its origin and destination, we choose to restrict the path choice set to the first three shortest paths as an approximation in the analytical models. This approximation allows us to model the traffic assignment with the complicated cost function provided by the queuing model.

The network topology and path information determine the other exogenous parameters. The rest of the variables, λ_i, μ_i^{eff}, $\hat{\mu}_i$,$P(N_i = \ell_i)$,P_i^f, ρ_i, $E[N_i]$, $E[T_i]$, c_t, l_{st}, y_t ,γ_i and p_{ij}, are endogenous. The list of endogenous variables are given in Table 2.1. For any network, the system of equations yields $|\mathcal{Q}|^2 + 9|\mathcal{Q}| + 2|\mathcal{T}| + \sum_{s \in \mathcal{S}} |\mathcal{P}_s|$ endogenous variables and the same number of equations. All the variables listed in Table 2.1 are implemented as endogenous variables and solved for, as described in Appendix B.

When these equations are solved in terms of the endogenous variables, one can extract the expected value of the average user delay for the entire network or a subnetwork. In this section, this system of equations shall be referred to as the 'analytical model'.

When such an analytical model of the full network that is modeled in R is built,

it shall be denoted as A_R , and the corresponding analytical model of the subnetwork modeled in C shall be denoted by A_C (Appendix A). We can obtain the analytical approximation of average user travel time in a subnetwork of choice ($E[T_{sub}(x_k)]$) by applying Little's law to that subnetwork. When this average user travel time for the subnetwork is computed using A_R , it shall be denoted as $E[T_{sub}(x_k)]_{A_R}$; when it is computed using A_C, it shall be denoted as $E[T_{sub}(x_k)]_{A_C}$. (Note that to compute $E[T_{sub}(x_k)]_{A_C}$, the subnetwork of choice will be the entire network modeled in C).

Table 2.1: List of endogenous variables

Notation	**Number of variables**
λ_i	$\lvert\mathcal{Q}\rvert$
μ_i^{eff}	$\lvert\mathcal{Q}\rvert$
$\hat{\mu}_i$	$\lvert\mathcal{Q}\rvert$
$P(N_i = \ell_i)$	$\lvert\mathcal{Q}\rvert$
P_i^f	$\lvert\mathcal{Q}\rvert$
ρ_i	$\lvert\mathcal{Q}\rvert$
$E[N_i]$	$\lvert\mathcal{Q}\rvert$
$E[T_i]$	$\lvert\mathcal{Q}\rvert$
c_t	$\lvert\mathcal{T}\rvert$
l_{st}	$\sum_{s\in\mathcal{S}} \lvert\mathcal{P}_s\rvert$
y_t	$\lvert\mathcal{T}\rvert$
γ_i	$\lvert\mathcal{Q}\rvert$
p_{ij}	$\lvert\mathcal{Q}\rvert^2$

Let $\mathcal{F}$ represent the set of all queues in a subnetwork of interest. Using Little's law,

$$E[T_{sub}(x_k)] = \frac{\sum_{i \in \mathcal{F}} E[N_i]}{\sum_{j \in \mathcal{F}} \gamma_j (1 - P(N_j = j_j))} \tag{2.43}$$

Our current objective is to obtain $\hat{e}(x_k)$. We use $|E[T_{sub}(x_k)]_{A_R} - E[T_{sub}(x_k)]_{A_R}|$ as one of the inputs for this, as described below.

2.3.4 Part 1: Training the prediction model

- Before running the SO algorithm, select feasible solutions to the SO problem randomly to create the training set $\mathcal{U}$. These solutions are picked the the same way the initial point is identified, and the details are given in Section 3.2.

- Evaluate each of these feasible solutions using both the simulation models R and C (one replication using each of model R and model C). Thus we have estimates $\hat{f}^R(x)$ and $\hat{f}^C(x)$ $\forall x \in \mathcal{U}$.

- Compute $w(x) = |\hat{f}^R(x) - \hat{f}^C(x)|, \forall x \in \mathcal{U}$

- Write the equations (2.24) through (2.36) and solve the analytical models A_R and A_C, to obtain $E[T_{sub}(x)]_{A_R}$ and $E[T_{sub}(x)]_{A_C}$, $\forall x \in \mathcal{U}$.

- Compute $h(x) = |E[T_{sub}(x)]_{A_R} - E[T_{sub}(x)]_{A_R}|, \forall x \in \mathcal{U}$

- Note that $x = [x(1), x(2), ..., x(n_c)]$. Fit the following quadratic model:

$$e(x, \alpha) = \alpha_1 + \alpha_2 h(x) + \sum_{i=1}^{n_c} \alpha_{2+i} x(i) + \sum_{i=1}^{n_c} \alpha_{n_c+2+i} x(i)^2 \tag{2.44}$$

  The parameter vector $\alpha = [\alpha_1, \alpha_2, ..., \alpha_{2n_c+2}]$ is calibrated using a linear least squares procedure (Appendix D) as follows to get the optimal set of parameters α^*.

$$\underset{\alpha}{\text{minimize}} \sum_{j \in \mathcal{U}} (w(x_j) - e(x_j, \alpha))^2 + \sum_{i=1}^{2n_c+2} (\varpi_0 \alpha_i)^2 \tag{2.45}$$

The first squared term in Equation (2.45) represents the difference between the simulated and predicted inaccuracies. The second term in the same equation arises due to the addition of an artificial set of $2n_c + 2$ data points which ensure that we always have at least $2n_c + 2$ data points to fit $2n_c + 2$ parameters. This ensures that the least squares matrix is of full rank, and hence the uniqueness of the parameters. The weight ϖ_0 is set a small value of 0.001, which minimizes the impact of these artificial data points on the value of the parameters. This minimization problem is solved using the Matlab function *lsqlin*, and the algorithm parameters are detailed in Appendix D.

2.3.5 Part 2: Predicting $\hat{e}(x_k)$ at iteration k

Now that we have a prediction model, compute $\hat{e}(x_k)$ as follows:

$$e(x, \alpha^*) = \alpha_1^* + \alpha_2^* h(x) + \sum_{i=1}^{n_c} \alpha_{2+i}^* x(i) + \sum_{i=1}^{n_c} \alpha_{n_c+2+i}^* x(i)^2 \tag{2.46}$$

This value of $\hat{e}(x_k)$ can be used to make the decision of choosing a simulation model for each iteration, as described at the beginning of section 2.2.2 .

Chapter 3

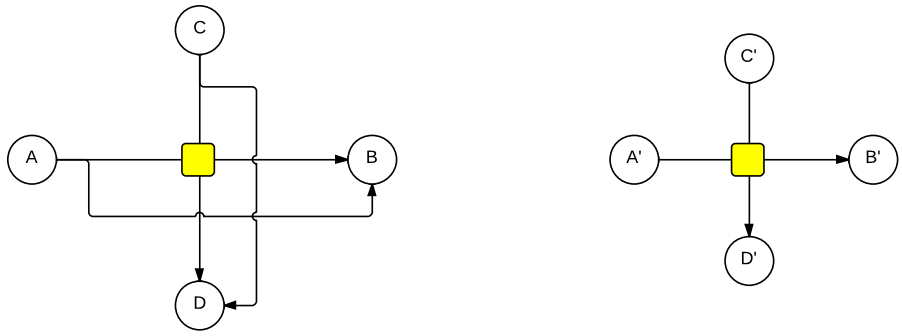
Results and Conclusions

In this section we present two case studies where the proposed algorithm is applied to a traffic signal design problem. The first case study is that of a toy network, while the second one uses data from a microsimulation model of Lausanne. We follow this up with a few concluding remarks.

3.1 Case study - toy network

We illustrate the performance of the proposed algorithm with a small toy network example. Figure 3-1 displays the networks of interest. The network on the left represents the full network R, which considers two OD apirs: $A \rightarrow B$ and $C \rightarrow D$. Each OD pair of network R has two path alternatives, one of which goes through a signalized intersection (denoted by the yellow square). The subnetwork C is displayed to the right of Figure 3-1. It contains the same OD pairs as the R network, but accounts only for the paths that travel through the intersection, i.e., it considers two OD pairs, each with one path each. A change in the signal plan of the intersection may affect the path choice probabilities. This change will be reflected when running simulator R but will not be reflected when running simulator C.

The objective function was the average user travel time for the users in the subnetwork modeled in C. The decision variable involved the green splits for two phases in one traffic signal, and hence the problem dimension (n_c) was 1. Using a total com-



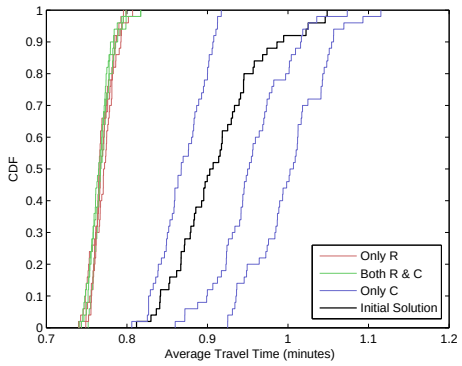
(a) regional network R (b) subnetwork C

Figure 3-1: (l) the full network modeled in R, (r) the subnetwork modeled in C (toy example)

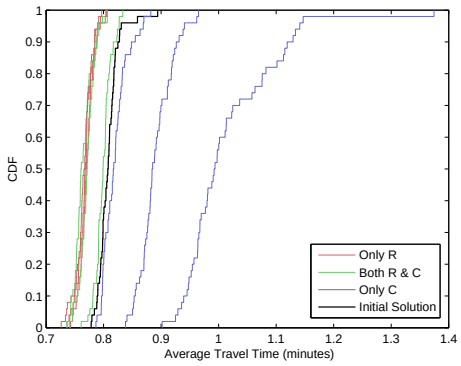
putational budget of 21 runs, we tested the different strategies with three different starting solutions. The same experiment was run three times on each of the starting solutions to identify trends.

The three strategies described in Section 2.1 were tried out with these two models. The objective function is the expected subnetwork travel time, i.e., the travel time in the links of subnetwork C. The intersection has two endogenous signal phases: one for east-west bound traffic and the second for north-south bound traffic. We evaluate the performance of the three strategies described in Section 2.1. We allow for a maximum of 21 simulation runs. That is approach 1 (resp. 2) allows for 21 runs of simulator R (resp. C), while approach 3 allows for a total of 21 runs, which consist of a combination of runs from R and from C.

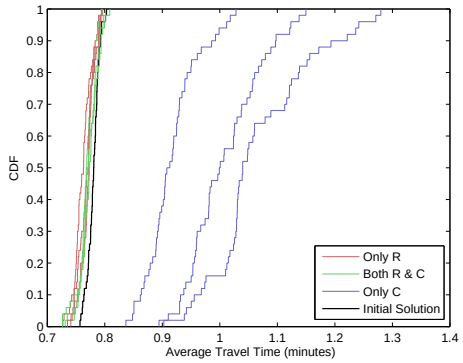
We consider three different initial signal plans. The feasibility of a signal plan is defined by Equations (2.5) and (2.7). We use the code of Stafford (2006) to random uniformly generate initial points. For each initial signal plan and each approach, we run the SO algorithm 3 times, allowing each time for a maximum of 21 simulation runs. Each time we run the SO algorithm, we obtain a new signal plan proposed by the SO algorithm. In order to evaluate the performance of this proposed signal plan, we embed it within the R simulator and run 50 simulation replications. We



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3

Figure 3-2: The results from three different initial solutions for the toy network

then plot the empirical cumulative distribution function (cdf) of these 50 simulation replications. Figure 3-2 displays the corresponding cdf curves. Each plot of the figure considers a different initial signal plan. Each plot contains 10 cdf curves:

- The black curve corresponds to the initial signal plan.
- The 3 blue curves correspond to the 3 signal plans proposed by only running the C simulator. This the least accurate yet also the least computationally-costly approach.
- The 3 red curves correspond to the 3 signal plans proposed by only running the R simulator. This is the most accurate yet also the most computationally-costly approach.
- The 3 green curves correspond to the 3 signal plans proposed by our approach.

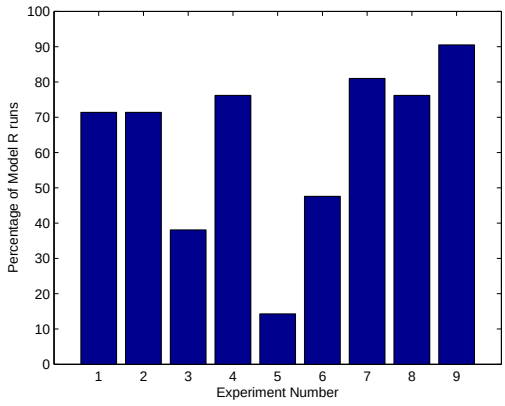


Figure 3-3: Percentage of model R runs in different experiments with the toy network

For each plot, the x-axis represents the average subnetwork travel time. For a given x value the corresponding y value of the curve represents the proportion of replications (out of the 50 replications) where the simulated average travel time was smaller than x. The more the cdf curves are shifted to the left, the higher the proportion of simulated observations with low average subnetwork travel times.

Table 3.1: Number of runs of each model, averaged across all the experiments (Toy Network)

Method	**Percentage of model R runs**	**Percentage of model C runs**
Only R	100	0
Only C	0	100
Both R and C	66.1	33.9

When running only the simulator C, all 3 plots of Figure 3-2 indicate that signal plans with poor performance are derived. In particular, in Figure 3-2(a) two out of the 3 plans proposed by C perform worse than the initial signal plan, in Figures 3-2(b) and 3-2(c) all 3 proposed plans are worse than the initial plan. In Figure 3-2(c) all 3 proposed plans perform significantly worse than the initial plan.

When running only the simulator R, all 3 plots of Figure 3-2 indicate that signal plans with good performance are obtained (with a subnetwork travel time average

of approximately 0.75 minutes). Note that for Figure 3-2(c) the improvement of the signal plans proposed by R compared to the initial plan is not large, yet the initial signal plan already had a good performance with low average travel times.

When running a combination of simulators R and C, the signal plans systematically yield performance similar to the signal plans propose by running only R. Additionally, for the proposed approach the R model was called on average 66% of the time, while the C model was called 34% of the time (Table 3.1). The number percentage of model R runs for different experiments are shown in Figure 3-3. In case of the toy network, this number seems to show a large variation from the average value of 66%. However, as we describe later, for a large scale city-network, this variation is much lesser.

These results indicate that the proposed approach identifies signal plans with good performance and does so at a lower computational cost, for the toy example. We present the results of similar experiments on a large-scale network in the next section.

3.2 Case study - Lausanne

Here we illustrate the effectiveness of our procedure on a microsimulation model of the Swiss city of Lausanne. The regional network (also referred to as the large-scale network) and the subnetwork are shown in Figure 3-4. The regional network modeled in R has a total of 653 links (modeled as a network of 922 queues) and 79 centroids (i.e nodes which that have non-zero demand or supply). This network has 2075 OD pairs with non-zero demand. On the other hand, the subnetwork modeled in C has 48 links (modeled as a network of 102 queues) and 16 centroids . Of these 16, only one of the centroids is common to model R. The others are artificially created to represent the exchange of flow between the subnetwork and the rest of the city. We compute the demand and supply for these artificial centroids by summing the contribution the paths of vehicles in an analytical model of R which pass through the subnetwork, but begin and end outside of it. This procedure is described in the appendix. Out of

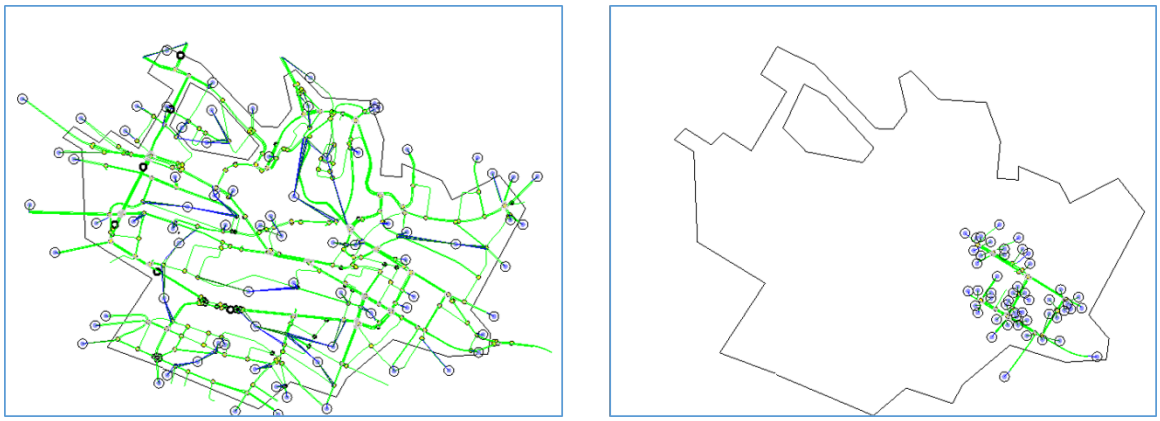


Figure 3-4: (l) the full network modeled in R, (r) the subnetwork modeled in C (Lausanne)

the 240 possible OD pair combinations that could result from these 16 centroids, 121 have non-zero demand.

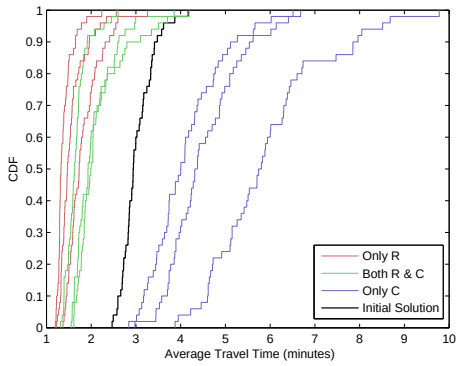
Table 3.2: Number of runs of each model, averaged across all the experiments (Lausanne Network)

Method	**Percentage of model R runs**	**Percentage of model C runs**
Only R	100	0
Only C	0	100
Both R and C	41.6	58.4

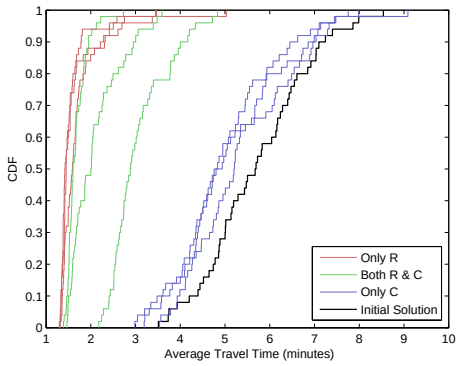
According to the route choice model that was embedded within the simulation model, each vehicle belonging to an OD pair choose one of the first three shortest paths between that OD pair. The route choice is made using a multinomial logit model, with faster routes having a higher probability of being chosen.

We implement the three strategies described at the beginning of Section 2.1 , with the objective of reducing the average user travel time for users in the subnetwork. We use the algorithm to design the signal plan for 9 intersections in the subnetwork which amount to a total of 51 endogenous phases.

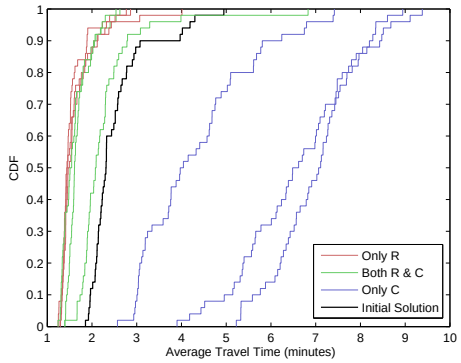
We carry out the experiment for three different initial points and with a computational budget of 200 simulation runs per experiment. For each of the initial points,



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3

Figure 3-5: The results from three different initial solutions for the Lausanne network

the same experiment is replicated thrice, to observe trends.

We encode the solution resulting from the SO algorithm into model R and run 50 simulation replications, end plot the empirical cumulative distribution function of these replications. Figure 3-5 has the curves for the three initial points. Each plot contains 10 different cdf curves, similar to the results of Section 3.1.

As seen with in the case of the toy network, using only model C results in a poor performance. In fact, for two of the three initial points, it leads to a signal plan that performs worse than the initial point (Figures 3-5(a) and 3-5(c)). On the other hand, when only model R is used, it results in signal plans perform much better than the initial points. In all of the experiments, the average subnetwork delay was reduced at least by a factor of 2.

While running experiments that use both models R and C, the proposed signal

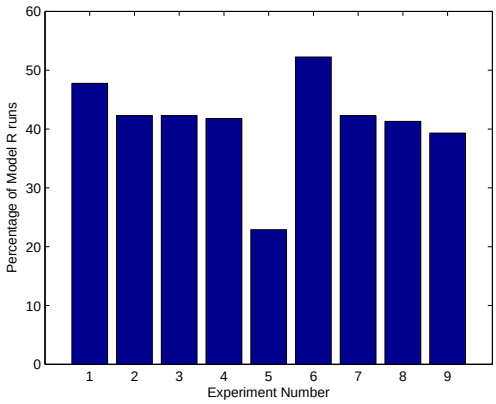


Figure 3-6: Percentage of model R runs in different experiments with the Lausanne network

plans give a performance that is close to that of the plans obtained by using only R. This performance is achieved in spite of model R being used in less than 42% of the iterations (Table 3.2). The percentage of model R runs in different experiments is shown in Figure 3-6), and there doesn't seem to be much variation from the mean value of 42% . Note that the computation time of model R is of the order of 90 seconds, while it is around 2.8 seconds for C. Therefore, the computation time used for the simulations alone is reduced by almost 78%.

However, this does not translate into a great reduction in the overall computation time as such, since the time taken to decide between the two models takes around 45 seconds in itself. This is due to the time taken to solve the traffic assignment analytically. However, by efficient implementation of the decision making framework, this time can be reduced significantly. Also, for an SO algorithm where multiple replications of the same point are required, this methodology is sure to result in major computation time savings, since the time spent on deciding between the model for a given point will be recovered by the savings resulting from multiple replications using the less expensive model C. Note that we run only one simulation replication per point in our case studies.

To illustrate the potential of our methodology, we consider the scenarios in Table 3.3 . Let $T_{choice}^{C/R}$ represent the computation time spent on choosing between the two models, and $T_{sim}^{C/R}$ represent the time spent on running simulations,per iteration, for

the method in which both models C and R are used. Similarly, let T^R_{choice} represent the computation time spent on choosing between the two models, and T^R_{sim} represent the time spent on running simulations, per iteration, for the method in which only model R is used. Note that $T^R_{choice} = 0$, as we don't spend any computational resource on making a choice in this method. Let the savings in computation time,per iteration, due to using the third strategy (of using models C and R) as compared to the computation time of the first strategy be denoted by T_{saved} %.

$$T_{saved} = 100\{\frac{T^R_{sim} - (T^{C/R}_{choice} + T^{C/R}_{sim})}{T^R_{sim}}\} \tag{3.1}$$

The computation time required to decide between the two models for a point is performed only once for each point irrespective of the number of replications the algorithm uses, and hence $T^{C/R}_{choice}$ is constant in Table 3.3 . On the other hand, the average simulation time saved per iteration ($T^R_{sim} - T^{C/R}_{sim}$) increases linearly with the number of replications. Hence, when the SO algorithm uses more than one replication per iteration, the overall computation time savings (T_{saved}) increases while the cost of taking this decision remains constant.

Table 3.3: Effect of the number of replications on the computation time savings per iteration

Number of replications per point	$T^{C/R}_{choice}$ **(seconds)**	$T^R_{sim} - T^{C/R}_{sim}$ **(seconds)**	T_{saved} **(percentage)**
1	45	50.6	6 %
3	45	151.7	39 %
5	45	252.9	46 %

3.3 Conclusions

We have addressed a large-scale traffic signal design problem using a simulation-based optimization framework. Our primary goal was to solve this optimization problem efficiently, without resorting to running a computationally intensive simulation model at every iteration.

In order to choose a simulation model at every iteration in the SO framework, we use an analytical traffic assignment model. Combining a queing network model for congestion with a multinomial logit model for route choice enabled us to model with good accuracy the traffic assignment in the network. Using the results of the traffic assignment, we are able to predict the relative inaccuracy of the small-scale model with respect to the large-scale model. This allows to trade-off the high computational costs of running accurate large-scale simulators with the lower costs of running less accurate smaller-scale simulators.

We illustrate the proposed approach with a signal control problem on a toy network. The proposed approach identifies signal plans with good performance and can do so at a lower computational cost than when systematically running the larger-scale simulator. The approach produced similar results when applied to a large-scale model that covers an entire city, resulting in a 78% reduction in the computation time spent on simulation. We also showed the potential of this method to reduce the overall computation time by 46% in the case of SO algorithms that use multiple replications of every point. Thus, we expect our work to be particularly useful for solving large-scale optimization problems using simulation based optimization.

Appendix A

Calibration of the demand for model C

We assume that a fully functional model R is given to us. In our case R is a microsimulation model in AIMSUN. Now, if our objective is to minimize the delay in a subnetwork of the city, we then carve out the subnetwork C from the larger model R. We extract model C from R in such a way that all the links of the subnetwork of interest are contained in C (Figure A-1).

We use the behavioral parameters from R in model C without any modification. A key input in the definition of model C is the traffic demand. The traffic demand in model C has two components. The first includes the traffic generated by the existing

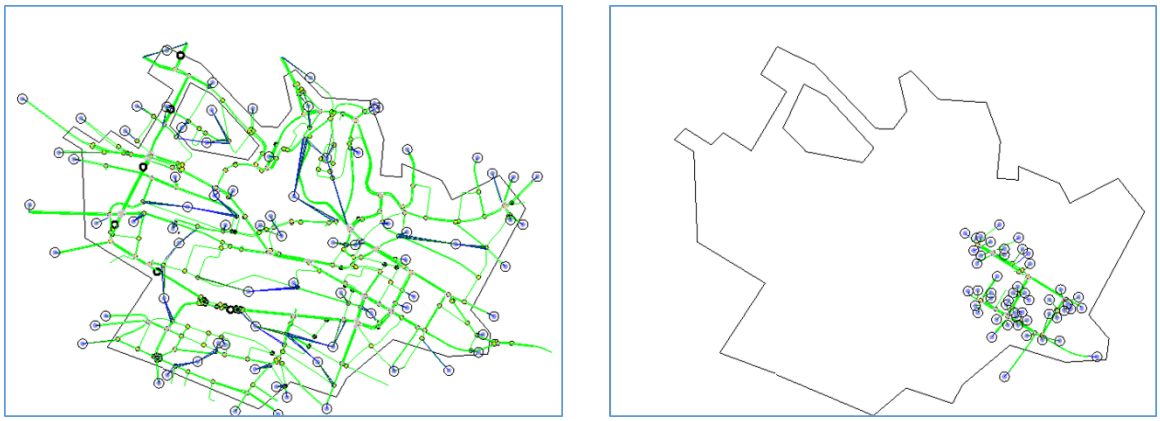


Figure A-1: (l) the full network modeled in R, (r) the subnetwork modeled in C (Lausanne)

sources and sinks located within the subnetwork. The second component includes vehicles that have their origins and destinations outside of the subnetwork, but travel through the subnetwork. For all practical purposes, this second component can be treated as being generated by a source located at the interface of models C and R and attracted to a sink that is also located at this interface.

Thus, in order to model the second component of the demand described above, we create dummy sources and sinks at the interface of R and C. The next step involved analyzing the different paths in model R and the flows on each of these paths to estimate the contribution of each of these paths to the pseudo demand matrix. While we could have chosen to analyze the path flows from model R, we chose to use an approximate method since the analysis of the paths of individual vehicles in AIMSUN is computationally intensive and time consuming. Hence, we used the path flows from analytical model A_R, corresponding to a randomly chosen signal plan, to come up with the pseudo demand matrix.

Let $\mathcal{J}$ be the set of links at the interface of model R and C. For each of these links, add two dummy nodes - an origin and a destination. let this set of dummy modes be represented by D. Now, each path in model A_R that passes through the subnetwork has to enter and exit through one of the links in $\mathcal{J}$. Since each link in $\mathcal{J}$ is associated with an origin and a destination (from $\mathcal{D}$), the flow in any path can simply be translated into a demand between two dummy nodes in $\mathcal{D}$ by identifying the entry and exit links of the path. The contributions of all the path-flows to the demand in these dummy matrices are summed up to obtain the pseudo demand matrix for C. The same demands are used in models C and A_C.

While the paths flows from A_R do change with different signal plans, we perform this procedure just once.

Appendix B

Solving the equations

We first attempted to solve the system of equations using the Matlab routine $fsolve$. We analytically computed the Jacobian and implemented it. While this method worked with networks of smaller sizes, it failed to scale to the full Lausanne network. That is, A_R couldn't be solved using this method. Hence, we adopted an approximate solution methodology.

We split the system of equations into two parts - the queuing model (Equations (2.24) to (2.29)) and the route choice model (Equations (2.32) to (2.36)). The queuing model effectively took the queue arrival rates γ_i and the transition matrix p_{ij} as inputs and could be solved to obtain the expected travel time on different queues $E[T_i]$. The choice model took $E[T_i]$ as input , and gave as output the queue arrival rates γ_i and the transition matrix p_{ij}.

The $fsolve$ routine was able to solve the queuing model without any problems. Therefore, we first started with an initial assumption for $E[T_i]$ - we set $E[T_i] = \ell_i, \forall i \in \mathcal{Q}$. Using these proxy costs, the route choice model was solved to obtain an initial estimate of the queue arrival rates γ_i and the transition matrix p_{ij}. These were then given as input to the queuing model, which was solved using $fsolve$. Now the queuing model is solved to obtain estimates of $E[T_i]$, which are used as inputs for the route choice model in the next iteration. This procedure of solving the route choice and the queuing model one after another is repeated 30 times at the end of which the values from the last $fsolve$ run are taken as the solution to the system of equations..

Listed below are the parameters we used for the $fsolve$ routing. We used the default values for the rest of the parameters. The same values were used for both the case studies (toy network and Lausanne).

Table B.1: parameters used in the $fsolve$ routine

Parameter	**Value**
'Algorithm'	'trust-region-dogleg'
'MaxFunEvals'	100000000
'MaxIter'	100
'TolFun'	1e-4
'TolX'	1e-4
'Jacobian'	'on'

Appendix C

SO parameters

The parameters of the SO algorithm were tuned a priori, and set to the values shown here. The same values were used for both the case studies (toy network and Lausanne).

Table C.1: Parameters of the SO algorithm

Parameter	**Value (toy network)**	**Value (Lausanne network)**
η_1	0.001	0.001
ν	0.9	0.9
ν_{inc}	1.2	1.2
$\bar{\tau}$	0.1	0.1
$\bar{u}$	10	10
s	1800 vehicles/hour	1800 vehicles/hour
n_{max}	21	201
δ	2	2
$\lvert\mathcal{U}\rvert$	3	25

Appendix D

Parameters used in *lsqlin* to estimate parameters using least squares

The minimization problems in Sections 2.2.3 and 2.3.5 are used to estimate parameters by minimizing a quadratic function. We used the *lsqlin* routine in Matlab for this purpose. Listed below are the parameters we used for the *lsqlin* method. We used the default values for the rest of the parameters.The same values were used for both the case studies (toy network and Lausanne).

Table D.1: parameters used in the *lsqlin* routine

Parameter	**Value**
'LargeScale'	'off'
'MaxFunEvals'	100000000
'MaxIter'	100000000
'TolFun'	1e-7

Bibliography

Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L. and Newman, P. A. (1999). Optimization with variable-fidelity models applied to wing design, *Technical Report CR-1999-209826*, NASA Langley Research Center, Hampoton, VA, USA.

Aw, A. and Rascle, M. (2000). Resurrection of "second order" models of traffic flow, *SIAM journal on applied mathematics* **60**(3): 916–938.

Ben-Akiva, M. E., Koutsopoulos, H. N., Mishalani, R. G. and Yang, Q. (1997). Simulation laboratory for evaluating dynamic traffic management systems, *Journal of Transportation Engineering* **123**(4): 283–289.

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

Bourrel, E. and Lesort, J.-B. (2003). Mixing microscopic and macroscopic representations of traffic flow: Hybrid model based on Lighthill-Whitham-Richards theory, *Transportation Research Record: Journal of the Transportation Research Board* **1852**(1): 193–200.

Broucke, M., Varaiya, P. and Kourjankski, M. (1996). SmartCap users guide, *Technical report*.

Buisson, C., Lebacque, J. and Lesort, J. (1996). STRADA, a discretized macroscopic model of vehicular traffic flow in complex networks based on the Godunov scheme, *Computational Engineering in Systems Applications (CESA)'96 IMACS Multiconference*, pp. 976–981.

Bunch, J. A., Hatcher, S. G., Larkin, J., Nelson, G. G., Proper, A. T., Roberts, D. L., Shah, V. and Wunderlich, K. E. (1999). Incorporating ITS into corridor planning: Seattle case study, *Technical report*.

Burghout, W. (2004). *Hybrid microscopic-mesoscopic traffic simulation*, PhD thesis, Department of Infrastructure, Division of Transportation and Logistics, Royal Institute of Technology.

Carter, R. G. (1986). *Multi-model algorithms for optimization*, PhD thesis, Rice University.

Chen, X., Osorio, C. and Santos, B. (2013). Travel time reliability in signal control problem: Simulation-based optimization approach, *Proceedings of the Transportation Research Board (TRB) Conference*, Washington DC, USA.

Deshpande, A., Göllü, A. and Varaiya, P. (1997). *SHIFT: A formalism and a programming language for dynamic networks of hybrid automata*, Springer.

Eldred, M. and Dunlavy, D. (2006). Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models, *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, number AIAA-2006-7117, Portsmouth, VA*, Vol. 199.

Forrester, A. I. and Keane, A. J. (2009). Recent advances in surrogate-based optimization, *Progress in Aerospace Sciences* **45**(1): 50–79.

Horowitz, R. (2004). Development of integrated meso/microscale traffic simulation software for testing fault detection and handling in AHS, *Technical Report UCB-ITS-PRR-2003-13*, Department of Electrical Engineering and Computer Sciences, Department of Mechanical Engineering, University of California, Berkeley.

Huang, D., Allen, T., Notz, W. and Miller, R. (2006). Sequential kriging optimization using multiple-fidelity evaluations, *Structural and Multidisciplinary Optimization* **32**(5): 369–382.

Magne, L., Rabut, S. and Gabard, J.-F. (2000). Towards an hybrid macro-micro traffic flow simulation model, *INFORMS*, Salt Lake City, USA.

Messner, A. and Papageorgiou, M. (1990). METANET: A macroscopic simulation program for motorway networks, *Traffic Engineering & Control* **31**(8-9): 466–470.

Montero, L., Codina, E., Barceló, J. and Barceló, P. (1998). Combining macroscopic and microscopic approaches for transportation planning and design of road networks, *Proceedings of the 19 th ARRB Transport Research Conference, Sydney*.

Newell, G. F. (1961). Nonlinear effects in the dynamics of car following, *Operations Research* **9**(2): 209–229.

Oh, J.-S., Cortés, C. E., Jayakrishnan, R. and Lee, D. (2000). Microscopic simulation with large-network path dynamics for advanced traffic management and information systems, *Proceedings of the 6th ASCE International Conference on Applications of Advanced Technologies in Transportation Engineering*.

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. and Chong, L. (2014). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation, *Transportation Science* . Forthcoming.

Osorio, C. and Nanduri, K. (2014a). Emissions mitigation: coupling microscopic emissions and urban traffic models for signal control. Submitted.

Osorio, C. and Nanduri, K. (2014b). Energy-efficient urban traffic management: a microscopic simulation-based approach, *Transportation Science* . Forthcoming.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R. and Kevin Tucker, P. (2005). Surrogate-based analysis and optimization, *Progress in aerospace sciences* **41**(1): 1–28.

Ratrout, N. T. and Rahman, S. M. (2009). A comparative analysis of currently used microscopic and macroscopic traffic simulation software., *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)* **34**: 121–133.

Robinson, T. D. (2007). *Surrogate-based optimization using multifidelity models with variable parameterization*, PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Rousseau, G., Scherr, W., Yuan, F. and Xiong, C. (2008). An implementation framework for integrating regional planning model with microscopic traffic simulation, *Logistics: the emerging frontiers of transportation and development in China: Proc. 8th Int. Conf. Chinese Logistics and Transportation Professionals*, pp. 3816–3825.

Selvam, K. (2014). Multi-model simulation-based optimization applied to urban transportation.

Sewall, J., Wilkie, D. and Lin, M. C. (2011). Interactive hybrid simulation of large-scale traffic, *Association for Computing Machinery (ACM) Transactions on Graphics (TOG)*, Vol. 30, ACM, pp. 135–152.

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

Sun, G., Li, G., Stone, M. and Li, Q. (2010). A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials, *Computational Materials Science* **49**(3): 500–511.

Treiber, M., Hennecke, A. and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations, *Physical Review E* **62**(2): 1805–1824.

Van Vliet, D. and Hall, M. (1997). Saturn 9.3-user manual, *The Institute for Transport Studies, University of Leeds, Leeds* .