Exploration and Exploitation Techniques for High-Dimensional Simulation-Based Optimization Problems in Urban Transportation

by

Timothy Tay

B.A., University of Cambridge (2015)

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Civil and Environmental Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

© Massachusetts Institute of Technology 2021. All rights reserved.

Author .................................................................
Department of Civil and Environmental Engineering
August 13, 2021

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

Accepted by ............................................................
Colette L. Heald
Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee
Abstract

Stochastic traffic and mobility simulation models are popular tools for modeling urban transportation networks. However, their use for optimizing urban transportation networks can be challenging due to their computationally intensive nature. This thesis focuses on high-dimensional simulation-based (SO) optimization problems. To find solutions with good performance efficiently, we need to balance exploration and exploitation. We propose techniques for achieving a better balance between exploration and exploitation when tackling high-dimensional SO problems in urban transportation.

The first part of the thesis considers a general-purpose exploration mechanism and introduces exploitation components to it. We propose an inverse cumulative distribution function (cdf) sampling mechanism that makes use of problem-specific prior information in the form of an analytical model to efficiently sample for points with good performance. The inverse cdf sampling mechanism can be used in conjunction with any optimization algorithm. We study whether problem-specific prior information should be used in the exploration (i.e., sampling) mechanism and/or the exploitation (i.e., optimization) algorithm when tackling a high-dimensional traffic signal control problem in Midtown Manhattan. The results show that the use of inverse cdf sampling mechanism as part of an optimization framework can help to quickly and efficiently identify solutions with good performance.

The second and third parts of the thesis focus on developing a framework to enable high-dimensional Bayesian optimization (BO) for stationary and dynamic transportation SO problems respectively. BO naturally combines exploration and exploitation. In the second part, we consider stationary problems and propose approaches to incorporate problem-specific prior information in the BO prior functions such as to jointly enhance both exploration and exploitation. This is done through the use of a stationary analytical surrogate traffic model. In the third part, we extend the BO framework to tackle dynamic problems by formulating and embedding a computation-
ally efficient dynamic analytical surrogate traffic model. For both parts, we evaluate their performance with a traffic signal control problems for a congested Midtown Manhattan (New York City) network. The proposed methods enhance the ability of BO to tackle high-dimensional urban transportation SO problems.

Thesis Supervisor: Carolina Osorio
Title: Visiting Associate Professor of Civil and Environmental Engineering
Acknowledgments

First and foremost, I would like to thank my advisor Professor Carolina Osorio for her guidance and encouragements as I made my way through my PhD journey. She is a constant source of insights, ideas and advice, regarding both research and life in general, and this has been really invaluable to me.

I would also like to thank Professor Saurabh Amin and Professor John Williams for serving on my thesis committee. Their feedback and insights have been very helpful throughout the process. I also owe my gratitude to Professor Nigel Wilson for serving on my general exam committee, and providing me with advice and guidance, especially during the early parts of my PhD journey.

Many thanks to Roberta Pizzinato for helping out with so many matters big and small, and to Kiley Clapper, Sarah Smith and Max Martelli for all their assistance in administrative matters.

The Osorio Lab at MIT has also played a huge part in supporting me during this journey. Linsen, Chao, Evan, Jing, Kevin, Tianli and Nate – thank you for brainstorming with me, as well as all the encouragements and advice.

I definitely could not have done this without the support of my friends. To my friends whom I met here in Boston – Chad, Chingkit, Rosary, Ashwin, Yuanzhen and many others – thank you all for making my time here so enjoyable. I hope we can continue to do fun things even after our paths diverge. To my other friends in other parts of the world – KJ, Megan, Shi Huan, Stacey, Daniel Yee and Daniel Ng – thank you for all the support you have given me.

Most importantly, I would like to thank my parents and siblings for everything. Their constant love, support and encouragement made me who I am today. To my fiancée Samantha, thank you so much for being there, providing me with the motivation during the lows, and celebrating with me during the highs. I feel really lucky to have you by my side as I complete this journey.

Lastly, my research is funded by a scholarship from the Agency for Science, Technology and Research. I am really grateful for this opportunity.
Contents

1 Introduction 17
  1.1 Motivation and Objective ....................................... 17
  1.2 Contributions .................................................. 19
  1.3 Thesis Structure ................................................ 21

2 A Sampling Strategy for High-Dimensional Simulation-Based Transportation Optimization Problems 23
  2.1 Introduction ..................................................... 23
  2.2 Methodology ...................................................... 29
    2.2.1 Main Idea .................................................... 29
    2.2.2 Traffic Signal Control Problem ............................ 31
    2.2.3 Formulation of the Analytical Sampling Distribution ...... 32
    2.2.4 Algorithm ................................................... 38
  2.3 Validation ........................................................ 42
  2.4 Case Study: Midtown Manhattan Signal Control .................. 49
    2.4.1 Experimental Design ......................................... 49
    2.4.2 Numerical Results ........................................... 53
    2.4.3 Computational Efficiency ................................. 61
  2.5 Conclusion ...................................................... 65

3 Bayesian Optimization Techniques for High-dimensional Transportation Problems 67
  3.1 Introduction ..................................................... 67
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1</td>
<td>High-Dimensional Bayesian Optimization</td>
<td>69</td>
</tr>
<tr>
<td>3.1.2</td>
<td>SO in Transportation</td>
<td>73</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Contributions</td>
<td>74</td>
</tr>
<tr>
<td>3.2</td>
<td>Method</td>
<td>76</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Bayesian Optimization</td>
<td>76</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Proposed Approach: Gaussian Process with Problem-Specific Information</td>
<td>82</td>
</tr>
<tr>
<td>3.3</td>
<td>Validation and Illustration</td>
<td>87</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Griewank Function</td>
<td>87</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Benchmark Methods</td>
<td>88</td>
</tr>
<tr>
<td>3.3.3</td>
<td>1-D Griewank Function</td>
<td>89</td>
</tr>
<tr>
<td>3.3.4</td>
<td>100-D Griewank Function</td>
<td>94</td>
</tr>
<tr>
<td>3.3.5</td>
<td>100-D Griewank Function with New Types of Biases</td>
<td>99</td>
</tr>
<tr>
<td>3.4</td>
<td>Case Study: Midtown Manhattan Traffic Signal Control</td>
<td>107</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Traffic Signal Optimization Problem</td>
<td>107</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Queueing Network Model</td>
<td>111</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Results</td>
<td>113</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusion</td>
<td>123</td>
</tr>
<tr>
<td>4</td>
<td>Bayesian Optimization for High-Dimensional Dynamic Urban Transportation Problems: A Traffic Model-Based Approach</td>
<td>127</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>128</td>
</tr>
<tr>
<td>4.2</td>
<td>Methodology</td>
<td>133</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Dynamic Analytical Traffic Model</td>
<td>133</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Objective Function Example</td>
<td>137</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Dynamic Bayesian Optimization</td>
<td>138</td>
</tr>
<tr>
<td>4.3</td>
<td>Validation</td>
<td>139</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Toy Networks</td>
<td>139</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Validation Results</td>
<td>141</td>
</tr>
<tr>
<td>4.4</td>
<td>Case Study: Traffic Signal Optimization</td>
<td>146</td>
</tr>
</tbody>
</table>
4.4.1 Dynamic Traffic Signal Optimization Problem .......... 148
4.4.2 Synthetic Arterial Network ........................................ 150
4.4.3 Case Study: Midtown Manhattan ............................... 162
4.5 Conclusion ................................................................. 171

5 Conclusion ................................................................. 173

A Appendices of Chapter 2 ................................................. 177
A.1 List of Notation ......................................................... 177
A.2 Derivation of Sampling Distributions and Transformation of $\rho$ to $x$
   Based on the Analytical Network Model (2.9) ............... 178
A.2.1 Derivation of the Joint cdf $G_{\rho_1,\ldots,\rho_m}(\rho_1,\ldots,\rho_m)$ ........ 178
A.2.2 Derivation of the Marginal cdf $G_{\rho_1}(\rho_1)$ (Eq. (2.17)) ........ 180
A.2.3 Derivation of Conditional cdf $G_{\rho_i|\rho_1,\ldots,\rho_{i-1}}(\rho_i|\rho_1,\ldots,\rho_{i-1})$
   (Eq. (2.19)) .............................................................. 181
A.2.4 Transformation from $\rho$ to $x$ ................................. 184
A.3 Details on Validation Plot Generation ......................... 186
A.4 Case Study Implementation Details ............................... 187
A.5 Simulation-based Optimization Algorithm ...................... 188
A.6 Comparing InvCDF-\phi and Unif-\phi .......................... 192

B Appendices of Chapter 3 ................................................. 195
B.1 List of Notation ......................................................... 195
B.2 Implementation Details for Validation with 100-D Griewank Function
   B.2.1 Optimizing the EI Acquisition Function .................... 196
   B.2.2 GP Implementation .............................................. 196
   B.2.3 Validation with 100-D Griewank Function .................. 197
B.3 Case Study Implementation Details ............................... 197

C Appendix of Chapter 4 .................................................... 201
C.1 List of Notation ......................................................... 201
C.2 Case Study Implementation Details ............................... 202
List of Figures

1-1 Use of problem-specification prior information for exploration and exploitation .................................................. 19

2-1 One-dimensional illustration of how the objective function (to be minimized) can be used to construct (a) the sampling probability density function (pdf), which can be integrated to form (b) the cumulative distribution function (cdf) so that the probability of sampling is concentrated in regions with better predicted performance. ............ 30

2-2 Toy network ............................................................................................................................................ 44

2-3 Queueing network representation of the toy network ................................................................................. 45

2-4 Two-dimensional projections of the objective function surface as estimated by the analytical model and the simulation model. ............ 46

2-5 Comparison of marginal green split distributions (a) obtained using the randomly generated sample points and (b) estimated using the simulation-based objective function values. ......................... 48

2-6 Map of Midtown Manhattan with the area of interest demarcated by a rectangle (MapQuest.com, Inc 2018) ........................................... 49

2-7 Midtown Manhattan model .......................................................................................................................... 50

2-8 Cdf’s of the average number of vehicles in the network achieved by the best signal plans generated by the uniform sampling mechanism and the inverse cdf sampling mechanism. ......................... 54
2-9 Cdf’s of the average number of vehicles in the network for the InvCDF-$\phi$ and Unif-$m$ methods using 50% and 10% exploration settings and considering 3 random initial points. ........................................... 56

2-10 Cdf’s of the average number of vehicles in the network for the InvCDF-$m$ and Unif-$m$ methods using 50% and 10% exploration settings and considering 3 random initial points. ........................................... 58

2-11 Cdf’s of the average number of vehicles in the network for the InvCDF-$m$ and InvCDF-$\phi$ methods using 50% and 10% exploration settings and considering 3 random initial points. ........................................... 60

2-12 Computational runtimes for (a, b) exploration (per exploration point), (c, d) exploitation, (e) simulation and (f) total computational runtime per iteration. ........................................... 62

3-1 Illustration of a GP posterior and acquisition function for a 1-D optimization problem. ........................................... 80

3-2 GP posteriors fitted to 2 initial observations for the 1-D Griewank function. ........................................... 90

3-3 Optimization performance of the different GP priors for the 1-D Griewank function problem. ........................................... 94

3-4 Optimization performance of the different GP priors for the 100-D Griewank function problem. ........................................... 96

3-5 Histograms and 2-D empirical cumulative distribution functions illustrating exploration in $f^A$-space and the corresponding objective function values. ........................................... 98

3-6 Optimization performance of the different GP priors for the 100-D Griewank function, when bias is introduced to $f^A$. ........................................... 101

3-7 2-D empirical cumulative distribution functions illustrating exploration in $f^A$-space when bias is introduced to $f^A$. ........................................... 104

3-8 Map of Midtown Manhattan with the simulated area demarcated by a rectangle (MapQuest.com, Inc 2018). ........................................... 110
<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-9</td>
<td>Midtown Manhattan model in Aimsun.</td>
</tr>
<tr>
<td>3-10</td>
<td>Mean performance at a given iteration achieved by the different GP priors.</td>
</tr>
<tr>
<td>3-11</td>
<td>2-D empirical cumulative distribution functions illustrating exploration in $f^A$-space.</td>
</tr>
<tr>
<td>3-12</td>
<td>Multidimensional scaling 2-D projections of all observations in the feasible region.</td>
</tr>
<tr>
<td>3-13</td>
<td>Empirical cdf illustrating distribution of (a) computational runtimes for optimizing the acquisition function at each iteration, and (b) total computational runtime per iteration.</td>
</tr>
<tr>
<td>4-1</td>
<td>Illustration of toy networks used for validation of the dynamic analytical traffic model.</td>
</tr>
<tr>
<td>4-2</td>
<td>Empirical cumulative distribution function illustrating the distribution of link lengths in the Midtown Manhattan network.</td>
</tr>
<tr>
<td>4-3</td>
<td>Example of vehicle density dynamics for the 2-link network (100m links, 720veh/h demand).</td>
</tr>
<tr>
<td>4-4</td>
<td>Example of vehicle density dynamics for the 4-link network (50m links, 720veh/h demand).</td>
</tr>
<tr>
<td>4-5</td>
<td>Synthetic arterial network model.</td>
</tr>
<tr>
<td>4-6</td>
<td>Mean performance at a given iteration for different levels of demand with 2 time intervals.</td>
</tr>
<tr>
<td>4-7</td>
<td>Mean performance at a given iteration for different levels of demand with 4 time intervals.</td>
</tr>
<tr>
<td>4-8</td>
<td>Average travel time by OD pair for different levels of demand with 2 time intervals.</td>
</tr>
<tr>
<td>4-9</td>
<td>Average travel time by OD pair for different levels of demand with 4 time intervals.</td>
</tr>
<tr>
<td>4-10</td>
<td>Map of Midtown Manhattan with the simulated area demarcated by a rectangle (MapQuest.com, Inc 2018).</td>
</tr>
</tbody>
</table>
4-11 Midtown Manhattan model in Aimsun. .......................... 164
4-12 Mean performance at a given iteration achieved by the different BO methods. ......................................................... 166
4-13 (a) Total mean queue length, (b) total actual queue length, and (c) total virtual queue length at a given time interval. ............... 170

A-1 Cdf’s of the average number of vehicles in the network for the Invcdf-ϕ and Unif-ϕ methods using 50% and 10% exploration settings and considering 3 random initial points. .............................. 193

C-1 Mean performance at a given iteration for different demand profiles. . 204
List of Tables

2.4 Origin-destination travel demand (i.e., external arrival rate) in veh/h 45
2.5 SO Methods ................................................................. 52
2.6 Proportion of Computational Budget for Exploration .................. 52
2.7 $T$-statistics for paired $t$-test (Invcdf-$\phi$ vs. Unif-$m$) .......... 55
2.8 $T$-statistics for paired $t$-test (Invcdf-$m$ vs. Unif-$m$) .......... 57
2.9 $T$-statistics for paired $t$-test (Invcdf-$m$ vs. Invcdf-$\phi$) ....... 59
3.1 GP Priors ................................................................. 89
3.2 One-sided paired $t$-test results ..................................... 115
3.3 One-sided two-sample $t$-test results ................................ 116
4.2 Dynamic analytical traffic model parameter values .................. 141
4.3 Simulation-based Level of Congestion Statistics (2-link Network) . 143
4.4 Simulation-based Level of Congestion Statistics (4-link Network) . 143
4.5 Mean Absolute Errors (2-link Network) .............................. 147
4.6 Mean Absolute Errors (4-link Network) .............................. 147
4.8 Demand ................................................................ 152
4.9 One-sided paired $t$-test results (SAN optimization performance) ...... 157
4.10 One-sided paired $t$-test results (average travel time by OD pair in SAN) 161
4.11 One-sided paired $t$-test results (MTM optimization performance) . 167
4.12 One-sided two-sample $t$-test results (MTM optimization performance) 167
B.2 Fixed Hyperparameter Values Tested .................................. 198
B.3 Pre-selected hyperparameter values .................................. 199
C.2 Pre-selected hyperparameter values (SAN) ........................................ 203
C.3 Pre-selected hyperparameter values (MTM) ..................................... 203
C.4 Demand Profiles ........................................................................... 204
Chapter 1

Introduction

1.1 Motivation and Objective

Urban transportation systems are essential for the movement of people and goods around urban areas. As cities grow in size, the demand for travel into, out of, and around the cities increase. This ever increasing travel demand has led to growing congestion in many cities around the world. Congestion results in delays, which represent a loss of efficiency in the urban transportation systems. According to a report by the Texas A&M Transportation Institute (Schrank, Eisele, and Lomax 2019), the average commuter in the US wasted 54 hours in congestion in 2017, costing them $1,080 in wasted time and fuel. Congestion also contributes other societal costs, such as the deterioration of air quality and noise pollution.

With the current push towards electric vehicles and other alternative modes of transportation (e.g. bike sharing), some of the emissions problem can be alleviated. To address the issues of reducing congestion and improving efficiency in the urban transportation systems, optimizing the traffic management strategies is a cost-effective solution. In the future, connected and autonomous vehicles also promise to bring about more efficient usage of the current road networks as well.

The development of these solutions to congestion in the urban transportation systems often rely on the use of high-resolution simulation models. For instance, many transportation agencies make use of high-resolution traffic simulation models, such
as microscopic simulators, to evaluate the effect of changes to their transportation networks. At the same time, simulations are often used in the development and training of autonomous vehicles to help reduce overall training times.

Recently, there has been interest in using high-resolution simulation models to address optimization problems. This is known as simulation-based optimization (SO). As an example, in the context of urban transportation, these optimization problems can take the form of identifying the best traffic management strategy (e.g. best traffic signal plan) with regards to network metrics (e.g. travel time, delays, network throughput, etc.). The high-resolution traffic simulation models, with their intricate description of the dynamics in the transportation system, can provide accurate estimates of the network metrics. However, these simulation models are computationally expensive to evaluate. This means that the number of simulation runs often has to be limited when addressing SO problems. Furthermore, city-wide (i.e. large-scale) optimization problems may involve many decision variables (i.e. high-dimensional). The high-dimensional nature of the optimization problems combined with the limited number of simulation runs makes large-scale urban transportation SO problems very challenging.

Existing methods for tackling SO problems are mostly general-purpose algorithms and tend to focus more on exploitation techniques (see Sections 2.1 and 3.1 for a review of some of these methods). They are inefficient when it comes to solving high-dimensional SO problems, where the simulations are computationally expensive to evaluate. On the other hand, SO algorithms which incorporate problem-specific information have also been proposed. While they have shown promising results in tackling large-scale urban transportation SO problems, they also focus on exploitation.

In this thesis, we further develop methods for solving high-dimensional transportation SO problems. The proposed methods aim to better balance exploration and exploitation, while incorporating problem-specific information. In the context of optimization, exploration refers to searching in regions with few evaluated points, while exploitation refers to searching in regions with good estimated performance.
Figure 1-1: Use of problem-specification prior information for exploration and exploitation

(Sun, Hong, and Hu 2014). Too much focus on exploration leads to a slow convergence to a good solution. Conversely, too much focus on exploitation leads to the optimization getting stuck in local optima. By balancing exploration and exploitation, the proposed methods are able to achieve enhanced optimization performance within finite computational budgets.

1.2 Contributions

This thesis focuses on tackling high-dimensional large-scale transportation SO problems. In the thesis, we proposed different methods to incorporate problem-specific prior information in SO algorithms to achieve more efficient optimization, while maintaining a balance between exploration and exploitation. This allows us to obtain solutions with good performance within a limited number of algorithmic iterations. Figure 1-1 shows how problem-specific prior information is used in each of the subsequent chapters.

In Chapter 2, we propose a sampling mechanism which can be used in conjunction with any optimization algorithm. This sampling mechanism is built upon an analytical model of the underlying traffic network (i.e. the problem-specific prior information) such that it assigns greater sampling probability to points with better predicted performance. As such, this allows the problem-specific prior information to be used for exploration, so that points with good predicted performance can be sampled efficiently. The proposed sampling mechanism provides an alternative ap-
proach to using problem-specific information for optimization purposes. Additionally, we provide insight into whether the problem-specific information should be used for exploration (i.e. sampling) and/or exploitation (i.e. optimization) when tackling an SO problem.

In addition, we also propose techniques for combining problem-specific information with Bayesian optimization (Chapter 3). The proposed approaches allow for the problem-specific information in the form of an analytical surrogate model to be used for both exploration and exploitation. More importantly, the proposed approaches enable Bayesian optimization to tackle high-dimensional transportation problems. The use of our proposed approaches allow for better solutions to be found within the limited simulation budget. In contrast, traditional Bayesian optimization is known to be relatively ineffective when it comes to tackling high-dimensional optimization problems. Furthermore, the effect of bias in the analytical surrogate model on optimization performance is also investigated in Chapter 3. This method is applied to a stationary SO problem in Chapter 3 and a dynamic SO problem (i.e. with time-dependent variables) in Chapter 4.

In order to carry over the ability of Bayesian optimization to balance exploration and exploitation, and apply it to dynamic SO problems, we propose a dynamic analytical traffic model in Chapter 4. This allows the proposed approaches from Chapter 3 to be extended to tackle dynamic problems through the use of a dynamic analytical surrogate model. This dynamic analytical traffic model is able to model the traffic dynamics of large-scale networks. At the same time, the model is computationally tractable, making it suitable for optimization purposes. With the dynamic analytical traffic model, we provide a Bayesian optimization framework for tackling dynamic SO problems.

The proposed methods can be used to address a variety of transportation SO problems. In this thesis, the proposed methods were tested in case studies where we considered a traffic signal control problem of Midtown Manhattan. The Midtown Manhattan network is considered a large-scale traffic network. It consists of a total of 698 roads, 2756 lanes and 444 intersection. The traffic signal control problem involves
controlling 97 signalized intersections, with a total of 259 signal phases. Hence, this makes it a large-scale traffic signal control problem.

1.3 Thesis Structure

The thesis is structured as follows:

Chapter 2 proposes an inverse cumulative distribution function sampling mechanism that makes use of problem-specific structural information in the form of an analytical model to efficiently sample for points with good performance. It shows how the structural information can be exploited to quickly find good solutions, while still allowing for exploration in the feasible region. The proposed method was presented as:


Chapter 3 proposes techniques that enable Bayesian optimization to tackle high-dimensional problems. The proposed approaches incorporate problem-specific prior information (in the form of an analytical transportation model) in the Gaussian process prior. This helps to promote exploration of the feasible region, while exploiting the problem-specific information, with the aim of finding better solutions. The proposed method was presented at conferences:


Chapter 4 extends the proposed BO framework from Chapter 3 to tackle the class of dynamic SO problems (i.e. problems with time-dependent variables). For this purpose, we formulate a computationally efficient dynamic traffic network that is suitable for optimization purposes, while being able to capture the temporal dynamics of large-scale traffic networks.

Chapter 5 summarizes the findings of this thesis. Possible future research directions are also discussed.
Chapter 2

A Sampling Strategy for
High-Dimensional Simulation-Based
Transportation Optimization
Problems

This chapter proposes an inverse cumulative distribution function (cdf) sampling mechanism to make use of problem-specific structural information in the form of an analytical model to efficiently generate exploration points with good performance. The resulting sampling distribution assigns greater probability to sampling points with better predicted performance. We also provide insight into whether problemspecific information should be used in the exploration component and/or the exploitation component of the optimization algorithm when tackling simulation-based optimization problems.

2.1 Introduction

Simulation-based urban traffic and mobility models have been extensively used by stakeholders to evaluate the performance of traffic management strategies (e.g. signal control, congestion pricing), as well as to evaluate new algorithms (e.g. dispatch
algorithms in ridesharing) before releasing it to production (Greenhall 2016). These simulators can embed detailed traveler behavior models of departure time choice, mode choice, route choice and response to real-time traffic information. Moreover, stochastic models can account for uncertainty in demand and in supply. For instance, the behavioral models above may be specified as disaggregate probabilistic models, such as discrete choice models. Furthermore, the simulators are able to model the interactions of a large number of heterogeneous travelers to a resolution that analytical models are unable to. A review of traffic simulation models is given in Barceló et al. (2010).

As the resolution, quantity and availability of urban mobility data increases, so does the interest by both public and private stakeholders in developing detailed, high-resolution mobility models. Moreover, with the increase in connectivity and real-time responsiveness of our mobility systems, comes an increase in the spatial dependency of network or link performance. This calls for an increase in the spatial coverage of the network to be studied. Stakeholders are shifting from performing a local (e.g., an arterial, a couple of intersections, a neighborhood) analysis to studying the impact across a full city or metropolitan area. This simultaneous increase in both the model resolution and its spatial coverage brings numerous methodological challenges.

In this work, we focus on the use of stochastic simulation-based models for continuous simulation-based optimization (SO) problems. We consider SO problems of the following form:

$$\min_{x \in \chi} f(x, z; p) \equiv E[F(x, z; p)],$$

where $f$ is the objective function, $F$ represents the stochastic output of a simulation run, $x$ is the high-dimensional vector of continuous decision variables, $\chi$ is the feasible region, $z$ denotes the vector of endogenous simulation variables and $p$ represents the vector of deterministic exogenous parameters. The constraints that define $\chi$ are assumed to be analytical, differentiable and convex. Some examples of continuous transportation SO problems include traffic signal optimization (Osorio and Bierlaire...
The main challenges in urban mobility SO are twofold. The first challenge is the computational cost of evaluating the simulator. Computational cost increases with both the model resolution and with the spatial network coverage. Hence, the new generation of mobility models is increasingly costly to evaluate. The second challenge involves dealing with high-dimensional optimization problems. Examples of high-dimensional transportation SO problems include OD calibration (Tympakianaki, Koutsopoulos, and Jenelius 2018, 2015, Zhang, Osorio, and Flötteröd 2017, Lu et al. 2015), traffic signal optimization (Chong and Osorio 2018, Osorio and Chong 2015) and air traffic control (Hutchison and Hill 2001).

Due to the high computational cost of evaluating the simulator, we consider a tight computational budget when tackling the SO problem. This means we work in a context with fewer objective function evaluations than the problem dimension. As a result, it is important to balance exploration and exploitation in order to find a good solution within the limited computational budget. In the context of optimization, exploration refers to the search in regions with few evaluated points, so as to improve our knowledge of the feasible region. Exploitation refers to the search in regions with good estimated performance (Sun, Hong, and Hu 2014).

The common approaches in the transportation SO literature have focused more on exploitation techniques. These approaches include general-purpose algorithms such as genetic algorithms (Jin, Ma, and Kosonen 2017, Sebastiani, Lüders, and Fonseca 2016, Paz et al. 2015, Stevanovic et al. 2008, Teklu, Sumalee, and Watling 2007, Yun et al. 2006) and simultaneous perturbation stochastic approximation (SPSA) (Ros-Roca, Montero, and Barceló 2021, Tympakianaki, Koutsopoulos, and Jenelius 2018, 2015, Lu et al. 2015, Balakrishna and Koutsopoulos 2008). The advantage of these algorithms is their broad generality, i.e., they are not limited to a specific type of
transportation problems. However, this generality comes with a lack of computational efficiency. The underlying algorithms are designed based on asymptotic properties and exploit little to no problem structure information. In other words, they treat the simulator as a black-box. Hence, they are not designed to be used within tight computational budgets, yet that is how they are used.

To design computationally efficient SO algorithms, one recent line of work has been to complement general-purpose SO algorithms with problem-specific structural information, i.e., treat the simulator as a gray-box. The problem-specific structural information provides prior knowledge of the problem, even before any simulation is done, thus helping to quickly identify some good solutions. This can come in the form of analytical models, which provide an approximation of the objective function (e.g. queueing network model representing a traffic network (Osorio and Bierlaire 2009)).

A technique that has been proven to successfully tackle various classes of transportation SO problems is that of metamodeling (e.g., Gu, Waller, and Saberi 2019, Chen et al. 2014, Osorio and Bierlaire 2013). In the metamodel SO approaches, the objective function \( f \) of Eq. (2.1) is replaced by a parametric analytical approximation \( m_k \) at a given iteration \( k \) of the SO algorithm. At each SO iteration \( k \), the problem (2.1) is tackled by minimizing \( m_k \). Past metamodel SO works have used problem-specific approximations of the objective function to derive \( m_k \) (Chong and Osorio 2018, Osorio and Chong 2015, Osorio and Nanduri 2015a,b, Osorio and Bierlaire 2013). In other words, these past metamodel SO approaches have used problem-specific structural information to improve the exploitation component of the SO method.

However, little work has been done in the transportation SO literature on the design of suitable exploration techniques. In most cases, the default exploration mechanism for existing SO methods (both general-purpose and problem-specific) is based on uniform sampling. This is often in the form of standard uniform random sampling (e.g., Sebastiani, Lüders, and Fonseca 2016, Osorio and Chong 2015, Paz et al. 2015) or Latin hypercube sampling (e.g., Jin, Ma, and Kosonen 2017, Chen et al. 2014).
This chapter proposes a method to improve the exploration component of an SO algorithm. Similar to past metamodel work, a problem-specific approximation (i.e., an analytical model) of the objective function is derived. However, instead of using this approximation as an analytical proxy for the objective function, this chapter proposes its use to define a sampling distribution. Points with good performance as predicted by the analytical model are assigned higher sampling probability.

This chapter also investigates the added value of using problem-specific structural information either to improve the exploitation component of an SO algorithm or the exploration component. The results of this paper show the potential of exploration mechanisms that incorporate problem-specific structural information to enhance the efficiency and the scalability of general-purpose SO algorithms.

The analytical model could be used to construct a joint sampling distribution. However, it can be difficult to draw samples from the joint distribution. Monte Carlo methods, including importance sampling and Markov chain Monte Carlo methods, are often used to address this problem. In importance sampling, the goal is to estimate an integral or an expectation value with respect to a target distribution, while reducing the variance on the estimate. Since the goal of importance sampling is different from the goal of this work, and importance sampling does not sample according to the target distribution, we will not discuss it in detail. For a review on importance sampling, we refer the reader to Tokdar and Kass (2010).

Markov chain Monte Carlo (MCMC) is a class of methods for generating samples from a sampling distribution while exploring the state space using a Markov chain mechanism. One of the most popular MCMC methods is the Metropolis-Hastings algorithm (Metropolis et al. 1953, Hastings 1970). The Metropolis-Hastings algorithm involves sampling from a proposal distribution (which is easy to sample from) in order to mimic samples drawn from the target distribution asymptotically. If the expressions for the full conditionals of the target distribution are known, the Gibbs sampler can then be used to draw samples from the target distribution (Geman and Geman 1984), eliminating the need to specify a proposal distribution.

In this study, we propose an inverse cumulative distribution function (cdf) sam-
pling mechanism to make use of problem-specific structural information in the form of an analytical model (Osorio and Bierlaire 2009, Osorio and Chong 2015) to efficiently sample for exploration points with good performance in a high-dimensional transportation optimization problem. The inverse cdf sampling mechanism provides a positive correlation between the probability of sampling a particular point and the predicted performance of the point. In contrast to uniform sampling, which is traditionally used for picking initial points for optimization, the inverse cdf sampling mechanism attempts to exploit prior information while still allowing for nonzero probability of sampling any point. Thus, the inverse cdf sampling mechanism should be able to pick out better points compared to a uniform sampling method, especially when working with the constraint of a limited computational budget. In addition, the analytical model gives a global approximation of the objective function without the need for any simulation evaluations, so this design should be able to achieve good short-term performance in a high-dimensional problem when used in an optimization framework.

The proposed methodology contributes in the following ways:

- **Scalability:** The inverse cdf sampling mechanism makes use of an analytical model that consists of a simple system of equations. The number of equations in the model is linear in the number of links in the network, which allows the inverse cdf sampling mechanism to be scaled to high dimensions.

- **Efficiency:** By exploiting prior information in the form of an analytical model of the traffic network, the inverse cdf sampling mechanism samples for exploration points with good performance with higher probability than the commonly-used uniform sampling mechanisms so that solutions with good performance can be found with just a small number of simulation evaluations.

For the rest of the chapter, we use the terms exploration mechanism and sampling mechanism (used to generate exploration points) interchangeably. In the following section, we present the methods used to implement the inverse cdf sampling mechanism. A validation of the inverse cdf sampling mechanism is done in Section 2.3. We
then test the inverse cdf sampling in a case study using a model of Mid-town Manhattan for a fixed time traffic signal control problem. The results of the case study are presented, along with a discussion, in Section 2.4. Our conclusions are provided in Section 2.5.

2.2 Methodology

In this section, we first explain the main idea behind the inverse cdf sampling mechanism in Section 2.2.1. The optimization problem of interest is formulated in Section 2.2.2. Section 2.2.3 derives the expressions for the sampling distributions. The algorithm for the inverse cdf sampling mechanism is summarized in Section 2.2.4.

2.2.1 Main Idea

To illustrate the main idea of the proposed approach, we consider a one-dimensional minimization problem for which we have an analytical approximate expression, denoted $f^A$, of the SO objective function $f$ in Eq. (2.1). Note that $f^A$ does not rely on the use of any simulation observations. The goal is to use $f^A$ to define a sampling distribution that assigns high sampling probability to regions of the feasible region with good predicted performance (i.e., low $f^A$ values) and low sampling probability to regions with bad predicted performance (i.e., high $f^A$ values). The probability density function (pdf) of such a sampling distribution is given by:

$$g(x) = \frac{1}{\kappa_0} (\kappa_1 - f^A(x)),$$

(2.2)

where $\kappa_1$ is a scalar upper bound of $f^A$, which ensures non-negativity of $g$, and $\kappa_0$ is a scalar normalization constant, which ensures that $g$ integrates to 1.

This is graphically illustrated in Figure 2-1a, where $f^A$ (resp. $g$) is represented by the dashed (resp. solid) line. In this figure, the magnitude of $g$ is inversely proportional to that of $f^A$. For instance, the interval $[a_1, a_2]$ has good predicted performance (i.e., low $f^A$ values) and thus high sampling probabilities (high $g$ values).
Figure 2-1: One-dimensional illustration of how the objective function (to be minimized) can be used to construct (a) the sampling probability density function (pdf), which can be integrated to form (b) the cumulative distribution function (cdf) so that the probability of sampling is concentrated in regions with better predicted performance.

The inverse holds for interval \([a_3, a_4]\), which has bad predicted performance and thus low sampling probabilities.

To sample according to the pdf \(g\), we use the inverse transform method, also known as inversion sampling or Smirnov transform. See for instance Casella and Berger (2002, Section 5.6.1, pages 247-249). The method assumes that the cumulative distribution function (cdf) of \(g\), denoted \(G\), is invertible, i.e., \(G^{-1}\) exists. It samples according to \(G^{-1}(U)\), where \(U\) is a standard uniform random variable (i.e., a uniform distribution with support \([0, 1]\)).

Figure 2-1b depicts the cdf \(G\) of the pdf \(g\) of Figure 2-1a. Intuitively, the cdf has steeper gradients in the regions where the magnitude of \(g\) is large. The inverse transform method samples independently from a standard uniform distribution, places these realizations along the \(y\)-axis of Figure 2-1b, and projects them according to \(G^{-1}\) on the \(x\)-axis. Recall the example above where interval \([a_1, a_2]\) has better predicted performance than interval \([a_3, a_4]\). In Figure 2-1b sampling from interval \([a_1, a_2]\) (resp. \([a_3, a_4]\)) occurs with probability \(u_2 - u_1\) (resp. \(u_4 - u_3\)). Since \((u_2 - u_1) > (u_4 - u_3)\), the probability of sampling points in \([a_1, a_2]\) is greater than that of sampling points in
[a_3, a_4]. In other words, sampling from the interval with better performance ([a_1, a_2]) occurs with a higher probability than sampling from the interval with worse performance ([a_3, a_4]).

The main challenge in our proposed approach is to formulate an approximation function \( f^A \) that balances providing: (i) a good approximation of the unknown SO objective function, and (ii) a computationally efficient and scalable (i.e., suitable for high-dimensions) way of sampling from the feasible region. In this chapter, we consider a specific transportation optimization problem and propose such a formulation.

### 2.2.2 Traffic Signal Control Problem

The proposed sampling mechanism is applicable to a broad range of continuous SO problems. To illustrate its performance, we consider a high-dimensional traffic signal control problem. For a review of traffic signal control terminology, see Osorio (2010, Appendix A, pages 119-121). We consider a fixed-time control strategy. A fixed-time signal plan is cyclic (i.e., periodic). In this chapter, the decision variables are the green splits (i.e., normalized green times) of each signal phase of each intersection in the network. All other control variables (e.g., offsets, stage structure, cycle times) are predetermined and assumed fixed. The notation used in the formulation of the traffic signal control problem is given below, and also summarized in Appendix A.1

- \( f \): SO objective function (expected number of vehicles in the road network);
- \( F \): random variable denoting the number of vehicles in the road network;
- \( x_j \): green split of signal phase \( j \) (decision variable);
- \( x \): vector of all green splits (decision vector);
- \( z \): vector of endogenous simulation variables.

**Exogenous problem parameters:**

- \( c_\ell \): cycle time of intersection \( \ell \);
- \( d_\ell \): fixed cycle time of intersection \( \ell \);
- \( x^{LB} \): vector of minimal green splits;
- \( \theta \): vector of exogenous simulation parameters;
- \( n \): total number of signal phases to be optimized (i.e., dimension of the decision vector \( x \));
- \( I \): set of signal controlled intersection indices;
- \( \mathcal{P}_1(\ell) \): set of signal phase indices of intersection \( \ell \).
The problem is formulated as follows:

\[
\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) = \mathbb{E}[F(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})] \quad (2.3)
\]

subject to \[
\sum_{j \in \mathcal{P}_1(\ell)} x_j = \frac{c_\ell - d_\ell}{c_\ell}, \; \forall \ell \in \mathcal{I} \quad (2.4)
\]

\[
\mathbf{x} \geq \mathbf{x}^{LB} \quad (2.5)
\]

The decision vector \(\mathbf{x} = (x_1, \ldots, x_n)\) consists of the green splits for each signal phase. The objective function \(f\) (Eq. (2.3)) is taken to be the expected number of vehicles in the road network. The expectation is defined by the stochastic traffic simulator, with \(F\) representing the corresponding random variable. This objective function depends on a vector of exogenous simulation parameters \(\boldsymbol{\theta}\), which represents, for instance, network topology or fixed lane attributes (e.g., lane length, maximum lane speed). It also depends on a vector of endogenous simulation variables \(\mathbf{z}\), which represent, for instance, link-level and network-level performance metrics (e.g., expected queue-lengths, speeds, flows, travel times). The right hand side of Eq. (2.4) represents the proportion of the cycle time that can be optimized (i.e., that is not fixed). In practice, portions of the cycle time are constrained to be fixed typically for safety considerations and to comply with local transportation regulations. These fixed portions can include all-red periods, where the signal is red for all traffic streams at the given intersection, as well as portions where some traffic streams have short green times, such as to enable a smooth transition between two signal phases. Eq. (2.4) equates, for a given intersection, the proportion of non-fixed cycle time to the total endogenous green time (i.e., sum of green splits) allocated to that intersection (left hand side). The green splits have lower bounds (Eq. (2.5)), which are typically determined based on safety considerations following guidelines from local transportation authorities.

### 2.2.3 Formulation of the Analytical Sampling Distribution

Recall from Eq. (2.2) that the sampling distribution \((g)\) is defined based on an analytical approximation \((f^A)\) of the simulation-based objective function \(f\) of Eq. (2.3).
In this section, we derive an expression for $f^A$. We represent a given road network by a finite (space) capacity queueing network. More specifically, each lane $i$ of the network is represented by a finite (space) capacity queue $M/M/1/k_i$, where $k_i$ denotes the finite space capacity of the underlying lane. By using finite, rather than infinite, space capacity queues, we account for the limited number of cars each lane can accommodate based on the physical length of the lane. This allows us to model vehicular spillback (i.e., when the length of the queue of vehicles exceeds the lane length, and spills back to upstream lanes) through the queueing theoretic concept of blocking. To introduce this model, we define the following notation. The index $i$ refers to a given queue. The notation used is also summarized in Appendix A.1.

**Endogenous variables of the analytical model:**

- $\rho_i$ traffic intensity;
- $\lambda_i$ arrival (i.e., demand) rate;
- $N_i$ number of vehicles in queue $i$;
- $P(N_i = k_i)$ probability of queue $i$ being full, also known as the blocking or spillback probability;
- $\mu_i$ service rate;

**Exogenous parameters of the analytical model:**

- $m$ total number of queues in the network;
- $\gamma_i$ external arrival (i.e., demand) rate;
- $p_{ij}$ probability of turning from queue $i$ to queue $j$;
- $k_i$ space capacity in terms of number of vehicles;
- $s$ saturation flow rate;
- $\mathcal{M}$ set of all queues;
- $\mathcal{U}_i$ set of upstream queues of queue $i$;
- $\mathcal{D}_i$ set of downstream queues of queue $i$;
- $\mathcal{L}$ set of signalized queues;
- $\mathcal{P}_2(i)$ set of signal phase indices of signalized queue $i$;
- $\mathcal{P}_3(i)$ index of the intersection that queue $i$ leads to;
- $\mathcal{P}_4(i)$ set of signal indices at the intersection that queue $i$ leads to, but not including the indices of signals which are green in favor of vehicles in queue $i$.

Hereafter the terms queue and lane are used interchangeably. The expected num-
ber of vehicles in the network is defined as:

\[ f^A(x) = E \left[ \sum_{i=1}^{m} N_i \right] \]  

(2.6)

\[ = \sum_{i=1}^{m} E[N_i] \]  

(2.7)

\[ = \sum_{i=1}^{m} \left( \frac{\rho_i}{1 - \rho_i} - \frac{(k_i + 1)\rho_i^{k_i+1}}{1 - \rho_i^{k_i+1}} \right). \]  

(2.8)

In Eq. (2.8), \( N_i \) depends on \( \rho_i \) (as shown in Eq. (2.8)), and \( \rho_i \) depends on \( x \). The transition from Eq. (2.7) to (2.8) is based on finite capacity queueing theory. See for instance, Bocharov et al. (2004, Chapter 3, pages 96-98). A derivation of this expression is given in Osorio and Chong (2015, Appendix A).

Since the expression of Eq. (2.8) does not depend explicitly on \( x \), but instead depends on \( \rho \) which itself is a function of \( x \), we simplify the notation by using \( \rho_i \) to refer to \( \rho_i(x) \). We now define the mapping between \( \rho \) and \( x \). To do this, we use the analytical network model of Osorio and Chong (2015, Eq. (6)), which is defined as the following system of 3m nonlinear equations:

\[ \lambda_i = \gamma_i (1 - P(N_i = k_i)) + \sum_{j \in \mathcal{U}_i} p_{ji} \lambda_j \]  

(2.9a)

\[ \rho_i = \frac{\lambda_i}{\mu_i} + \left( \sum_{j \in \mathcal{D}_i} p_{ij} P(N_j = k_j) \right) \left( \sum_{j \in \mathcal{D}_i} \rho_j \right) \]  

(2.9b)

\[ P(N_i = k_i) = \frac{1 - \rho_i}{1 - \rho_i^{k_i+1} \rho_i^{k_i}}. \]  

(2.9c)

Eq. (2.9a) is a flow conservation equation relating the demand rate of queue \( i \) (left hand side of the equality) to the sum of the demand rate of vehicle trips that start in queue \( i \) (first term of the right hand side) and of the demand rate of vehicle trips that arise from upstream queues (second term of the right hand side). More specifically, the demand rate for trips that start in queue \( i \) is represented by \( \gamma_i \), and the term \( (1 - P(N_i = k_i)) \) enforces that trips can only start in queue \( i \) if it is not full. The turning probabilities \( (p_{ij}) \) are assumed exogenous, thus the impact of the
signal plans on the travelers route choices is not accounted for. Eq. (2.9b) defines the traffic intensity. The first term of the right hand side is the traffic intensity when the queue is not full (i.e., when there is no spillback). The second term accounts for the impact in queue $i$ due to spillbacks from its downstream queues. Eq. (2.9c) gives the expression of the spillback probability (i.e., the blocking probability) as defined for an $M/M/1/k_i$ queue (see for instance, Bocharov et al. (2004, Chapter 3, pages 96-98)).

In Osorio and Chong (2015, Eq. 6), $\lambda_i$ (resp. $\rho_i$) is denoted $\lambda_i^{eff}$ (resp. $\rho_i^{eff}$) and is referred to as the effective arrival rate (resp. effective traffic intensity). In Osorio and Chong (2015), the effective traffic intensity is used to approximate the traffic intensity. In this chapter, we use the model of Osorio and Chong (2015) and we simplify the terminology by not using the term effective. We refer the reader to Osorio and Chong (2015, Section 3.3) for a discussion on how the effective traffic intensity differs from the traffic intensity.

The endogenous variables of the above system of equations are related to the decision vector (the green split vector $x$) by the following linear equations:

$$\mu_i = s \left( e_i + \sum_{j \in P_2(i)} x_j \right) \forall i \in \mathcal{L},$$

where $s$ denotes an exogenous scalar that represents the saturation flow rate (i.e., maximum queue discharge rate), and $e_i$ is an exogenous parameter that represents the ratio of fixed green time to cycle time for signalized queue $i$.

Eq. (2.10) states that the service rate of a signalized queue $i$ is given by the saturation rate scaled by the proportion of cycle time the queue has a green phase. This proportion is given by the term in parenthesis, which depends on the fixed (i.e., not optimized) time (term $e_i$) and the variable time (summation term, which represents the sum of the green splits of the signal phases of queue $i$).

We combine Eq. (2.2) and (2.8) to obtain the joint sampling pdf $g_i$ in the $\rho$ space. We use the notation $g_{\rho_1, \ldots, \rho_m}$ to indicate that the joint pdf is with regards to
\rho_1, \ldots, \rho_m. In other words, we have:

\begin{equation}
 g_{\rho_1, \ldots, \rho_m}(\rho_1, \ldots, \rho_m) = \frac{1}{\kappa_0} (\kappa_1 - f^A(\rho)).
 \end{equation}

The computation of the scalars \( \kappa_0 \) and \( \kappa_1 \) is detailed in Section 2.2.4.

Based on Eq. (2.11) we compute the corresponding joint cdf \( G_{\rho_1, \ldots, \rho_m}(\rho_1, \ldots, \rho_m) \) through integration:

\begin{equation}
 G_{\rho_1, \ldots, \rho_m}(\rho_1, \ldots, \rho_m) = \frac{1}{\kappa_0} \int_0^\rho \kappa_1 - f^A(\tilde{\rho}) d\tilde{\rho},
 \end{equation}

The inverse transform method is defined for univariate distributions, while \( G \) (of Eq. (2.12)) is an \( m \)-variate distribution. We use the method of Rosenblatt (1952) to transform an \( m \)-variate distribution into a uniform distribution on the \( m \)-dimensional hypercube. This transformation breaks down the \( m \)-variate distribution into a set of univariate conditional distributions, allowing the components of \( \rho_i \) to be sampled sequentially. The transformation is given by:

\begin{align}
 \rho_1 &= G_{\rho_1}^{-1}(u_1) \\
 \rho_2 &= G_{\rho_2|\rho_1}^{-1}(u_2) \\
 & \quad \vdots \\
 \rho_m &= G_{\rho_m|\rho_1, \ldots, \rho_{m-1}}^{-1}(u_m),
\end{align}

where \( u_j \) denotes a realization of a univariate standard uniform random variable, \( G_{\rho_i}^{-1} \) denotes the inverse of the cdf of the marginal distribution of \( \rho_i \), and \( G_{\rho_i|\rho_1, \ldots, \rho_j}^{-1} \) denotes the inverse of the cdf of \( \rho_i \) conditional on \( \rho_1, \ldots, \rho_j \). Note that in Eq. (2.13)-(2.15), the order of conditioning is defined by the user.

Before introducing the expressions for the marginal and conditional cdf's, we summarize the notation to be used hereafter:

- \( G_{\rho_i} \) marginal cdf of \( \rho_i \);
- \( G_{\rho_i|\rho_1, \ldots, \rho_j} \) conditional cdf of \( \rho_i \) conditional on \( \rho_1, \ldots, \rho_j \);
- \( \kappa_0 \) scalar normalization constant;
- \( \kappa_1 \) scalar upper bound of \( f^A \).
\( \rho_i \) traffic intensity of queue \( i \);
\( u_i \) realization of a univariate standard uniform random variable;
\( \hat{\rho}_i \) upper bound on \( \rho_i \) (exogenous);
\( k_i \) space capacity of queue \( i \) in terms of number of vehicles;
\( m \) total number of queues in the network;
\( M \) set of all queues.

By explicitly computing the integral in Eq. (2.12), we can obtain the analytical expression of the joint cdf. The derivation of the expression for the joint cdf (Eq. (2.12)) is provided in Appendix A.2.1. We define \( \hat{\rho} = (\hat{\rho}_1, \ldots, \hat{\rho}_m) \) to be the vector containing the upper bounds on the possible values of all \( \rho_i \), such that \( \rho_i \in [0, \hat{\rho}_i], \forall i \in M \). Then, the analytical expression of the marginal cdf of \( \rho_1 \) and the conditional cdf of \( \rho_i \) can be obtained:

\[
\begin{align*}
  u_1 &= G_{\rho_1}(\rho_1) \\
  &= \frac{1}{\kappa_0} \left[ (\kappa_1 \rho_1 - h(\rho_1)) \prod_{k=2}^m \hat{\rho}_k - r_1 \rho_1 \right], \\
  u_i &= G_{\rho_i|\rho_1,\ldots,\rho_{i-1}}(\rho_i|\rho_1, \ldots, \rho_{i-1}) \\
  &= \frac{\kappa_1 \rho_i}{\kappa_1 \hat{\rho}_i} \prod_{k=i+1}^m \hat{\rho}_k - q_i \rho_i - r_i \hat{\rho}_i - h(\rho_i) \prod_{k=i+1}^m \hat{\rho}_k \\ & \quad \text{for } i = 2, \ldots, m,
\end{align*}
\]

where \( q_i = \sum_{j=1}^{i-1} \frac{d h(\rho_j)}{d \rho_j} \left( \prod_{k=i+1}^m \hat{\rho}_k \right) \) for \( i = 2, \ldots, m \), \( r_i = \sum_{j=i+1}^m \left( h(\hat{\rho}_j) \prod_{k=i+1}^m \hat{\rho}_k \right) \forall i \in M \),

\[
\begin{align*}
  h(\rho_i) &= \int_0^{\rho_i} \frac{\rho_i}{1 - \rho_i} - \frac{(k_i + 1)\rho_i^{k_i+1}}{1 - \rho_i^{k_i+1}} d\rho_i \\
  &= -\rho_i - \log(1 - \rho_i) + \rho_i \log(1 - \rho_i^{k_i+1}) + \sum_{\alpha=1}^{\infty} \frac{\rho_i^{\alpha k_i + \alpha + 1}}{\alpha (\alpha k_i + \alpha + 1)} \forall i \in M,
\end{align*}
\]

\[
\frac{dh(\rho_i)}{d\rho_i} = \frac{\rho_i}{1 - \rho_i} - \frac{(k_i + 1)\rho_i^{k_i+1}}{1 - \rho_i^{k_i+1}}.
\]

The marginal cdf of \( \rho_1 \) (Eq. (2.17)) is obtained from the joint cdf by letting \( \rho_i = \hat{\rho}_i \) for \( i = 2, \ldots, m \). In other words, the variables \( \rho_i, i = 2, \ldots, m \) are marginalized out of the
joint cdf. The conditional cdf can be computed from the joint cdf by using Leibniz’s integral rule (Abramowitz and Stegun 1964, Eq. (3.3.7)) to give the expression (2.19). The variable \( q_i \) (Eq. (2.20)) and constant \( r_i \) (Eq. (2.21)) were defined to simplify the expressions of the marginal cdf (Eq. (2.17)) and the conditional cdf (Eq. (2.19)). The variable \( h(\rho_i) \) (Eq. (2.23)) corresponds the integral of the expected number of vehicles in queue \( i \) (i.e., the term in parentheses in Eq. (2.8)) with respect to \( \rho_i \). The derivative \( \frac{dh(\rho_i)}{d\rho_i} \) (Eq. (2.24)) then corresponds to the expected number of vehicles in queue \( i \). Both \( h(\rho_i) \) and \( \frac{dh(\rho_i)}{d\rho_i} \) are used to simplify the expressions in Eq. (2.17)-(2.21). A detailed derivation of the expression of the marginal cdf (Eq. (2.17)) and conditional cdf (Eq. (2.19)) is given in Appendices A.2.2 and A.2.3 respectively.

2.2.4 Algorithm

We first provide some details when implementing the inverse cdf sampling mechanism. The complete algorithm is presented in Algorithm 1.

Constrained Sampling

When generating a set of values for \( \rho \) using inverse cdf sampling, the constraints given by Eq. (2.4), (2.5), (2.9) and (2.10) have to be satisfied. Eq. (2.4) and (2.5) provide the upper and lower bounds on \( x_i \) respectively. Combining Eq. (2.5) and (2.10) allows the computation of a lower bound on \( \mu_i \), which occurs when all the phases involving queue \( i \) are assigned the minimum green split. From Eq. (2.10), the lower bound \( \mu_i^{LB} \) is then given by

\[
\mu_i^{LB} = \sum_{j \in P_2(i)} x_j^{LB} s + e_i s, \tag{2.25}
\]

where \( x_j^{LB} \) is the \( j^{th} \) component of \( x^{LB} \). Eq. (2.4) and (2.10) together can be used to derive an upper bound on \( \mu_i \) based on the maximum allocable green split for a given intersection (i.e., one phase associated with link \( i \) is allocated all the available green split for that intersection, and the rest of the phases are allocated only the minimum
green split). The upper bound $\mu^UB_i$ given by

$$\mu^UB_i = \left( \frac{c_{P_3(i)} - d_{P_3(i)}}{c_{P_3(i)}} - \sum_{j \in P_4(i)} x^LB_j \right) s + e_i s. \quad (2.26)$$

The term in brackets in Eq. (2.26) represents the maximum allocable green split, which is the fraction of cycle time left for the phase after assigning the minimum green split to the other phases, and accounting for fixed cycle time. The constraint (2.26) is not the tightest possible constraint in the sense that, at a given intersection, the sum of service rates of the queues leading to the intersection could be greater than the flow capacity of the intersection. This would manifest itself as the sum of green splits at that intersection not adding up to the cycle time of the intersection when transforming the sampled values of $\rho$ to $x$ (Step 3 of Algorithm 1). This issue is overcome by scaling the green splits proportionally, as mentioned in Appendix A.2.4.

The constraints on $\mu_i$ can be related to constraints on $\rho_i$ through Eq. (2.9b), where replacing $\mu_i$ with $\mu^LB_i$ provides the upper constraints on $\rho_i$ and vice versa:

$$\rho_i \leq \frac{\lambda_i}{\mu^LB_i} + \left( \sum_{j \in D_i} p_{ij} P(N_j = k_j) \right) \left( \sum_{j \in D_i} \rho_j \right), \quad (2.27)$$

$$\rho_i \geq \frac{\lambda_i}{\mu^UB_i} + \left( \sum_{j \in D_i} p_{ij} P(N_j = k_j) \right) \left( \sum_{j \in D_i} \rho_j \right). \quad (2.28)$$

Note that the right hand sides of the constraints (2.27) and (2.28) contain endogenous variables (e.g. $\rho_j$ for $j \neq i$), so they do not act as bounds when drawing samples of $\rho_i$. Instead, when drawing a sample for $\rho_i$ from the conditional cdf (Eq. (2.19)), the constraints (2.27) and (2.28) are only checked after all $\rho_j; j \in D_i$ have been sampled. If $\rho_i$ does not satisfy the constraints (2.27) and (2.28), it is then resampled from the conditional cdf (Eq. (2.19)) again. This process is repeated for all $i$ until a complete set of values for $\rho$ that satisfy the constraints is obtained. This method of rejection sampling (Robert and Casella 1999, Chapter 2.3) ensures that the sampling of $\rho_i$ is consistent with the sampling distribution defined by the conditional cdf.
Support of the Joint cdf

To obtain the marginal cdf of \( \rho_1 \) (Eq. (2.17)), we evaluate the integral in Eq. (2.12), and marginalize the variables \( \rho_i, i = 2, \ldots, m \) out of the joint cdf by letting \( \rho_i = \hat{\rho}_i, i = 2, \ldots, m \) (see Appendix A.2.2 for details). In other words, the support of the joint cdf (Eq. (2.12)) is taken to be \([0, \hat{\rho}]\). It should be noted that from Eq. (2.8), \( f^A(\rho) \) is undefined for any \( \rho_i = 1 \). This is true, by extension, for Eq. (2.23) and (2.24). As such, for the case studies of this chapter, we set \( \hat{\rho}_i = 0.999 \) for all \( i \). Note that in theory \( \rho_i \) can be greater than 1 (when there is hypercongestion). This occurs when the arrival rate to queue \( i \) is greater than its service rate (see Eq. (2.9b)), resulting in a build-up of congestion on queue \( i \). Hence, by assuming that the greatest value that \( \rho_i \) can take is \( \hat{\rho}_i = 0.999 \), the distribution is actually truncated. However, the truncation means there is more sampling probability mass in the region \( \rho_i < 1 \), where the arrival rate of link \( i \) is smaller than its service rate. This means that congestion is less likely to occur in queue \( i \). Thus, this can have the effect of generating \( \rho_i \) sample values that result in less congestion in queue \( i \).

Computation of \( \kappa_0 \) and \( \kappa_1 \)

Recall from Eq. (2.2) that \( \kappa_1 \) is a scalar upper bound of \( f^A \), which represents the expected number of vehicles in the network (Eq. (2.6)). The number of vehicles in each queue is bounded above by the queue's space capacity \( (k_i) \). Thus, we set as upper bound:

\[
\kappa_1 = \sum_{i=1}^{m} k_i. \tag{2.29}
\]

The normalization constant \( \kappa_0 \) is obtained by explicitly evaluating the following integral:

\[
\kappa_0 = \int_0^\hat{\rho} \kappa_1 - f^A(\rho) d\rho. \tag{2.30}
\]
The resulting expression used to compute $\kappa_0$ is given by:

$$
\kappa_0 = \kappa_1 \prod_{i=1}^{m} \hat{\rho}_i - \sum_{i=1}^{m} \left[ h(\hat{\rho}_i) \left( \prod_{j \neq i}^{m} \hat{\rho}_j \right) \right]
$$

(2.31)

The derivation of Eq. (2.31) is similar to the derivation of the joint cdf, as given in Appendix A.2.1 (Eq. (A.2.1) - (A.2.16)).

**Computation of an Infinite Summation**

When computing the value of $h(\rho_i)$ in Eq. (2.23), we have to evaluate an infinite summation term. This infinite summation term arises from evaluating the integral in Eq. (2.12) to obtain the analytical expression of the joint cdf. It is obtained from the definite integral $\int_0^{\rho_i} \log(1 - \hat{\rho}_i^{k+1}) d\hat{\rho}_i$ by integrating the Taylor expansion of the term in the integral (see Appendix A.2.1, Eq. (A.2.12) to (A.2.14) for details). Since we consider $\rho_i < 1$ and the definite integral can be evaluated to be a constant, the infinite summation term converges to a constant. For computational purposes, the infinite summation term is approximated by taking the sum of just the first 100 terms, i.e.,

$$
\sum_{\alpha=1}^{\infty} \frac{\rho_i^{\alpha k_i + \alpha + 1}}{\alpha (\alpha k_i + \alpha + 1)} \approx \sum_{\alpha=1}^{100} \frac{\rho_i^{\alpha k_i + \alpha + 1}}{\alpha (\alpha k_i + \alpha + 1)}.
$$

(2.32)

The approximation using the first 100 terms enables the sum to be computed quickly, while keeping the error small.

**Inverse cdf Sampling Algorithm**

The inverse cdf sampling algorithm for generating one realization of $x$ is summarized in Algorithm 1. In brief, a uniform random vector is transformed to $\rho$ using the inverse transform method as specified by Eq. (2.13)-(2.15). After sampling each $\rho_i$, the constraints on $\rho_i$ (Eq. (2.27) and (2.28)) are checked. If the constraints are not satisfied, $\rho_i$ is resampled based on a newly generated uniform random number according to Eq. (2.19). After all the constraints have been satisfied, the complete set of values for $\rho$ is transformed into the set of values of $x$ using a pseudoinverse as
described in detail in Appendix A.2.4.

2.3 Validation

To validate the proposed method we consider a synthetic toy network defined in Osorio and Yamani (2017). The network is depicted in Figure 2-2. It consists of 20 single-lane links and 4 intersections, with a main arterial (horizontal links) and 4 side roads (vertical links). Travel demand is defined such that vehicles only have straight paths (i.e., they do not make any right or left turnings at the intersections). The signal plans of each intersection are thus composed of two signal phases (i.e., there are no signal phases for turning movements). Origins and destinations of trips occur at the boundaries of the network which are represented by the circles in Figure 2-2. The queueing representation of the network is displayed in Figure 2-3. Each rectangle represents a queue. The arrows show the possible traffic movements or turns. The numbers in the rectangles represent the signal phase index (i.e., the index of the signal phase for which the underlying traffic movement has green). The numbers in brackets outside the rectangles denote the indices of the origin and destination nodes. Table 2.4 shows the origin-destination (OD) travel demand of the network. OD pairs which are not included in Table 2.4 have a demand of zero.

First, we compare an estimate of the simulation-based objective function to its analytical approximation (i.e., we compare \( f \) of Eq. (2.3) to \( f^A \) of Eq. (2.8)). This provides insights on how well \( f^A \) approximates \( f \). For each intersection, the two signal phases are constrained by the linear equality constraint Eq. (2.4). Thus, there is only one degree of freedom per intersection. This leads to a total of four degrees of freedom. Figure 2-4 displays contour plots for each of the 2-dimensional projections of the objective function, while keeping the other two signal phases fixed at the histogram bin center value where the inverse cdf sampling assigns the highest marginal probability as shown in Figure 2-5a. Details on how the plots are generated are given in Appendix A.3. In both Figures 2-4a and 2-4b, the first (i.e., top) row of plots display the analytical approximations, while the second (i.e., bottom) row display
Algorithm 1: Inverse cdf sampling

0. Initialization

(a) Initialize all exogenous parameters
   Initialize $x^{LB}$, $s$, $e_i$ ($\forall i \in \mathcal{L}$), $p_{ij}$ ($\forall i, j \in \mathcal{M}$) for the underlying network
   Initialize $c_\ell$ and $d_\ell$ for all intersections $\ell$
   Set $\hat{\rho}_i = 0.999$, $\forall i \in \mathcal{M}$ (see Section 2.2.4)
   Set $\kappa_1$ according to Eq. (2.29)

(b) Generate standard uniform random vector $\mathbf{U} = (u_1, \ldots, u_m) \in [0, 1]^m$

(c) Compute $h(\hat{\rho}_i), \forall i \in \mathcal{M}$ according to Eq. (2.23)

(d) Compute $r_i, \forall i \in \mathcal{M}$ according to Eq. (2.21)

(e) Compute the normalization constant $\kappa_0$ according to Eq. (2.31)

1. Compute inverse of marginal cdf for the first component $\rho_1$

   (a) Solve Eq. (2.17) to obtain $\rho_1$

2. Compute inverse of conditional cdf for the remaining components $\rho_i$ for $i = 2, \ldots, m$,

   (a) Compute $\frac{\partial h(\rho_{i-1})}{\partial \rho_{i-1}}$ according to Eq. (2.24)

   (b) Compute $q_i$ according to Eq. (2.20)

   (c) Conditioning on the sampled components $\rho_1, \ldots, \rho_{i-1}$, solve Eq. (2.19)
      to obtain $\rho_i$

   (d) Check constraints on $\rho_i$ if $\rho_j, j \in \mathcal{D}_i$ have been sampled
      i. Check that $\rho_i$ satisfies the constraints (2.27) and (2.28)
      ii. if constraints are met
          continue to Step 3
      else
          while constraints are not met
          A. Let $\Psi$ denote the set of indices of the components which do not satisfy the constraints (2.27) and (2.28)
          B. Generate standard uniform random vector $\mathbf{U} \in [0, 1]^{|\Psi|}$
          C. for $i \in \Psi$,
             Repeat Steps 2a-2c for component $i$
          D. Check that $\rho_i$ satisfies the constraints (2.27) and (2.28)
Algorithm 2: Inverse cdf sampling (continued)

3. **Transform \( \rho \) to \( x \)**
   
   Input: a vector of \( \rho \) values \( (\rho_1, \ldots, \rho_m) \)

   (a) Compute \( P(N_i = k_i), \forall i \in \mathcal{M} \) according to Eq. (2.9c)

   (b) Compute \( \lambda_i, \forall i \in \mathcal{M} \) according to Eq. (2.9a)

   (c) Compute \( \mu_i, \forall i \in \mathcal{M} \) according to Eq. (2.9b)

   (d) Let \( \Xi \) be the matrix with elements \( \xi_{ij} = 1 \) if \( j \in P_2(i), 0 \) otherwise;
       \( \mu = (\mu_1, \ldots, \mu_m) \); and \( e = (e_1, \ldots, e_m) \). Then, using a pseudoinverse of
       \( \Xi \), compute \( x \) according to Eq. (2.10) (see Appendix A.2.4, Eq. (A.2.45)
       for details)

---

Figure 2-2: Toy network

...the simulation-based estimates. For each row of plots, the mapping of colors to
numerical values is displayed on the right. Figure 2-4 indicates that for all pairs of
signal phases, the analytical and the simulation-based functions have similar trends,
with the gradient of the function having the same sign when comparing a given
region of a projection. The contour lines also have similar patterns, indicating that
the minimum identified by the analytical model is close to that of the simulator. Note
however that the values represented on the color maps differ by an order of magnitude.
This indicates that the magnitude of the functions differ. Recall from Figure 2-1 that
the inverse cdf sampling is based on a comparison of values of \( f^A \). In other words, \( f^A \)
is used to identify regions of the feasible region that have better objective function values. Thus, the most important aspect is for $f^A$ to capture the relative trends of $f$. It is not necessary for it to accurately approximate the magnitude of $f$.

Let us now compare the marginal sampling distributions obtained from the analytical model to those of the simulator and to those of a uniform sampling mechanism. We make comparisons with a uniform sampling mechanism, as it is the most commonly used exploration mechanism in SO (e.g., Sebastiani, Lüders, and Fonseca 2016, Osorio and Chong 2015, Paz et al. 2015). The gray bars of Figure 2-5a display an estimate of the marginal sampling distribution obtained from the analytical approximation $f^A$. The estimates are obtained from 1000 realizations (or draws) of the decision vector. The x-axis displays the values of the corresponding decision variable (i.e., the green split of the corresponding signal phase) and the y-axis is the estimated sampling probability. For comparison, we also display an estimate of a uniform sampling mechanism (displayed as white bars). To sample uniformly from the feasible region (defined by Eq. (2.4)-(2.5)) we use the sampling mechanism of Stafford (2006). Figure 2-5b displays the marginal sampling distribution as estimated using the simulator-based objective function values. Note that the scale on the y-axis of Figure 2-5b is different from that of Figure 2-5a, so as to allow the distributions to be
Figure 2-4: Two-dimensional projections of the objective function surface as estimated by the analytical model and the simulation model.
seen more clearly. More details on the experimental design underlying these figures are given in Appendix A.3.

From Figure 2-5a, we see that the sampling distribution of the proposed approach differs from that of a uniform sampling distribution. We also note that, due to the linear cycle time constraint (Eq. (2.4)), the shapes of the distributions for phases 2, 4, 6 and 8 are mirror images of those of phases 1, 3, 5 and 7 respectively. Hence, we focus our analysis on phases 1, 3, 5 and 7. The gray bars for phases 1, 5 and 7 in Figure 2-5a show that the proposed method assigns a high probability to large green split values for phases 1, 5 and 7. This means that it favors the traffic along the main arterial, as opposed to that of the side roads. This is consistent with the corresponding travel demand (Table 2.4), which is defined such that the arterial has a higher demand compared to the side roads for these intersections. The arterial has a demand of 700 veh/h, while the side roads have demands of 100 veh/h, thus justifying the higher probability that phases 1, 5 and 7 are assigned values greater than 0.8. For phases 3 and 4, the proposed method assigns almost equal amounts of green time to each phase. Again this is consistent with the underlying demand for these intersections: the main arterial and the side road at the second intersection are almost equal, with demands of 700 veh/h and 600 veh/h respectively.

The distribution as estimated using the simulation-based objective function values (Figure 2-5b) has the same trends as that of the proposed method. Namely: (i) for phases 1, 5 and 7, a higher sampling probability is assigned to signal plans that favor the traffic along the main arterial compared to the side roads, and (ii) for phases 3 and 4, both roads (arterial and side) are equally favored. Comparing the distribution of the proposed method to that of the simulator, we also observe that the modes of each phase match. This indicates that the proposed method assigns a higher probability of sampling in the region where the minimum is located than uniform sampling. However, it should be noted that the detailed shapes of the proposed method’s distributions (Figure 2-5a) and the simulation-based distributions (Figure 2-5b) differ. The simulation-based distributions tend to be flatter compared to the distributions of the proposed method.
Figure 2-5: Comparison of marginal green split distributions (a) obtained using the randomly generated sample points and (b) estimated using the simulation-based objective function values.
2.4 Case Study: Midtown Manhattan Signal Control

2.4.1 Experimental Design

We now apply our proposed method to a signal control problem for the area of Midtown Manhattan (MTM) in New York City. The area of interest is demarcated by a rectangle in the map of Figure 2-6. We simulate traffic from 3pm - 6pm and control the signal plans for the peak hour of 5pm-6pm. We control the green splits of 97 intersections for a total of 259 green splits (i.e., decision variables). Osorio and Chong (2015) tackled a problem with 17 intersections and 99 green splits, which is considered to be high-dimensional in the field of signal control. Furthermore, problems with around 200 dimensions are considered high-dimensional in the field of continuous SO (Wang, Wan, and Chang 2016). Thus, this is considered a high-dimensional signal control SO problem.

The network topology of the simulator is displayed in Figure 2-7. The MTM simulation model, which is implemented in the Aimsun software (TSS-Transport Simulation Systems 2015), consists of a total of 698 roads, 2756 lanes and 444 intersections.
In the simulated hour of 5-6pm, the expected demand is over 21700 trips per hour, distributed across more than 3500 origin-destination pairs. In this simulation model, all of the green times have a lower bound of 6 seconds. The corresponding components in the vector of minimal green splits $\mathbf{x}^{LB}$ (Eq. (2.5)) are the ratio of 6 seconds to the cycle time for the intersection which the green split belongs to. Additional implementation details are provided in Appendix A.4.

To evaluate the added value of the proposed exploration mechanism, we embed it within a simulation-based optimization (SO) algorithm. The SO algorithm used is based on the metamodel SO algorithm of Osorio and Bierlaire (2013). A metamodel is a parametric analytical approximation of the simulation-based objective function. At each iteration, the SO algorithm can either evaluate an exploration point or an exploitation point. The exploration point is generated through the user-defined explo-
ration mechanism (e.g., inverse cdf sampling mechanism or uniform sampling mechanism). The exploitation point is obtained by solving the metamodel optimization problem. Hereafter, solving the metamodel optimization problem is referred to as exploitation. For more details on the SO algorithm used here, we refer the reader to Appendix A.5.

We consider 4 instances of this SO algorithm that differ in 2 ways: (i) whether or not the information from the analytical model is used to specify the exploration mechanism, and (ii) whether or not the information from the analytical model is used to specify the metamodel to enhance exploitation. These 4 algorithm instances are summarized in Table 2.5. The first column defines the name of the algorithm instance. Column 2 indicates whether or not the exploration mechanism uses information from the analytical model. If it does, then the proposed inverse cdf sampling mechanism is used. If it does not, then a uniform sampling mechanism is used. The uniform sampling mechanism has been traditionally used as part of the SO algorithm of Osorio and Bierlaire (2013). Column 3 indicates whether or not the metamodel uses information from the analytical model to enhance exploitation. If it does, then the metamodel is denoted \( m \) and is defined as the sum of a scaled function of \( f^A \) and a quadratic polynomial. If it does not, the metamodel is denoted \( \phi \) and is a quadratic polynomial. These two metamodel definitions have been traditionally used for metamodel SO (see e.g., Osorio and Bierlaire (2013, Eq. 3 and 4)). In addition, the notation of \( m \) and \( \phi \) is consistent with past work (e.g., Osorio and Bierlaire 2013, Osorio and Chong 2015, Zhang, Osorio, and Flötteröd 2017). The method Unif-\( m \) is that used in past signal control SO work (Osorio and Bierlaire 2013, Osorio and Chong 2015, Osorio and Nanduri 2015a,b, Chen, Osorio, and Santos 2019), while the method Unif-\( \phi \) served as the benchmark in those past works.

The algorithms of rows 1 and 2 have a common general-purpose metamodel, which does not use the problem-specific information from the analytical model to enhance exploitation. Thus, their comparison serves to evaluate the added value of complementing a general-purpose SO algorithm with a problem-specific exploration mechanism. The algorithms of rows 3 and 4 have a common metamodel that uses
problem-specific information from the analytical model for exploitation. Thus, their comparison serves to evaluate the added value of using a problem-specific exploration mechanism given that the metamodel already has a problem-specific component.

For each algorithm instance, we perform experiments that allocate either 10%, 50% or 100% of the computational budget to exploration, while the remaining computational budget (i.e., 90%, 50% and 0%, respectively) is used for exploitation (i.e., simulate the solutions of the metamodel optimization problems). The computational budget is defined in terms of the number of simulation runs assigned to evaluate exploration points and exploitation points.

We first consider the case of 100%. In this case, there is no computational budget assigned to exploitation. In other words, only exploration is done. This allows us to directly compare the ability of the proposed exploration mechanism to that of a uniform sampling mechanism in finding good exploration points. We then consider the cases of 50%, where exploration takes place every other iteration of the SO algorithm, and 10%, where exploration takes place every 10th iteration.

To estimate the objective function, for each signal plan, we take the sample mean from 5 simulation replications. For each replication, we compute the average (over time) number of vehicles in the network during the simulated interval of 5pm - 6pm. This average is obtained from observations of the number of vehicles in the network.
collected every minute. We consider a tight computational budget of 250 simulation evaluations, i.e., each algorithm run is terminated once a total of 250 simulation evaluations are carried. Since we use 5 simulation replications per point, this means that the computational budget allows to simulate a total of 50 points. For the 100% exploration setting, we consider 1 initial point. For the 50% and 10% exploration settings, we consider 3 initial points. The initial points are randomly and uniformly sampled from the feasible region using the code of Stafford (2006). For each initial point, we run each SO algorithm 3 times.

2.4.2 Numerical Results

For each algorithm run, we identify the best solution and evaluate its performance based on 50 simulation replications. These 50 simulation replications are then used to construct the cumulative distribution function (cdf) for each solution. In Figures 2-8 - 2-11, the x-axis represents the average number of vehicles in the network, i.e., the simulation-based estimate of the objective function. For a given value along the x-axis, the corresponding value on the y-axis shows the fraction of simulation replications (out of 50) that are smaller than the value on the x-axis. Since the objective is to minimize the expected number of vehicles in the network, the more the cdf of a given signal plan lies to the left of the figure, the better the performance of the underlying signal plan.

Figure 2-8 shows the cdf’s for 100% exploration (i.e., when no exploitation is done). The red dashed line represents the uniform sampling, while the solid line denotes inverse cdf sampling. This 100% exploration setting allows us to directly compare the ability of uniform sampling and inverse cdf sampling to identify good exploration points. In Figure 2-8, the cdf’s of inverse cdf sampling are all to the left of those of uniform sampling. This indicates that, for all 3 runs, inverse cdf sampling outperforms uniform sampling. This shows that the information of the analytical model increases the chance of finding exploration points with good performance. Thus, there is an added value of using problem-specific analytical information to specify the exploration mechanism.
Figure 2-8: Cdf’s of the average number of vehicles in the network achieved by the best signal plans generated by the uniform sampling mechanism and the inverse cdf sampling mechanism.

Note that at the 100% exploration setting, the inverse cdf sampling mechanism and the uniform sampling mechanism are the same as the Invcdf-$\phi$/Invcdf-$m$ and Unif-$\phi$/Unif-$m$ methods respectively (since no exploitation is done). The difference in performance between the inverse cdf sampling mechanism and the uniform sampling mechanism in Figure 2-8 is hence representative of the difference in performance between the Invcdf-$\phi$/Invcdf-$m$ and Unif-$\phi$/Unif-$m$ methods at 100% exploration. This difference in performance between the Invcdf-$\phi$ and Unif-$\phi$ methods carries over to the 50% and 10% exploration cases too. For a more detailed comparison, we refer the reader to Appendix A.6. Given that Unif-$\phi$ performs much worse than Invcdf-$\phi$, we will not compare Unif-$\phi$ with the other methods in the subsequent figures.

The results of the experiments with exploration proportions of 50% and 10% for the Invcdf-$\phi$ and Unif-$m$ methods are given in Figure 2-9. To further determine if the differences in performance of the proposed plans are statistically significant, we performed a one-sided paired $t$-test based on the average values of the 3 runs for each initial point. Here, the null hypothesis states that the mean of the 50 simulation replications (averaged over the 3 runs for a given initial point) using the signal plans
proposed by Invcdf-ϕ is not smaller than that of Unif-m, while the alternative hypothesis states that the mean of the 50 simulation replications using the signal plans proposed by Invcdf-ϕ is smaller than that of Unif-m. The resulting t-statistics are shown in Table 2.7. Each row considers a given initial point. The second column shows the t-statistics for 50% exploration, while the third column shows the t-statistics for 10% exploration. Each t-test is considered at the 10% level of significance, with 49 degrees of freedom, leading to a critical value of -1.299. The t-tests with t-statistics smaller than the critical value at 10% level of significance are displayed in bold, indicating that the signal plans proposed by Invcdf-ϕ are statistically better than those of Unif-m.

Figure 2-9 shows that the signal plans proposed by the Invcdf-ϕ method are generally similar or better in performance than those proposed by Unif-m. This is true for the 50% exploration case as seen in Figure 2-9. The results of the t-tests (Table 2.7) also show that the signal plans proposed by Invcdf-ϕ are statistically better than those of Unif-m for 2 out of 3 initial points. For initial point 1, the signal plans proposed by both methods have similar performance. In the 10% exploration setting, the signal plans proposed by Invcdf-ϕ performed worse than those of Unif-m for initial point 1 (Figure 2-9b). However, Invcdf-ϕ identified signal plans that are statistically better than those of Unif-m for initial points 2 and 3, as seen in Table 2.7. The comparison between Invcdf-ϕ and Unif-m shows that if one had to choose between using information of the analytical model only in the exploration mechanism and using the information only during exploitation, then it can be advantageous to choose the former.

We now consider whether it is beneficial to use the information from the analytical model for both exploration and exploitation (i.e., Invcdf-m). We compare the

<table>
<thead>
<tr>
<th>Initial Point</th>
<th>50% Exploration</th>
<th>10% Exploration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.2412</td>
<td>5.3719</td>
</tr>
<tr>
<td>2</td>
<td>-3.3832</td>
<td>-4.0747</td>
</tr>
<tr>
<td>3</td>
<td>-4.4261</td>
<td>-2.3207</td>
</tr>
</tbody>
</table>
Figure 2-9: Cdf’s of the average number of vehicles in the network for the Invcdf-$\phi$ and Unif-$m$ methods using 50% and 10% exploration settings and considering 3 random initial points.
Table 2.8: T-statistics for paired t-test (Invcdf-\(m\) vs. Unif-\(m\))

<table>
<thead>
<tr>
<th>Initial Point</th>
<th>50% Exploration</th>
<th>10% Exploration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.3121</td>
<td>-1.5955</td>
</tr>
<tr>
<td>2</td>
<td>-2.9563</td>
<td>-4.5719</td>
</tr>
<tr>
<td>3</td>
<td>-2.8764</td>
<td>-2.0474</td>
</tr>
</tbody>
</table>

performance of signal plans proposed by Invcdf-\(m\) with those proposed by Unif-\(m\) and Invcdf-\(\phi\). For clarity purposes, the results are plotted on two separate figures. Figure 2-10 compares Invcdf-\(m\) and Unif-\(m\), while Figure 2-11 compares Invcdf-\(m\) and Invcdf-\(\phi\).

The plots in Figure 2-10 compare Invcdf-\(m\) and Unif-\(m\). Similar to the comparison between Invcdf-\(\phi\) and Unif-\(m\), we also perform one-sided, paired \(t\)-tests to determine if the signal plans proposed by Invcdf-\(m\) are significantly better than those proposed by Unif-\(m\). The \(t\)-statistics are displayed in Table 2.8. The critical value is -1.299, and \(t\)-statistics with values below the critical value are shown in bold, which indicates that the signal plans proposed by Invcdf-\(m\) are statistically better than those of Unif-\(m\).

Figure 2-10 shows that the plans proposed by the Invcdf-\(m\) method perform similar to or better than those proposed by Unif-\(m\). For the 50% exploration setting, the plans proposed by Invcdf-\(m\) outperform those proposed by Unif-\(m\) for 2 out of 3 initial points, as shown in Table 2.8. The difference in performance is more pronounced for the 10% exploration setting. Table 2.8 shows that the plans proposed by Invcdf-\(m\) outperform those proposed by Unif-\(m\) for all initial points in the 10% exploration setting. A possible reason for the bigger difference in performance under the 10% exploration setting is that Unif-\(m\) is over-exploiting as a result of the greater proportion of the computational budget allocated to exploitation. This leads to Unif-\(m\) getting trapped in local optima with poorer performance compared to the signal plans identified by Invcdf-\(m\). Invcdf-\(m\) is able to perform better than Unif-\(m\), since information from the analytical model is used to enhance exploration as well, giving it a better balance between exploration and exploitation. As such, the results of Figure 2-10 and Table 2.8 show that using the information from the analytical model in the exploration mechanism can lead to improvements, even if the information is already being
Figure 2-10: Cdf’s of the average number of vehicles in the network for the InvCdf-m and Unif-m methods using 50% and 10% exploration settings and considering 3 random initial points.
used for exploitation. This is especially true when the exploration proportion of the computational budget low.

Figure 2-11 compares the signal plans proposed by Invcdf-φ and Invcdf-m. The results of the one-sided, paired \( t \)-tests, comparing the signal plans proposed by Invcdf-φ and Invcdf-m, are provided in Table 2.9. For the 50% exploration setting, the plots in Figure 2-11 indicate that the signal plans proposed by Invcdf-m have similar performance to those proposed by Invcdf-φ. This is confirmed by the results of the \( t \)-tests in Table 2.9, which shows that there is no significant improvement in performance of the signal plans proposed by Invcdf-m, compared with Invcdf-φ, for all the initial points in the 50% exploration setting. In the case of the 10% exploration setting, Figure 2-11b shows that the signal plans proposed by Invcdf-m are better than those of Invcdf-φ for initial point 1. The difference is significant as shown by the \( t \)-test for initial point 1 (Table 2.9). However, for initial points 2 and 3, there is no significant difference in performance of the signal plans proposed by Invcdf-m and Invcdf-φ in the 10% exploration setting.

The results of Figure 2-11 and Table 2.9 shows that there is limited value to using information from the analytical model for both exploration and exploitation when the exploration proportion of the computational budget is large. When the exploration proportion is low, using the information from the analytical model for exploitation, in addition to exploration, may lead to some improvements.

In summary, the results show that methods which use information from the analytical model to enhance exploration through the inverse cdf sampling mechanism leads to better solutions than methods that use a uniform sampling mechanism. This highlights the benefits of using problem-specific structural information in the exploration mechanism of the optimization algorithm to quickly identify good solutions.
Figure 2-11: Cdf’s of the average number of vehicles in the network for the Invcdf-$m$ and Invcdf-$\phi$ methods using 50% and 10% exploration settings and considering 3 random initial points.
within a limited number of iterations.

2.4.3 Computational Efficiency

Each iteration of the SO algorithm involves three computationally intensive tasks: (i) identifying an exploitation point by solving the metamodel optimization problem; (ii) generating an exploration point with the exploration mechanism; (iii) using the traffic simulator to evaluate the new points derived from (i) or (ii). Here, we compare the computational runtime required for these three tasks, with a total of 270 observations for exploration, 630 observations for exploitation and 4500 observations for simulations collected while running the experiments of Section 2.4.2. The experiments were run on a laptop computer with an Intel Core i7 (2.30 GHz) CPU and 12 GB of RAM. Figure 2-12 displays the cdf’s for the computational runtimes for the respective tasks. The x-axis represents the computational runtime required for the task in seconds. For a given value along the x-axis, the corresponding value on the y-axis shows the fraction of observations that had runtimes smaller than the value on the x-axis. The red dashed line represents the Unif- method, the black solid line represents the Invcdf- method, the blue line with stars denotes the Unif- method and the magenta line with circles denotes the Invcdf- method.

Figure 2-12a shows the runtimes needed for exploration. Figure 2-12b shows a zoomed-in section of the plot so that the curves for Unif- and Unif- can be seen in more detail. The Unif- (red dashed line) and Unif- (blue line with stars) methods, which make use of uniform sampling, has an overall mean runtime of 0.1 seconds. On the other hand, the Invcdf- (black solid line) and Invcdf- (magenta line with circles) methods use inverse cdf sampling, which has an overall mean runtime of 84 seconds. Thus, drawing a sample using the inverse cdf sampling mechanism takes about 83.9 seconds longer than drawing a sample using uniform sampling. The step-like appearance of the empirical cdf curves for the Invcdf- and Invcdf- methods is the result of having to resample components which do not satisfy the constraints (see Step 2d in Algorithm 1); in some cases, some components have to be resampled multiple times. In Figure 2-12a, the difference between the curves for Invcdf- and
Figure 2-12: Computational runtimes for (a, b) exploration (per exploration point), (c, d) exploitation, (e) simulation and (f) total computational runtime per iteration.
Invcdf-\(m\) can be attributed to the randomness when generating exploration points, since the inverse cdf sampling mechanism used in the two methods are the same, and does not depend on the metamodel in use or previous simulation evaluations.

Figure 2-12c illustrates the runtimes required to solve the metamodel optimization problem during an exploitation iteration. Figure 2-12d shows a zoomed-in section of the plot so that the curves for Unif-\(\phi\) and Invcdf-\(\phi\) can be seen in more detail. The Unif-\(\phi\) and Invcdf-\(\phi\) methods, which make use of the \(\phi\)-metamodel for optimization, tend to have very fast mean runtimes of 1.4 seconds and 1.0 seconds respectively. This due to the fact that the \(\phi\)-metamodel consists of a quadratic polynomial which is relatively easy to optimize. Figure 2-12d also shows that metamodel optimization for the Invcdf-\(\phi\) method is faster than the Unif-\(\phi\) method. On the other hand, the Unif-\(m\) and Invcdf-\(m\) methods, which use the \(m\)-metamodel for optimization, have mean runtimes of 1764 seconds and 1694 seconds respectively. Based on Figure 2-12c, it can also be seen that metamodel optimization for the Unif-\(m\) and Invcdf-\(m\) methods have runtimes that seem to be distributed in a bimodal fashion. Metamodel optimization steps that took less than 1500 seconds have mean runtimes of 427 seconds and 343 seconds respectively for the Unif-\(m\) and Invcdf-\(m\) methods, while metamodel optimization steps that took 1500 seconds or longer have mean runtimes of 3570 seconds and 3189 seconds respectively for the Unif-\(m\) and Invcdf-\(m\) methods. The longer runtimes for the metamodel optimization steps tend to occur in the later iterations of the algorithm runs. This suggests that \(m\)-metamodel becomes takes longer to optimize when it has been parametrically fitted to a number of points (e.g., 15).

Figure 2-12e shows that the runtimes needed for simulation are quite similar for all the SO methods. Each simulation evaluation has a mean runtime of 466 seconds.

Finally, Figure 2-12f shows the total computational runtimes needed per iteration of the SO algorithm. The runtime for a given iteration is computed by summing the mean of the simulation runtimes (since the simulation replications can be run in parallel), and the runtime for exploration or exploitation (depending on whether it is an exploration iteration or exploitation iteration). The mean runtimes per iteration
required by the Unif-\(\phi\) and Invcdf-\(\phi\) methods are 481 seconds and 509 seconds respectively, while the mean runtimes of the Unif-\(m\) and Invcdf-\(m\) methods are much longer, with mean runtimes of 1450 seconds and 1434 seconds respectively. The runtimes of the Unif-\(m\) and Invcdf-\(m\) methods are again bimodal due to the metamodel optimization step. Iterations which took less than 2000 seconds have a mean runtime of 630 seconds and 601 seconds for the Unif-\(m\) and Invcdf-\(m\) methods respectively, while iterations which took 2000 seconds or longer have a mean runtime of 4024 seconds and 3690 seconds for the Unif-\(m\) and Invcdf-\(m\) methods respectively.

In Section 2.4.2, we saw that using information from the analytical model for exploration (i.e., Invcdf-\(\phi\)) leads to similar or better performance (in terms of objective function value) compared to using the information for exploitation (i.e., Unif-\(m\)) for both 50% and 10% exploration (Table 2.7). When combining this observation with the computational runtimes, we see that it is beneficial to use the information for exploration, rather than exploitation. Figure 2-12f shows that the Invcdf-\(\phi\) method requires much less computational runtime (509 seconds per iteration on average) compared to the Unif-\(m\) method (1450 seconds per iteration on average). This corresponds to a 65% reduction in computational runtime per iteration on average.

Furthermore, we showed in Section 2.4.2 that using information from the analytical model for both exploration and exploitation (i.e., Invcdf-\(m\)) identifies signal plans that are statistically indifferent in performance compared to those of Invcdf-\(\phi\) under the 50% exploration setting (Table 2.9). In the 10% exploration setting, Invcdf-\(m\) is statistically better than Invcdf-\(\phi\) for 1 out of 3 initial points, representing a slight improvement. However, Invcdf-\(m\) would require 1778 seconds on average to generate a sample point and optimize once (i.e., 84 seconds to sample by inverse cdf sampling and 1694 seconds to optimize the metamodel), compared to 85 seconds for Invcdf-\(\phi\) (i.e., 84 seconds to sample by inverse cdf sampling and 1 second to optimize the metamodel). This represents a 2100% increase in computational cost. Hence, while Invcdf-\(m\) may find solutions slightly better than those of Invcdf-\(\phi\), it requires significantly more runtime compared to Invcdf-\(\phi\).

In summary, when taking into account both the signal plan performance and
computational runtimes of the different methods, we see that using information from the analytical model in the exploration mechanism (Invcdf-φ) provides the best trade-off between performance and runtime. Using the information for both exploration and exploitation (Invcdf-m) can potentially lead to better solutions, but with a significant increase in computational costs.

2.5 Conclusion

This chapter studies the question of whether problem-specific structural information from analytical models should be used in the exploration mechanism and/or the exploitation mechanism of the optimization algorithm when attempting to solve a transportation SO problem. We also present an exploration mechanism for efficient high-dimensional stochastic simulation-based transportation optimization. It puts forward the idea of making use of an analytical model, which contains structural information of the traffic network, in the exploration mechanism of the optimization framework. This increases the probability of sampling points with good performance, while still having a non-zero chance of sampling in other regions for exploration.

We evaluated the proposed approach on a fixed-time signal control problem of Midtown Manhattan. The results show that using the problem-specific structural information in the exploration mechanism via an inverse transform method was able to achieve similar or better performance to the existing method of using the structural information in the exploitation mechanism. This was in addition to reducing the overall amount of computational runtime required. Thus, the inverse cdf sampling mechanism can be used as part of an optimization framework to quickly and efficiently identify solutions with good performance, when there exists prior information on the structure of the problem.

While the use of inverse cdf sampling has shown promising results, there are some limitations to its use. For instance, since the objective function has to be integrated in order for the inverse cdf sampling to work, there is a need to find an objective function whose integral has a closed analytical form.
Another area that can be studied more is the order in which the conditionals (as shown in Eq. (2.13) - (2.15)) are taken. It is possible that the order of the conditionals has an implication on the sampling distribution. Although randomizing the order for sampling with the toy network did not produce any noticeable changes, this cannot be said to be true for other networks.

Moving forward, we will investigate possible ways to adapt the inverse cdf sampling method into a sequential design. This could be done in a way similar to the expected improvement criterion in the Efficient Global Optimization algorithm (Jones, Schonlau, and Welch 1998). The idea is to take into account points that have already been evaluated by the simulator, so as to update the analytical model. Based on the updated analytical model, the exploration mechanism would then be able to pick out the points with better performance with greater probability. At the same time, it would also be useful to further incentivize the exploration mechanism to sample points far away from previously evaluated points. This is especially so after finding a good solution, so that there would be a better exploration-exploitation balance, which could lead to even better solutions.
Chapter 3
Bayesian Optimization Techniques for High-dimensional Transportation Problems

This chapter explores the use of BO with Gaussian process (GP) models to tackle high-dimensional transportation problems. It proposes formulations of the prior mean function and covariance function of the GP that enable BO to incorporate problem-specific transportation information, while remaining computationally tractable. This is done through the use of an analytical surrogate model. We also investigate the effects of different forms of bias in the surrogate model on the BO performance.

3.1 Introduction

This chapter considers high-dimensional simulation-based problems, where the objective function is defined by a stochastic urban mobility simulator. These simulators are commonly used by transportation agencies and operators (e.g. ridesharing operators) to evaluate changes to their network designs or operations (Stone 2021, TSS-Transport Simulation Systems 2019, 2009). For instance, transportation agencies may use simulators to evaluate traffic management strategies, such as congestion pricing and traffic signal control. Ridesharing operators may use simulators to evaluate new algorithms
before releasing it to production (Greenhall 2016). These simulators can embed detailed models of traveler behavior, such as mode choice, route choice and response to real-time traffic conditions. Furthermore, the resolution of the simulation models, as well as the ability to simulate large-scale networks, are constantly improving. This makes it all the more enticing for transportation agencies and operators to make use of simulators in their planning and operations. For an in-depth review of existing traffic simulation models, we refer the reader to Barceló et al. (2010).

However, the higher simulation model resolution and greater spatial coverage of networks being simulated also lead to increasing computational demand of simulators. When used within simulation-based optimization (SO) frameworks, the computational cost of evaluating the simulator becomes the main computational bottleneck. Few function evaluation (i.e. few simulation runs) are carried out when using these simulators for SO.

At the same time, transportation agencies and operators are interested in optimizing for entire urban networks. For some problems, such as urban traffic signal optimization (Osorio and Chong 2015) and origin-destination (OD) matrix calibration (Zhang, Osorio, and Flötteröd 2017, Lu et al. 2015), this leads to optimization problems of high dimensions. Hence, this points to the need for algorithms that perform well for high-dimensional problems and few function evaluations.

In this work, we consider high-dimensional simulation-based optimization problems that have continuous and general (i.e. non-convex) objective functions, with unknown analytical forms. The constraints are assumed to be analytical and differentiable, and are typically convex. In the case study of this chapter (Section 3.4), they consist of bound and linear equality constraints. Such a problem can generally be formulated as:

$$\min_{x \in \chi} f(x, z; p) \equiv \mathbb{E}[F(x, z; p)],$$

(3.1)

where $f$ is the objective function, $F$ represents the stochastic output of a simulation run, $x$ is the high-dimensional vector of continuous decision variables, $\chi$ is the feasible...
region, \( z \) denotes the vector of endogenous simulation variables and \( p \) represents the vector of deterministic exogenous parameters. In Eq. (3.1), one simulation run represents the realization of \( F \). The exogenous parameters and endogenous variables are specific to the problem in question. Examples for a traffic signal control problem are provided in Section 3.4.1.

We consider a context with fewer function evaluations than the problem dimension (i.e. a tight computational budget). In this context, it is crucial for the optimization algorithm to balance exploration and exploitation so as to find a good solution within the limited computational budget. In the context of optimization, exploration refers to the search in regions with few evaluated points, while exploitation refers to searching in regions with good estimated performance (Sun, Hong, and Hu 2014).

3.1.1 High-Dimensional Bayesian Optimization

Bayesian optimization (BO) was first popularized by the efficient global optimization algorithm proposed by Jones, Schonlau, and Welch (1998). In recent years, BO has become widely-used for tackling global optimization problems where the objective function is, computationally or financially, expensive to evaluate (Shahriari et al. 2015), and hence the number of objective function evaluations allowed is small. It provides an efficient approach for the optimization of expensive-to-evaluate functions (Sasena, Papalambros, and Goovaerts 2002, Jones, Schonlau, and Welch 1998). The efficiency stems from being able to incorporate prior belief about the problem to inform the search for the optimum, as well as BO’s ability to balance between exploration and exploitation when optimizing the objective function (Brochu, Cora, and De Freitas 2010).

Most often, Gaussian process (GP) models are used as part of BO to approximate the unknown objective function. GPs provide an attractive way to construct an approximation of the objective function (Shahriari et al. 2015), as analytical expressions are available for the GP posterior. This includes an analytical estimate of the variance of the GP posterior predictions, which is used in BO to balance the exploration-exploitation trade-off. Other models have been used to approximate the
objective function, such as the tree Parzen estimator (Bergstra, Yamins, and Cox 2013, Bergstra et al. 2011) and random forests (Shahriari et al. 2015, Hutter 2009), but these models tend to be less suitable when working with a limited number of evaluated points. This is due to the data set (i.e. evaluated points) being split at every decision node of the trees. With a limited number of evaluated points, the trees have to be shallow to prevent overfitting. Furthermore, Mockus (1994) showed that the GP prior distribution satisfies the conditions necessary for BO to converge to the optimum. Hence, this makes GP models suitable for use in BO. A brief introduction to GPs is provided in Section 3.2.1. For a comprehensive guide on GPs, we refer the reader to Williams and Rasmussen (2006).

Historically, GPs are more frequently referred to as Kriging in geostatistics. This name has carried over to the metamodel-based optimization literature, where the Kriging metamodel is commonly used to approximate the objective function (see e.g. Kleijnen (2017)). While known by different names, the Kriging metamodels and GPs are mathematically equivalent. For instance, ordinary Kriging is equivalent to a GP with a constant prior mean and deterministic observations (Kleijnen 2017), while stochastic Kriging (Ankenman, Nelson, and Staum 2010) is equivalent to a GP with a noise term and is used to model a stochastic function. The main difference between BO and Kriging metamodel-based optimization is that the latter directly optimizes the metamodel at each iteration (see e.g. Osorio and Bierlaire (2013)), while an acquisition function is derived from the GP and optimized in BO instead. The acquisition function helps to guide the search for the optimum, and is designed to systematically balance the exploration-exploitation trade-off. A more detailed explanation of the acquisition function is provided in Section 3.2.1.

The ability of BO, in combination with GPs, to balance between exploration and exploitation makes it an attractive method for tackling transportation problems. It has previously been employed to tackle problems such as simulation model calibration (Sha, Ozbay, and Ding 2020), OD matrix calibration (Schultz and Sokolov 2018) and toll optimization (Gu, Waller, and Saberi 2019, Chen et al. 2014). Despite the successes of BO, it is widely acknowledged that BO is mostly limited to low-dimensional
problems, typically less than 10 dimensions (Wang et al. 2016, Kandasamy, Schneider, and Póczos 2015). Scaling BO for use in higher-dimensional problems has been a major challenge in the field that remains unsolved. For instance, the transportation problems previously addressed with BO (Sha, Ozbay, and Ding 2020, Schultz and Sokolov 2018, Chen et al. 2014) are relatively low-dimensional. Since transportation problems can often be high-dimensional (e.g. greater than 100 dimensions), this limitation of BO has restricted its use in tackling transportation problems in general.

There are two key reasons for the poor performance at higher dimensions. First, there is the challenge of modeling the objective function at higher dimensions – the number of observations needed to get a good coverage of the feasible region increases exponentially with dimensions (Moriconi, Kumar, and Deisenroth 2020, Wang et al. 2016, Kandasamy, Schneider, and Póczos 2015). In the context of SO, this presents a significant practical challenge due to the high computational demand of obtaining observations through simulation. Second, there is the challenge of globally optimizing the acquisition function at every iteration (Rana et al. 2017, Kandasamy, Schneider, and Póczos 2015). The acquisition function often has many flat regions and can be highly multimodal, particularly in higher dimensions, making it tricky to find the optimum.

There have been many attempts to tackle high-dimensional BO. Most of the existing work assumes that the objective function mostly depends on a lower-dimensional “active" subspace (Munteanu, Nayebi, and Poloczek 2019, Schultz and Sokolov 2018, Wang et al. 2016, Chen, Castro, and Krause 2012). For instance, Wang et al. (2016) tackled a problem with a billion dimensions, by projecting the higher-dimensional space to a lower-dimensional subspace through random embedding. Li et al. (2017) also proposed a dropout strategy to optimize only a subset of variables at each iteration. The “active" subspace approach has also been used for transportation problems (Schultz and Sokolov 2018), illustrating its potential in reducing the dimensionality of the problem when no prior information is available. One disadvantage of using the “active" subspace approach is that additional simulation observations may be needed to identify the subspace. This may encroach on the already limited number
of available simulation evaluations when working with a tight computational budget.

Moriconi, Kumar, and Deisenroth (2020) showed that projecting the data onto a lower-dimensional subspace can lead to underfitting of the GP. Hence, they proposed the quantile GP for selecting on the best observation for each parameter subconfiguration when fitting the GP. The elastic GP (Rana et al. 2017) tries to overcome the problem of flatness in the acquisition function by making the covariance function length-scale large enough, so as to get some significant (i.e. nonzero) gradient values to aid the acquisition function optimization. The Add-GP-UCB model (Li et al. 2016) treats the acquisition function as an additive function of mutually exclusive lower-dimensional components, assuming that the acquisition function can be decomposed into an additive form. This allows the optimization of the acquisition function to be done in a lower-dimensional space.

A relevant branch of BO involves the use of multi-output GPs (Liu, Cai, and Ong 2018, Poloczek, Wang, and Frazier 2017, Kandasamy et al. 2016, Swersky, Snoek, and Adams 2013). It attempts to use correlation between the objective function and multiple low-fidelity, analytical or simulation-based, models, in order to inform the search for the optimum. For transportation SO problems, analytical transportation models can take the place of the low-fidelity model, while the objective function is provided by the computationally expensive simulator. The main aim of using multi-output GPs is to reduce the number of objective function evaluations, and hence computing time, needed to find the optimum. While multi-output GPs could potentially be used to provide better coverage of the feasible region through correlated observations of the low-fidelity models (see e.g. Swersky, Snoek, and Adams (2013)), the large number of observations required for a high-dimensional problem could lead to another computational bottleneck – computing the GP posterior involves a matrix inversion and typically scales as $O(t^3)$ (Williams and Rasmussen 2006, Chapter 8), where $t$ is the number of observations. For conventional (i.e. single-output) GPs, the $O(t^3)$ computational complexity applies when computing the GP posterior as well, but the number of observations tends to be smaller since the objective function is expensive to evaluate.
3.1.2 SO in Transportation

In the past, common approaches used to tackle transportation SO problems have focused more on exploitation techniques. Some of these approaches consisted of general-purpose algorithms, including genetic algorithms (Jin, Ma, and Kosonen 2017, Sebastiani, Lüders, and Fonseca 2016, Paz et al. 2015, Stevanovic et al. 2008, Teklu, Sunalee, and Watling 2007, Yun et al. 2006) and simultaneous perturbation stochastic approximation (SPSA) (Tympakianaki, Koutsopoulos, and Jenelius 2018, Lu et al. 2015, Tympakianaki, Koutsopoulos, and Jenelius 2015). While these general-purpose algorithms can easily be applied to different problem types, they are not designed to be used under a tight computational budget. Instead, they tend to be designed based on asymptotic properties. When used in high-dimensional SO, the large number of objective function evaluations (i.e. simulation runs) required means that they are rather computationally inefficient. This lack of computational efficiency of these general-purpose algorithms also stems from the fact that they treat the simulator as a black-box, exploiting little to no problem-specific information or problem structure.

A different approach to SO, that also focuses on exploitation, treats the simulator as a grey-box. It combines information from the simulator with problem-specific prior information in the form of analytical transportation models with the aim of identifying good solutions efficiently. Osorio and Bierlaire (2013) proposed a metamodel SO framework, where the main idea is to use an analytical surrogate model to approximate the simulation-based objective function to enhance the exploitation ability of the SO algorithm. The surrogate approximate is improved parametrically to yield what is known as a metamodel. This metamodel SO approach has been used to tackle various types of high-dimensional problems in the transportation field, including urban traffic signal optimization (Chong and Osorio 2018, Osorio and Chong 2015, Osorio and Nanduri 2015a,b), OD matrix calibration (Zhang, Osorio, and Flötteröd 2017), congestion pricing (Osorio and Atastoy forthcoming) and car-sharing network design (Zhou, Osorio, and Fields forthcoming). However, the metamodel SO approach does not explicitly try to balance exploration and exploitation. For in-
stance, a general-purpose sampling strategy (e.g. uniform random sampling) is often used for exploration. In trying to tackle this issue, the surrogate model was used to define a sampling distribution in Chapter 2. This sampling distribution assigns greater sampling probability to points with better predicted performance, as opposed to uniform sampling.

The methodology of this chapter follows a similar line of reasoning: it is based on the premise that SO algorithms can become more efficient for high-dimensional problems by using information from simple analytical surrogate models. However, unlike past work, this chapter considers the use of the surrogate model to jointly improve both exploitation and exploration.

### 3.1.3 Contributions

Existing methods for tackling transportation SO problems often place more emphasis on exploitation than exploration. On the other hand, BO achieves efficiency by balancing exploration and exploitation. However, BO faces significantly challenges when scaling to high-dimensional problems. To address these issues, we propose a method to incorporate problem-specific prior information in the BO framework using GPs. This is unlike past work in the field of transportation which employed the use of BO with standard non-informative priors (Sha, Ozbay, and Ding 2020, Schultz and Sokolov 2018, Chen et al. 2014).

In this chapter, we use a computationally efficient analytical transportation model as the problem-specific surrogate model. While it is clear that good prior information can enhance the performance of BO, it’s may not be that obvious as to where this prior information should be used. We formulate 2 approaches to incorporate information from the surrogate model. The first incorporates it in the prior mean function (which mainly governs exploitation) of the GP, the second does so in the covariance function (which mainly governs exploration) in a novel way. The proposed method therefore allows to use the surrogate model information for both exploration and exploitation. More importantly, the proposed paper analyses the added value of each approach and discusses when to choose one approach over the other. Using the surrogate model in
the prior mean function of the GP allows the algorithm to exploit the problem-specific prior information to quickly identify good solutions. On the other hand, the proposed covariance function is designed to encourage the algorithm to explore in regions with different surrogate model value from points which have already been evaluated.

The contributions of this chapter are summarized as follows:

- **Enable BO to tackle high-dimensional problems:** The proposed approaches of incorporating information from the surrogate model allows the BO algorithm to efficiently tackle high-dimensional optimization problems with a limited number of simulation runs, by exploiting the correlations between the objective function and surrogate model to encourage targeted exploration of the feasible region.

- **Use of problem-specific information for exploration and exploitation:** We propose 2 approaches for incorporating approximate problem-specific information from an analytical surrogate model in the BO framework. The first approach involves placing the surrogate model in the prior mean function of the GP, while the second approach uses the surrogate model in the GP covariance function. The proposed approaches allow for the problem-specific information to be used for both exploration and exploitation. This differs from previous transportation SO work which use the problem-specific information mainly for exploitation.

- **Evaluation of effect of bias in surrogate:** We evaluate the impact that different types of bias present in the surrogate can have on the optimization performance. Based on this result, we provide an understanding of when to use the surrogate model for exploitation or exploration as a function of the accuracy, correlation and type of bias of the surrogate.

- **Generality:** The method can be generalized to other classes of optimization problems with expensive-to-evaluate objective functions, as long as a suitable problem-specific surrogate model for approximating the objective function is available.
In the following section, we briefly present how GPs and BO work, followed by the proposed method used to combine problem-specific prior information in GPs, for use in BO. We then illustrate, in Section 3.3, how the method works in the case of the 1-dimensional Griewank function, followed by a validation of the method using the 100-dimensional Griewank function. The proposed method is tested in a case study using a model of Midtown Manhattan for a traffic signal optimization problem. The results of the case study are presented and discussed in Section 3.4. Lastly, the conclusions are provided in Section 3.5. Appendix B.1 summarizes the notation used in this chapter. Appendix B.2.3 provides implementation details for the validation experiments of Section 3.3. The implementation details for the case study of Section 3.4 are provided in Appendix B.3.

3.2 Method

We first provide a brief explanation of how GPs and BO work in Section 3.2.1. Readers who are familiar with BO and GPs may wish to skip to Section 3.2.2. In Section 3.2.2, we propose methods to incorporate problem-specific prior information into GPs through the prior mean function and the covariance function.

3.2.1 Bayesian Optimization

Bayesian optimization consists of two main components – a model of the objective function that can be updated with the observations at every iteration, as well as the acquisition function which decides on the next point to evaluate. In Section 3.2.1, we provide a brief introduction to one of the most popular objective function models for use in BO – Gaussian processes. This is followed by an explanation of the role of the acquisition function in BO in Section 3.2.1. A summary of the notation used in this section is provided in Appendix B.1.
Gaussian Process.

A GP represents a distribution over functions, and is specified by a prior mean function \( m(\cdot) \) and a covariance function \( k(\cdot, \cdot) \). Specifically, given a set of \( t \) points \( \mathbf{x}_{1:t} = [\mathbf{x}_1, \ldots, \mathbf{x}_t]^T \), where \( \mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, n \), with objective function estimates \( f_{1:t} \), where \( f_i = f(\mathbf{x}_i) \), the GP prior can be defined by the multivariate normal distribution:

\[
\mathbf{f}_{1:t} \sim \mathcal{N}(\mathbf{m}_{1:t}, \mathbf{K}(\mathbf{x}_{1:t}, \mathbf{x}_{1:t})) ,
\]

where \( \mathbf{m}_{1:t} \) is the vector of mean function values such that \( m_i = m(\mathbf{x}_i) \), and \( \mathbf{K}(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}) \) is the covariance matrix, with the \((i, j)^{th}\) element being \( k(\mathbf{x}_i, \mathbf{x}_j) \) (i.e. the covariance between \( \mathbf{x}_i \) and \( \mathbf{x}_j \)).

The choice of prior mean function and covariance function is up to the user. Simply speaking, the mean function in GPs defines the mean value of the normal distributed objective function estimate at a given point in the feasible region, while the covariance function defines the covariance between two points. Ideally, the prior mean function and the covariance function should be chosen such that they best represent the objective function. For instance, if the objective function contains a periodic component, the covariance function should ideally contain a periodic component too (e.g. see Chapter 5.4.3 of Williams and Rasmussen (2006)). In the case where the form of the objective function is completely unknown, a zero function or constant is often used as the prior mean function, while a popular choice of covariance function is the squared exponential function:

\[
k^{SE}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_0^2 \exp \left( -\frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{2\ell^2} \right) ,
\]

where the hyperparameters \( \sigma_0^2 \) and \( \ell \) are the amplitude and characteristic length-scale of the covariance respectively. The hyperparameter \( \ell \) determines the Euclidean distance between two points required for the two points to effectively be uncorrelated. In Section 3.2.2, we show how problem-specific information, when available,
can be incorporated in the prior mean function and the covariance function to aid the optimization of high-dimensional problems. For more information and examples of covariance functions, we refer the reader to Chapter 4 of Williams and Rasmussen (2006).

One of the factors contributing to the popularity of GPs is its analytical tractability. As seen from Eq. (3.2), the points \( \mathbf{x}_{1:t} \) are jointly Gaussian in the GP prior. Similarly, a new point \( \mathbf{x}_* \) would also be jointly Gaussian under the GP prior:

\[
\begin{bmatrix}
\mathbf{f}_{1:t} \\
\mathbf{f}(\mathbf{x}_*)
\end{bmatrix} \sim \mathcal{N}
\begin{bmatrix}
\mathbf{m}_{1:t} \\
\mathbf{m}(\mathbf{x}_*)
\end{bmatrix},
\begin{bmatrix}
K(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}) & K(\mathbf{x}_{1:t}, \mathbf{x}_*) \\
K(\mathbf{x}_*, \mathbf{x}_{1:t}) & K(\mathbf{x}_*, \mathbf{x}_*)
\end{bmatrix}
\]  

(3.4)

where \( K(\mathbf{x}_*, \mathbf{x}_{1:t}) = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_t)] \),

\[
k(\mathbf{x}_{1:t}, \mathbf{x}_*) = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_t)]^T.
\]

(3.5)

If the points \( \mathbf{x}_{1:t} \) and estimates \( \mathbf{f}_{1:t} \) are treated as previous observations (i.e. training points), it is possible to condition on them to limit the distribution of possible functions as predicted by the GP (i.e. fitting the GP prior to the observations). This is known as the GP posterior, and can be obtained analytically using the Sherman-Morrison-Woodbury formula (see e.g. Appendices A.2 and A.3 of Williams and Rasmussen (2006) for details on the derivation). In the case of noisy observations (assuming additive i.i.d. Gaussian noise with variance \( \tau^2 \)), the resulting predictive distribution given the \( t \) observations takes on a Gaussian distribution, with predictive mean value \( \mu_t(\mathbf{x}_*) \) and predictive variance \( \sigma_t^2(\mathbf{x}_*) \):

\[
f(\mathbf{x}_*) | \mathbf{f}_{1:t}, \mathbf{x}_1, \ldots, \mathbf{x}_t \sim \mathcal{N}(\mu_t(\mathbf{x}_*), \sigma_t^2(\mathbf{x}_*))
\]

(3.7)

\[
\mu_t(\mathbf{x}_*) = \mathbf{m}(\mathbf{x}_*) + \mathbf{k}(\mathbf{x}_*, \mathbf{x}_{1:t}) \left[ K(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}) + \tau^2 \mathbf{I} \right]^{-1} (\mathbf{f}_{1:t} - \mathbf{m}(\mathbf{x}_*))
\]

(3.8)

\[
\sigma_t^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathbf{x}_*, \mathbf{x}_{1:t}) \left[ K(\mathbf{x}_{1:t}, \mathbf{x}_{1:t}) + \tau^2 \mathbf{I} \right]^{-1} k(\mathbf{x}_{1:t}, \mathbf{x}_*)
\]

(3.9)

where \( \mathbf{I} \) is the identity matrix. Eq. (3.8) indicates that the posterior mean function \( \mu_t(\mathbf{x}_*) \) is fitted according to the difference between the observed objective function value and the prior mean function value (i.e. \( \mathbf{f}_{1:t} - \mathbf{m}(\mathbf{x}_*) \)). Eq. (3.9) shows
that the predictive variance \( \sigma_t^2(x_*) \) of the GP posterior is reduced in areas where there is strong covariance with existing observations (i.e. when the term \( k(x_*, x_{1:t}) [K(x_{1:t}, x_{1:t}) + \tau^2 I]^{-1} k(x_{1:t}, x_*) \) is large). Note that the computation of the matrix inversion \( [K(x_{1:t}, x_{1:t}) + \tau^2 I]^{-1} \) in Eq. (3.8) and (3.9) has a computational complexity of \( O(t^3) \), and may pose a challenge when there is a large number of observations (e.g. \( 10^5 \) observations). However, in the case of SO, the number of observations tends to be too small for this to be a problem.

A 1-D example of the GP posterior is illustrated in Figure 3-1. The two stars represent the observations which the GP is fitted to, giving rise to the posterior mean function and the predictive variance as depicted by the solid line and shaded region respectively. In this figure, the shaded region shows the values \( \pm \sigma \) away from the posterior mean function. The objective function is represented by the dashed line. The global minimum lies at \( x = 0 \), with local minima at around \( x = \pm 6.3 \). The GP posterior can be interpreted as follows: at any point \( x_* \) in the feasible region, the GP posterior predicts that the corresponding objective function value is normally distributed with mean \( \mu(x_*) \) and variance \( \sigma^2(x_*) \), as given by Eq. (3.8) and (3.9) (note that the subscript \( t \) are frequently omitted from \( \mu(x_*) \) and \( \sigma^2(x_*) \) for simplicity, and it can be assumed to include all available observations unless otherwise stated). In other words, for given point \( x_* \) in Figure 3-1, the GP posterior predicts the objective function value to be normally distributed about \( \mu(x_*) \), with the shaded region showing the values \( \pm \sigma \) away from the mean. The observations provide information about the objective function at those points, thus reducing the predictive variance of the GP around those points. However, as the objective function being modeled is noisy, there is still a non-zero variance around the observations. In regions far from the observations, the GP posterior mean function tends towards the prior mean function, which is zero in this example.

**Acquisition Function.**

Given the posterior mean function and the predictive variance of the GP posterior, the acquisition function is used to identify the next point to evaluate. The role of the
acquisition function is to balance exploitation and exploration when selecting the next evaluation point. In a minimization (resp. maximization) problem, selecting a point with a small (resp. large) posterior mean function value corresponds to exploitation, while selecting a point with a large predictive variance represents exploration. The acquisition function, therefore, is a function of both $\mu(x^\ast)$ and $\sigma^2(x^\ast)$. The next point to evaluate is then chosen by maximizing the acquisition function.

Several acquisition functions with analytical expressions have been proposed in the BO literature, including the probability of improvement (Kushner 1964), expected improvement (Jones, Schonlau, and Welch 1998, Mockus, Tiesis, and Zilinskas 1978) and upper confidence bound (Srinivas et al. 2009). In this work, we work with the expected improvement (EI) acquisition function, which is defined as follows (for a minimization problem):

$$EI(x^\ast) = \mathbb{E} \left[ \max \{0, f_{\text{min}} - f(x^\ast)\} \right]$$

$$= (f_{\text{min}} - \mu(x^\ast))\Phi(Z) + \sigma(x^\ast)\phi(Z),$$  \hspace{1cm} \text{(3.11)}

$$\text{where } Z = \frac{f_{\text{min}} - \mu(x^\ast)}{\sigma(x^\ast)}$$

$$\text{(3.12)}$$

where $f_{\text{min}}$ denotes the smallest objective function estimate of all the observations; $\phi(\cdot)$ and $\Phi(\cdot)$ in Eq. (3.11) denote the probability density function (pdf) and cumu-
lative distribution function (cdf) of the standard normal distribution respectively. In Eq. (3.11), the first component can be interpreted as the exploitation component, as it is proportional to the possible improvement in objective function value $f_{\text{min}} - \mu(x^*)$. The second component can then be interpreted as the exploration component, as it is proportional to the predictive variance. To select the next point to evaluate $x_{t+1}$, we solve the following maximization problem:

$$x_{t+1} = \arg \max_{x} EI(x).$$

(3.13)

Implementation details for solving Eq. (3.13) are provided in Appendix B.2.1.

EI is chosen as the acquisition function as it is more commonly used than probability of improvements, and also it does not require additional parameters, unlike the upper confidence bound acquisition function (Snoek, Larochelle, and Adams 2012). When working with noisy observations, the EI acquisition function may not be properly defined since the true distribution of $f$ is not known (see e.g. Section 5.1 of Frazier (2018)). There exist alternative acquisition functions that may be more suitable for noisy observations, such as the knowledge gradient function (Frazier, Powell, and Dayanik 2009, Wu and Frazier 2016), the entropy search function (Hennig and Schuler 2012) and the predictive entropy search function (Hernández-Lobato, Hoff- man, and Ghahramani 2014). However, these alternative acquisition functions often require Monte Carlo methods to evaluate. This makes them computationally less efficient compared to the EI function.

The EI acquisition function is also depicted in the 1-D example in Figure 3-1 by the dotted line. As can be seen in the figure, the EI acquisition function has the largest values where there is a combination of a small posterior mean function value and a large predictive variance. In particular, the largest EI values lie in the region $x < 0$ in this iteration, where there are currently no observations, indicating that the algorithm should explore in this region. Furthermore, the EI value is zero around $x = 3.5$, since the GP posterior predicts that a minimum is very unlikely to lie in that area. The predictive variance around $x = 3.5$ is also close to zero.
3.2.2 Proposed Approach: Gaussian Process with Problem-Specific Information

For some optimization problems, we may have access to problem-specific prior information or have some prior beliefs about the shape of the objective function. For instance, in transportation problems involving an urban road network, the underlying road network topology would be known beforehand. Hence, it may be possible to derive a surrogate model $f^A(\cdot)$ that approximates or correlates with the objective function. In this section, we show how information from $f^A$ can be incorporated in the GP prior mean function (Section 3.2.2) and the covariance function (Section 3.2.2) for exploitation and exploration, in order to enable efficient high-dimensional BO.

Prior Mean Function.

As mentioned in Section 3.2.1, a constant prior mean is typically chosen when there is no available prior information about the objective function:

$$m(x) = \beta,$$  \hspace{1cm} (3.14)

where $\beta$ is a constant. In the case of a zero prior mean function, $\beta$ is taken to be 0.

However, given a surrogate model $f^A(\cdot)$ that approximates the objective function, it would be natural to use it in the prior mean function:

$$m(x) = \alpha f^A(x),$$  \hspace{1cm} (3.15)

where $\alpha$ is a scaling constant that can be fitted according to the data. Assuming $f^A$ correlates well with the objective function, it can inform the acquisition function about the locations of possible optima through the posterior mean function as shown in Eq. (3.8). This thus allows BO to exploit the problem-specific prior information to efficiently identify good solutions even at higher dimensions. This is illustrated using a 1-D example in Section 3.3.3.
Covariance Function.

The choice of the covariance function can give rise to different GP posteriors. As shown by Eq. (3.8) and (3.9), the choice of the covariance function affects both the posterior mean function and the predictive variance. The covariance between two points, as defined by the covariance function, represents how correlated the two points are believed to be. Standard covariance functions, such as the squared exponential covariance function (Eq. (3.3)), work with the assumption that points close to one another have similar objective function values (i.e. the objective function is smooth). In the case of the squared exponential covariance function, the exponential term ensures that $k(x_i, x_j) \to \sigma_0^2$ when $x_i$ and $x_j$ are close to each other in Euclidean distance, and $k(x_i, x_j) \to 0$ when the Euclidean distance between $x_i$ and $x_j$ becomes large. In fact, this is a necessary condition for BO to converge to the optimum (Mockus 1994).

However, other than assuming a smooth objective function, the standard covariance functions do not exploit any problem structure at all. Here, we propose a covariance function that is able to incorporate problem-specific prior information in the form of a surrogate model $f^A$ to efficiently tackle high-dimensional BO problems:

$$k^A(x_i, x_j) = \sigma_0^2 \exp \left( -\frac{\|x_i - x_j\|^2}{2\ell^2} \right) \exp \left[ -\frac{(f^A(x_i) - f^A(x_j))^2}{2(\ell^A)^2} \right], \quad (3.16)$$

where $\ell^A$ is the surrogate model length-scale. The surrogate model-based covariance function $k^A$ differs from the standard squared exponential covariance function $k^{SE}$ in Eq. (3.3) in the addition of a second exponential component. This additional exponential component uses the difference between the $f^A$ values of the two points to further attenuate the covariance between the two points. Put differently, the difference in $f^A$ values is used as an additional distance measure, which incorporates the problem-specific prior information, to determine how ated the two points are. This means that for a given Euclidean distance, we are incorporating the prior belief that two points with similar $f^A$ values have high covariance and vice versa. At the same time, the first exponential component in Eq. (3.16) ensures, that when
the Euclidean distance between the two points is large, the covariance between the
two points goes to zero, reflecting the uncertainty of making predictions in regions
without any observations. The hyperparameter $\ell^A$ effectively determines how big of
a difference in $f^A$ values is required for the two points to become uncorrelated.

The squared exponential covariance function forms the basis for $k^A$ in Eq. (3.16),
allowing us to directly compare the performance of $k^A$ and $k^{SE}$ to determine the
added value of using the surrogate model in the covariance function. However, we
would like to emphasize that the surrogate model component can be added to other
covariance functions, such as the Matérn covariance functions. Hence, the proposed
method of incorporating problem-specific information in the covariance function is
highly generalizable.

The proposed covariance function $k^A$ is able to encourage exploration of parts of
the feasible region with different surrogate model values than previously evaluated
points (i.e. observations). Since $f^A$ just approximates or correlates with the objective
function $f$, the optimum of $f$ may not coincide with the optimum of $f^A$. Hence,
encouraging explorations in areas with different $f^A$ values can increase the chances
of finding the optimum of the objective function.

As seen in Eq. (3.9), the predictive variance is reduced in regions with non-zero
covariance to previously evaluated points. Using $k^{SE}$ as the covariance function means
that only the regions with nearby observations in the Euclidean distance sense have
reduced posterior variance. However, using $k^A$ as the covariance function allows for
points that are further away from observations to have greater covariance if they
have similar $f^A$ values as the previously evaluated points, thus resulting in a smaller
predictive variance. Hence, with the predictive variance being attenuated based on
the difference in $f^A$ values compared to those of the previously evaluated points, there
is a larger predictive variance in regions with $f^A$ values that are different from those of
the evaluated points. Since the acquisition function (Eq. (3.11)) assigns greater values
to regions with high predictive variance, exploration in these regions are encouraged.
The effect of using $k^A$ as the covariance function is illustrated using a 1-D problem
in Section 3.3.3.
The effect of using $k^A$ is particularly significant in higher dimensional problems where there is a limited number of observations. The limited number of observations results in a poor coverage of the high-dimensional feasible region. This makes it difficult to model the objective function well if using a GP prior with no problem-specific prior information. However, by using $k^A$ as the covariance function, both the posterior mean function and the predictive variance will have access to $f^A$ (see Eq. (3.8) and (3.9)). This helps to tackle the two problems which plague conventional high-dimensional BO. First, while the limited number of observations provides a poor coverage of the high-dimensional feasible region, it provides a decent coverage of the 1-D space of $f^A$ difference. In regions with no nearby observations by Euclidean distance, there could still be significant covariance between points with similar $f^A$ values as the previously evaluated points. This helps to inform the GP on possible objective function values in the unexplored regions, based on similarities in $f^A$ values. Second, when using $k^A$ as the covariance function, the posterior mean function and the predictive variance are informed on possible objective function values in unexplored regions. This means the posterior mean function and the predictive variance are less likely to be constants in the unexplored regions. As a result, the acquisition function would have fewer flat regions, allowing it to be more easily optimized using a gradient ascent approach. Hence, this can allow BO to be effective in high-dimensional problems. Section 3.3.3 provides an illustration of how the use of $k^A$ as the covariance function can help BO in high dimensional spaces.

Algorithm.

The BO algorithm is summarized in Algorithm 3. The GP hyperparameters are first initialized. A set of random initial points in the feasible region are then sampled and evaluated, providing the initial observations for fitting the GP posterior. At every iteration, the GP hyperparameters are fitted to the observations through maximum likelihood estimation (for details, see Section 5.4.1 of Williams and Rasmussen (2006)). Then, the acquisition function, which is based on the GP posterior (see Eq. (3.10)-(3.12)), is maximized to select the next point for evaluation, as described
Algorithm 3: Bayesian optimization

1. Initialization
   
   (a) Choose a prior mean function and a covariance function
   
   (b) Initialize GP hyperparameters: $\alpha$, $\beta$, $\sigma_0^2$, $\ell$, $\ell^A$ (depending on chosen prior mean function and covariance function)
   
   (c) Randomly sample $t_0$ points $\mathbf{x}_{1:t_0}$ from the feasible region and evaluate through simulation to obtain $f_{1:t_0}$

2. Optimization
   
   for $t = t_0, \ldots, T - 1$
   
   (a) Update GP hyperparameters through maximum likelihood estimation based on the data $\{\mathbf{x}_{1:t}, f_{1:t}\}$ (see e.g. Section 5.4.1 of Williams and Rasmussen (2006), and Appendix B.2.2)
   
   (b) Fit the GP to the data $\{\mathbf{x}_{1:t}, f_{1:t}\}$ to obtain the posterior mean function $\mu_t(\mathbf{x}_*)$ (Eq. (3.8)) and the predictive variance $\sigma_t^2(\mathbf{x}_*)$ (Eq. (3.9))
   
   (c) Identify the next point to evaluate $\mathbf{x}_{t+1}$ by maximizing the acquisition function (Eq. (3.13))
   
   (d) Evaluate the point through simulation to obtain $f_{t+1}$

in Section 3.2.1. This is repeated until the optimization budget of $T$ iterations is exceeded. Details on how the GPs are implemented are provided in Appendix B.2.2.

The main difference in the algorithm, compared to a standard BO approach, when incorporating problem-specific information in the GP is the set of hyperparameters required (see Step 1(b) of Algorithm 3). If the surrogate model is used in the prior mean function (Eq. (3.15)), the prior mean function hyperparameter required is $\alpha$. Otherwise, if a constant prior mean function is used (Eq. (3.14)), the hyperparameter required is $\beta$. Similarly, if $k^{SE}$ is used as the covariance function, the only covariance function hyperparameters required are $\sigma_0$ and $\ell$. If $k^A$ is the chosen covariance function, the additional hyperparameter $\ell^A$ is required as well. Having initialized the right set of hyperparameters, the rest of the algorithm is the same regardless of choice of prior mean function and covariance function.

It should be mentioned that the use of a surrogate model in the GP involves a computational trade-off. We have added problem-specific information through the
analytical surrogate model, but this comes at the cost of evaluating and optimizing functions that depend on $f^A$. Hence, it’s use in the prior mean function (Eq. (3.15)) (resp. the covariance function (Eq. (3.16))) could lead to greater computational runtimes compared to using the constant prior mean function (Eq. (3.14)) (resp. the standard squared exponential covariance (Eq. (3.3))). The increase in computational runtime also depends on the surrogate model being used. Hence, the chosen model should ideally have low compute times.

### 3.3 Validation and Illustration

In this section, we validate the proposed method of incorporating problem-specific prior information in the prior mean function and the covariance function of the GP works using the Griewank function as the objective function. We first define the Griewank function and introduce the optimization problem in Section 3.3.1. Then, we present the different benchmark methods in Section 3.3.2. In Section 3.3.3, we validate the proposed method using the 1-D Griewank function. Following that, we show that the proposed method is able to tackle high-dimensional BO using a 100-D Griewank function for illustration in Section 3.3.4. In Section 3.3.5, we illustrate the effects that biases in the analytical model can have on the optimization performance.

#### 3.3.1 Griewank Function

The $D$-dimensional Griewank function (Griewank 1981) is defined as follows:

$$g(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right), \quad (3.17)$$

where $x = [x_1, \ldots, x_D]^T$. The Griewank function is chosen, as it is a continuous and non-convex function with multiple local minima – properties which are present in many objective functions of optimization problems in transportation. Furthermore, the Griewank function is easily generalizable to any number of dimensions, allowing us to test the proposed method on problems of any dimension.
To make it a stochastic problem, we add i.i.d. Gaussian noise $\tau$ with mean 0 and variance 0.01 to the Griewank function $g(x)$ (Eq. (3.18)). In addition, we set the feasible region as $[-10,10]^D$ (Eq. (3.20)). The noisy $D$-dimensional Griewank function optimization problem that we consider in this section is summarized by Eq. (3.18)-(3.20):

$$\min_x f(x) = g(x) + \tau,$$

subject to $\tau \sim \mathcal{N}(0,0.01)$,

$$x_i \in [-10,10] \quad \forall i = 1,\ldots,D.$$  

(3.18)  
(3.19)  
(3.20)

### 3.3.2 Benchmark Methods

In this problem, we assume there is prior knowledge that the objective function (Eq. (3.18)) has a quadratic component. Based on this, we choose a quadratic model as the analytical model:

$$f^A(x) = \|x\|_2^2.$$  

(3.21)

Note that $f^A$ in Eq. (3.21) is not an accurate model of the Griewank function, in the sense that it does not reflect the undulations and local minima. However, it still has significant correlation with the Griewank function, by following the same general trend. In fact, the minimum of $f^A$ is perfectly aligned with the global minimum of the Griewank function at $x = 0$.

To evaluate our proposed method, we make use of 4 different GP priors for the Griewank function optimization problems, as well as the case study in Section 3.4. The 4 different GP priors are summarized in Table 3.1, and they differ according to whether (i) the prior mean function and (ii) the covariance function are based on the analytical model $f^A$ or not. The first column of Table 3.1 defines the names of the 4 GP priors used. The second column indicates whether the prior mean function is based on the analytical model as defined in Eq. (3.15) and (3.21). If not, the constant prior mean in Eq. (3.14) is used. The third column indicates whether the covariance
Table 3.1: GP Priors

<table>
<thead>
<tr>
<th>Name</th>
<th>Prior Mean Function Based on analytical model?</th>
<th>Covariance Function Based on analytical model?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed-Covariance</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Proposed-Mean</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Proposed-Combined</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

function is based on the analytical model as shown in Eq. (3.16) and (3.21). If it is not, the standard squared exponential covariance function given in Eq. (3.3) is used.

The Standard GP prior represents a general-purpose GP prior, which does not use any problem-specific information, and is taken as a benchmark for the other 3 proposed GP priors that make use of problem-specific information in the prior mean function and/or the covariance function. The comparison between Standard and Proposed-Covariance (resp. Proposed-Mean) allows us to evaluate the effect of using a problem-specific covariance function (resp. prior mean function). Comparing Proposed-Combined with Proposed-Covariance (resp. Proposed-Mean) serves to evaluate the added value of using the problem-specific information in the prior mean function (resp. covariance function).

### 3.3.3 1-D Griewank Function

Using the 1-D Griewank function, we first illustrate how the choice of GP prior affects the posterior when fitted to initial observations. For this example, 2 initial observations (Step 1c of Algorithm 3) are provided for fitting the GP posteriors, with the same pair of initial observations used across all 4 GP priors. For each observation, the objective function estimate is obtained by taking the mean of 4 simulations (i.e. 4 random draws of the noisy Griewank function in Eq. (3.18)). The fitting of the GP posterior is done by updating the GP hyperparameters through maximum likelihood estimation (see Step 2a of Algorithm 3).

Figure 3-2 plots the 4 different GP posteriors after fitting to the initial observations, along with the objective function and their respective EI acquisition functions.
Figure 3-2: GP posteriors fitted to 2 initial observations for the 1-D Griewank function.

In each plot, the x-axis represents the feasible region for this 1-D case, while the left and right y-axes represent the objective function value and EI value respectively. The initial observations are depicted by the two markers, with the marker styles and colors differing based on the GP prior. The marker styles and colors are consistent with those used in subsequent figures comparing the different GP priors. The posterior mean function is represented by the red solid line, while the shaded region represents the predictive variance and shows the values ±σ away from the posterior mean function. The objective function is depicted by the blue dashed line, with the global minimum occurring at \( x = 0 \) along with two other local minima in this 1-D case. The EI acquisition function for this iteration is shown by the orange dotted line.

The GP posterior for Standard is shown in Figure 3-2a. It can be observed that in the regions close to the observations, the posterior mean function models the
objective function relatively well, while reducing the predictive variance around the observations. Correspondingly, the EI function is assigned a high value where there is a combination of a small posterior mean function value and large predictive variance, and vice versa. However, in the region with no observations (i.e. \( x < 0 \)), we see that the posterior mean function simply tends towards the constant prior mean function, while the predictive variance goes towards \( \sigma_0^2 \). Since the posterior mean function and the predictive variance are essentially constant in the region \( x < 0 \), the EI function is also flat. As mentioned in Section 3.1.1, the acquisition function can have many flat regions. This is especially true in higher dimensions. Hence, Figure 3-2a shows why it can be difficult to find the global maximum of the acquisition function for high-dimensional problems.

In the case of Proposed-Covariance (Figure 3-2b), the GP posterior mean function and the predictive variance are no longer constant in the region \( x < 0 \) despite the lack of observations, unlike Standard. In fact, the GP posterior predicts similar objective function values for points with similar \( f_A \) values, which can be seen by noting that the posterior mean function is almost symmetric about \( x = 0 \) and that \( f_A \) is a symmetric function about \( x = 0 \). Note that even though Proposed-Covariance does not use an informative prior mean function, its posterior mean function is affected by \( k_A \) (see Eq. (3.8)). Hence, this explains the symmetry in the posterior mean function. In addition, the predictive variance around points with similar \( f_A \) values as the previously evaluated points are also reduced as expected. As a result of the behavior of the posterior mean function and the predictive variance in the region without observations \( (x < 0) \), the EI function is no longer flat, which is important when using a gradient-ascent approach when maximizing the acquisition function.

Proposed-Mean (Figure 3-2c) is similar to Standard, in that the posterior mean function tends towards the prior mean function in the region \( x < 0 \) which has no observations. The only difference is that the prior mean function in Proposed-Mean is \( m(x) = \alpha x^2 \), thus explaining why the posterior mean function increases quadratically as \( x \) becomes more negative in \( x < 0 \). At the same time, the predictive variance goes towards \( \sigma_0^2 \), since the covariance function in Proposed-Mean does not exploit
any problem-specific information. As a result of the non-constant posterior mean function in the region \( x < 0 \), the EI function is also no longer flat, unlike the case for Standard.

The GP posterior of Proposed-Combined (Figure 3-2d) is similar to that of Proposed-Covariance, in that similar objective function values are predicted for points with similar \( f^A \), due to the use of the problem-specific covariance function. The predictive variance is also reduced for points with similar \( f^A \) values as the previously evaluated points. The main difference between the GP posteriors of Proposed-Combined and Proposed-Covariance is that the posterior mean function of Proposed-Combined has an additional quadratic component, leading to larger \( \mu \) values for points further away from \( x = 0 \) as compared to that of Proposed-Covariance. This difference is also reflected in the EI function, where the peaks at around \( x = \pm 6 \) and \( x = \pm 10 \) for Proposed-Combined are smaller than those in Proposed-Covariance. In fact, the EI peak near \( x = \pm 10 \) is smaller than the one near \( x = \pm 6 \) for Proposed-Combined, unlike Proposed-Covariance. This shows how \( f^A \) in the prior mean function can bias the posterior mean function, thus affecting the EI function. If \( f^A \) is a good approximation of the objective function, this could allow the optimization problem to be solved more quickly.

The illustrations of the GP posteriors for the 4 different priors in Figure 3-2 shows how problem-specific information in the form of an analytical model can help when used in the prior mean function and/or the covariance function, by making the EI function less flat. In particular, the use of \( k^A \) as the covariance function in Proposed-Covariance and Proposed-Combined allows the GP model to more efficiently model the objective function using fewer observations, by taking advantage of the correlation between \( f^A \) and the objective function. As a consequence, the \( k^A \) covariance function encourages the exploration of points with different \( f^A \) values from those of the previously evaluated points to further improve the model of the objective function. Hence, this is how the \( k^A \) covariance function can make high-dimensional BO more efficient.

Furthermore, it should also be noted that the quadratic component in the objective
function is relatively weak compared to the sinusoidal component – the weight of the cosine term is much greater than the quadratic term (see Eq. (3.17)). The quadratic model $f^A$ captures only the quadratic trend, but does not accurately model the sinusoidal component of the objective function. However, as seen in Figures 3-2b and 3-2d, the $k^A$ covariance function still allows the correlation between $f^A$ and the objective function to be exploited when modeling the objective function. In particular, it infers that the objective function is symmetric about $x = 0$ based on $f^A$. Hence, this suggests that models that capture a general trend in the objective function are sufficient. They do not need to accurately model the objective function. This opens the door for a variety of models to be used as $f^A$ in other problems.

Next, we look at the optimization performance using the 4 different GP priors. After obtaining the 2 initial observations, we continue with the Optimization step (Step 2) of Algorithm 3. At every iteration, a new point is identified by maximizing the EI acquisition function, and the objective function estimate of that point is taken as the mean of 4 simulations. In addition, the best solution at the end of each iteration is identified and simulated 2 more times, and the objective function estimate is updated as the mean of all the simulations for that solution up to that iteration. This is so as to account for the noise and obtain a more accurate objective function estimate of the current best solution. The algorithm stops when the computational budget is reached. For this example, the computational budget is taken to be 30 iterations in total, including the 2 initial observations (i.e. 28 optimization iterations).

Figure 3-3 shows the performance of BO with the 4 different GP priors. The x-axis shows the optimization iteration number, while the y-axis shows the objective function estimate of the best solution at a given iteration. The optimization algorithm is run 3 times for each GP prior. Each line in Figure 3-3 depicts the mean of the best objective function estimate of the 3 runs, while the shaded regions represent the values $\pm 1$ standard error away from the mean. Standard corresponds to the black dash-dot line, Proposed-Covariance to the blue dashed line, Proposed-Mean to the magenta dotted line, and Proposed-Combined to the green solid line. In this plot, the sooner the curve reaches a $y$-value of 0, the more efficient the GP prior is for
From Figure 3-3, we observe that all 4 GP priors are able to find a solution with objective function value close to the global minimum (i.e. 0), as expected for a 1-D problem. Hence, this shows that the 4 GP priors work at low dimensions. Furthermore, Figure 3-3 also shows that the GP priors that make use of problem-specific information in the prior mean function and/or the covariance function (i.e. Proposed-Covariance, Proposed-Mean and Proposed-Combined) are able to identify a solution with objective function value close to the global minimum more quickly than Standard. Hence, even for a 1-D problem, having access to problem-specific information makes BO more efficient.

### 3.3.4 100-D Griewank Function

We next consider the 100-D noisy Griewank function as the objective function. This can be considered a high-dimensional problem in the context of BO, which has typically been limited to problems of less than 10 dimensions (Wang et al. 2016, Kandasamy, Schneider, and Póczos 2015).

For this 100-D problem, Algorithm 3 is implemented with 10 initial observations provided for fitting the GP posteriors, and a computational budget of 400 iterations.
The BO performance of the 4 different GP priors are illustrated in Figure 3-4. Each line in Figure 3-4 depicts the mean of the best objective function estimate of the 3 runs, while the shaded regions represent the values ±1 standard error away from the mean. As can be seen from Figure 3-4, the best solution identified by Standard has an objective function value of about 1.3, and it is unable to find a solution close to the minimum within the limited computational budget. This is expected given the high-dimensional nature of the problem, and no problem-specific prior information is provided in the prior for Standard. Proposed-Mean outperformed the solutions of Standard, but is unable to find a solution close to the global minimum. On the other hand, Proposed-Covariance and Proposed-Combined, which make use of the $k^A$ as the covariance function, are able to consistently find a solution close to the minimum of 0 within 75 iterations. Hence, this shows the effectiveness of using the problem-specific information in the covariance function for BO in a high-dimensional setting. In addition, Proposed-Combined is able to find a solution close to the minimum faster than Proposed-Covariance, suggesting that the additional

(i.e. 390 optimization iterations). The set of 10 initial observations are sampled uniformly at random from the feasible region, with the same set of 10 observations used across the 4 different GP priors. Similar to the 1-D problem, for each observation, the objective function estimate is obtained by taking the mean of 4 simulations (i.e. 4 random draws of the noisy Griewank function in Eq. (3.18)). In addition, at every iteration, the best solution at the end of each optimization iteration is identified and simulated 2 more times, and the objective function estimate is updated as the mean of all the simulations for that solution up to that iteration. However, instead of updating the GP hyperparameters at every iteration (Step 2a of Algorithm 3), the hyperparameters are kept fixed throughout the optimization run as it is observed that this resulted in better optimization performance (see Appendix B.2.3 for details on hyperparameter selection). The poorer optimization performance when the GP hyperparameters are updated is likely due to the maximum likelihood estimation having trouble finding a good set of hyperparameter values, resulting in large fluctuations in the estimated hyperparameter values at each iteration.

The BO performance of the 4 different GP priors are illustrated in Figure 3-4. Each line in Figure 3-4 depicts the mean of the best objective function estimate of the 3 runs, while the shaded regions represent the values ±1 standard error away from the mean. As can be seen from Figure 3-4, the best solution identified by Standard has an objective function value of about 1.3, and it is unable to find a solution close to the minimum within the limited computational budget. This is expected given the high-dimensional nature of the problem, and no problem-specific prior information is provided in the prior for Standard. Proposed-Mean outperformed the solutions of Standard, but is unable to find a solution close to the global minimum. On the other hand, Proposed-Covariance and Proposed-Combined, which make use of the $k^A$ as the covariance function, are able to consistently find a solution close to the minimum of 0 within 75 iterations. Hence, this shows the effectiveness of using the problem-specific information in the covariance function for BO in a high-dimensional setting. In addition, Proposed-Combined is able to find a solution close to the minimum faster than Proposed-Covariance, suggesting that the additional
access to the problem-specific information in its prior mean function does help with exploitation, hence making the optimization more efficient.

The effectiveness of the $k^A$ covariance function in finding solutions with objective function values close to the global minimum of 0 can be attributed to its ability to get the BO algorithm to explore the $f^A$-space (i.e. evaluate points with different $f^A$ values from previously evaluated points). As illustrated by the 1-D Griewank function example, using the $k^A$ covariance function (Figures 3-2b and 3-2d) results in the GP posterior predicting similar objective function values for points with similar $f^A$ values. At the same time, the predictive variance is also reduced for points with similar $f^A$ values as the previously evaluated points. As a result, points with different $f^A$ values from the previously evaluated points have greater predictive variances, which naturally encourages exploration for these points.

The amount of exploration in the $f^A$-space done by the 4 different GP methods, along with the corresponding objective function values $f$, is shown in Figure 3-5. Each row of plots represent one of the 4 different GP priors. The left column of Figure 3-5 consists of 2-D histograms illustrating the distribution of $f^A$ values (x-axis) and the corresponding $f$ values (y-axis) of the evaluated points. Each histogram shows the distributions for the different GP priors. The histograms are plotted using data aggregated over 3 algorithm runs (i.e. $400 \times 3 = 1200$ observations). In each histogram...
plot, the color of the bin represents the number of observations that fall within that bin, with a lighter color meaning a greater number of observations. Note that the scale of the color bar differs for each GP prior.

The right column of Figure 3-5 plots the $f^A$ values (x-axis) and the corresponding $f$ values (y-axis) for every observation, aggregated over 3 algorithm runs. The contour lines further represent the 2-D empirical cumulative distribution function (ecdf) of $f^A$ and $f$ based on the observations, which is defined as

$$
ecdf(f^A, f) = \frac{1}{N} \times |\{(f^A_i, f_i) : f^A_i \leq f^A, f_i \leq f\}|, \tag{3.22}
$$

where $N$ is the total number of observations (i.e. 1200), and $|\cdot|$ denotes the cardinality of a set. In words, the ecdf, as defined in Eq. (3.22), at a given point $(y_1, y_2)$ is the fraction of observations that have $f^A$ and $f$ values less than or equal to $y_1$ and $y_2$ respectively. In the plots, the contour lines denote the 2-D ecdf in steps of 0.1 (i.e. going from one contour line to the next in the increasing contour direction indicates a gain in ecdf of 0.1 and vice versa). In other words, the space between 2 adjacent contour lines contains 10% of the total number of observations. Hence, the 2-D ecdf contour lines help to better visualize the distribution of observations in the plots. The histogram (left column) and ecdf (right column) of Figure 3-5 present the same 2-D distribution of observations in different ways. However, the ecdf does it without having to discretize (i.e. bin) the $f^A$ and $f$ values. Hence, subsequent results (Figures 3-7 and 3-11) will be presented using the 2-D ecdf.

From Figure 3-5, it can be seen that the observations of Proposed-Covariance and Proposed-Combined (Figures 3-5d and 3-5h respectively) cover a much larger range of $f^A$ values than those of Standard and Proposed-Mean (Figures 3-5b and 3-5f respectively), showing that the use of the $k^A$ covariance function does indeed encourage exploration in the $f^A$-space. Note that the $f^A$ can also be viewed as a 1-D projection of $x$. Hence, exploration in the $f^A$-space can be used as a proxy to visualize exploration in the high-dimensional $x$-space.

Furthermore, since the minimum of $f^A$ coincides with the minimum of $f$ and
Figure 3-5: Histograms and 2-D empirical cumulative distribution functions illustrating exploration in $f^A$-space and the corresponding objective function values.
the $f^A$ value at the minimum is unique in this problem, Proposed-Covariance and Proposed-Combined can easily find solutions with $f$ close to 0 when exploring in regions with $f^A$ values close to 0, as shown by the observations in the lower left corner of Figures 3-5d and 3-5h. On the other hand, Standard and Proposed-Mean, which do not use the $k^A$ covariance function, do not even explore regions with $f^A$ values below 200. This can reduce their chances of finding solutions with $f$ values close to 0. Therefore, Figure 3-5 helps to show how the $k^A$ helps to tackle high-dimensional problems.

3.3.5 100-D Griewank Function with New Types of Biases

In choosing the analytical model $f^A$ for the Griewank function problem, we picked a quadratic model (Eq. (3.21)) such that the minimum of the quadratic model lies at the same position as the global minimum of the Griewank function (i.e. $x = 0$), which resulted in very efficient optimization for Proposed-Covariance and Proposed-Combined. However, in real-life applications, it is highly unlikely that the chosen analytical model would have an optimum that perfectly aligns with the optimum of the objective function. Here, we further consider scenarios where there are different types of biases in $f^A$. The different types of biases in $f^A$ that we consider are:

- **Inverted** (i.e. $f^A(x) \leftarrow -f^A(x)$): The analytical model is inverted so that it is a completely inaccurate model of the objective function. Instead, the inverted model is now anti-correlated with the objective function, with its maximum coinciding with the objective function minimum at $x = 0$. This illustrates the case where the analytical model is inversely biased in modeling the objective function, but is still able to provide information through (anti-)correlation.

- **Shifted** (i.e. $f^A(x) \leftarrow f^A(x - 1)$): The analytical model is shifted in the $x$-space, so that the minimum of the model lies at $x = 1$ instead. This illustrates the case where the minimum of $f^A$ and the global minimum of the objective function no longer coincide, leading to a reduction in correlation between $f^A$ and the objective function.
- **Shifted-Inverted** (i.e. \( f^A(x) \leftarrow -f^A(x - 1) \)): The analytical model is first shifted in the \( x \)-space, before being inverted. This means that the maximum of \( f^A \) lies at \( x = 1 \), and does not coincide with the global minimum of the objective function. While \( f^A \) still has some negative correlation with the objective function, the anti-correlation would not be as strong as that of the Inverted model, illustrating the case where \( f^A \) has a combination of the two biases above.

The optimization performance of the 4 different GP priors using the analytical models with the different biases are plotted in Figure 3-6. Each plot represents one of the 4 different GP priors. In each plot, each curve represents the mean of the best objective function estimate of 3 optimization run, with the respective shaded region depicting \( \pm 1 \) standard error away from the mean. The black solid line represents the Original \( f^A \) as given by Eq. (3.21), the blue dashed line shows the Inverted \( f^A \), the magenta dotted line indicates the Shifted \( f^A \), and the green dash-dot line corresponds to the Shifted-Inverted \( f^A \).

Standard does not make use of \( f^A \) in the prior, hence the 4 curves in Figure 3-6a correspond to the optimization performance of the exact same GP prior, which should result in similar performance, yielding an average best objective function value of around 1.3 at the end of 390 optimization iterations. Any differences in the 4 curves are due to the random initial points used in the multistart gradient ascent routine when maximizing the acquisition function at every iteration.

Proposed-Covariance makes use of \( f^A \) in the covariance function. As can be seen from Figure 3-6b, Proposed-Covariance is still able to find a solution with objective function close to 0 in around 90 iterations when using the Inverted \( f^A \), although it required more iterations on average than when using the Original \( f^A \) (around 50 iterations). This shows that Proposed-Covariance can be less sensitive to inversion bias in \( f^A \). However, when using the Shifted \( f^A \) and Shifted-Inverted \( f^A \), Proposed-Covariance is unable to find a solution with objective function close to 0 within the computational budget. This shows that as the correlation (or mutual information) between \( f^A \) and the objective function decreases, the added value of using \( k^A \) as the covariance function decreases too.
Proposed-Mean (Figure 3-6c) uses $f^A$ only in the prior mean function. When using the Shifted $f^A$, the optimization performance showed a slight deterioration compared to that with the Original $f^A$, illustrating the impact of using an analytical model, where the minimum does not align with the global minimum of the objective function, in the prior mean function. Furthermore, as the set of hyperparameter values used is the same as that used for the Original $f^A$ (i.e. $\alpha = 0.001$), this meant that the prior belief when using the Inverted and Shifted-Inverted $f^A$ is that the objective function behaves like a negative quadratic function. Since this is untrue, the optimization performance of Proposed-Mean with the Inverted and Shifted-Inverted $f^A$ is very poor. While choosing a suitable set of hyperparameter values would certainly help to improve the results when $f^A$ is inverted, it can be difficult to find the right
set of hyperparameter values in practice. Even if we choose to estimate the hyperparameter values (e.g., through maximum likelihood estimation), the limited number of observations poses a huge challenge to finding the right set of hyperparameter values for a high-dimensional problem. This suggests that if an inversion bias might exist in $f^A$, it may be best not to use $f^A$ in the prior mean function.

Proposed-Combined (Figure 3-6d) takes advantage of $f^A$ in both the prior mean function and the covariance function. When using the Inverted $f^A$, Proposed-Combined is still able to find a solution with objective function value close to 0. However, it required about 160 optimization iterations for the mean best objective function value to reach 0, compared with around 70 when using the Original $f^A$. This larger increase in number of iterations needed, compared with Proposed-Covariance, can be attributed to the fact that Proposed-Combined uses the Inverted $f^A$ in the prior mean function as well, which makes it a bad prior. However, it is still able to recover, highlighting the usefulness $k^A$ as the covariance function even if $f^A$ is biased. When working with the Shifted and Shifted-Inverted $f^A$, Proposed-Combined is unable to find a solution with objective function value close to 0 within the computational budget. Similar to Proposed-Covariance, this highlights decreased benefits of using $k^A$ as the covariance function when the correlation (or mutual information) between $f^A$ and the objective function is reduced. Working with the Shifted $f^A$, Proposed-Combined is able to reach its best mean objective function value (around 0.8) with just 30 iterations, compared with 200 iterations when working with the Shifted-Inverted $f^A$. Again, this can be attributed to the inversion of $f^A$ and its use in the prior mean function.

From Figure 3-6, we also see that Proposed-Covariance and Proposed-Combined with the Shifted and Shifted-Inverted $f^A$ still perform better than Standard and Proposed-Mean. Even when working with the Shifted and Shifted-Inverted $f^A$, Proposed-Covariance and Proposed-Combined are still able to find solutions with objective function values below 1, while Standard and Proposed-Mean are unable to do so even when working with the Original $f^A$. This helps to show the effectiveness of the $k^A$ covariance function in tackling high-dimensional problems, even when $f^A$ does not perfectly model the objective function. It also highlights the importance of
embedding surrogate information for exploration, more so than for exploitation.

We next consider the amount of exploration done in the \( f^A \)-space by each GP prior, along with the corresponding objective function values \( f \), for the different \( f^A \) models. Even though \( f^A \) is correlated to \( f \), its minimum may not coincide with the minimum of \( f \) (as in Shifted and Shifted-Inverted). As such, considering \( f^A \) as a metric of exploration allows us to see if the different GP priors will search in regions with bigger \( f^A \) values in their attempts to minimize \( f \). Figure 3-7 plots the observations and 2-D ecdfs to illustrate the distribution of \( f^A \) and \( f \) values explored. Each row of plots represent one of the 4 different GP priors, while each column represents one of the types of bias in \( f^A \). The first column of plots shows the distribution of observations for the Original \( f^A \), and are the exact same plots as the right column of Figure 3-5. Note that for the Inverted (second column) \( f^A \) and Shifted-Inverted (last column) \( f^A \), the definition of the 2-D ecdf is modified to highlight the anti-correlation between \( f^A \) and \( f \). The modified 2-D ecdf definition is

\[
ecdf(f^A, f) = \frac{1}{N} \times |\{(f_i^A, f_i) : f_i^A \geq f^A, f_i \leq f\}|, \tag{3.23}
\]

where the first “\( \leq \)” sign is changed to “\( \geq \)”. This changes the direction of the contour lines, so that the ecdf value increases as \( f^A \) becomes more negative, allowing the distribution of points to be visualized more easily. Otherwise, the interpretation of the 2-D ecdf for Inverted \( f^A \) and Shifted-Inverted \( f^A \) remain the same. For Shifted \( f^A \) (third column), the 2-D ecdf as defined by Eq. (3.22) is used.

We first focus on the Standard GP prior (first row of Figure 3-7). As previously mentioned, Standard does not make use of \( f^A \) in the prior, hence the exploration behavior should be independent of the type of bias in \( f^A \). As such the distribution of observations for Standard using the Inverted \( f^A \) and Shifted-Inverted \( f^A \) (Figures 3-7b and 3-7d) are essentially similar, and are mirror images of Figure 3-7a. The different distribution of observations seen in Figure 3-7c for the Shifted \( f^A \) can be attributed to randomness, where the BO run managed to identify a relatively good solution with \( f \) value around 1.1, leading to a cluster of observations around that region. In
Figure 3-7: 2-D empirical cumulative distribution functions illustrating exploration in $f^A$-space when bias is introduced to $f^A$. 
all cases, Standard does not explore as large a range of \( f^A \) values as compared to Proposed-Covariance or Proposed-Combined.

Moving on to the Inverted \( f^A \) (second column of Figure 3-7), we first note the inverse relationship between \( f^A \) and \( f \), due to the anti-correlation between Inverted \( f^A \) and \( f \). We also see that Proposed-Covariance and Proposed-Combined (Figures 3-7f and 3-7n respectively) explored a larger range of \( f^A \) values, compared to Standard and Proposed-Mean (Figures 3-7b and 3-7j respectively). Similar to the case with Original \( f^A \), the \( k^A \) covariance function encourages exploration in the \( f^A \)-space, regardless of the sign and magnitude of the correlation between \( f^A \) and \( f \). Moreover, the maximum of the Inverted \( f^A \) (i.e. \( f^A = 0 \)) is aligned with the minimum of \( f \), with the maximum \( f^A \) value occurring at a unique point (i.e. \( x = 0 \)), hence Proposed-Covariance and Proposed-Combined are still able to efficiently find solutions with \( f \) close to 0 when exploring in regions with \( f^A \) values close to 0. In contrast, Proposed-Mean (Figure 3-7j) only uses Inverted \( f^A \) in its prior mean function, which is completely inaccurate in describing the objective function. As a result of the misinformation in the prior, Proposed-Mean tends to explore regions with smaller (i.e. more negative) \( f^A \) values and greater \( f \) values, compared to when using the Original \( f^A \). At the same time, the range of \( f^A \) values explored by Proposed-Mean (-3033 to -4659) is smaller when using the Inverted \( f^A \) (264 to 4544), compared to Original \( f^A \).

The Shifted \( f^A \) model (third column of Figure 3-7) illustrates the case when the minimum of \( f^A \) no longer aligns with the global minimum of \( f \). With Shifted \( f^A \), the global minimum has an \( f^A \) value of 10, which is an \( f^A \) value that no longer occurs at a unique point. In other words, there are other points, which are not the global minimum, in the feasible region with \( f^A \) values of 10 as well. While the \( k^A \) covariance function in Proposed-Covariance and Proposed-Combined still ensures that they still explore a large range of \( f^A \) values, as shown in Figures 3-7g and 3-7o, this implies that exploring the \( f^A \)-space does not always lead to finding the global minimum of \( f \). However, from Figure 3-7g, Proposed-Covariance seemed to have allocated more simulation budget to exploring the region with \( f^A \) values close to 0, as compared to Proposed-Combined where the observations are more uniformly
distributed (Figure 3-7o). This could have led to the better optimization performance for Proposed-Covariance (mean of 0.665) compared to Proposed-Combined (mean of 0.838) when using Shifted $f^A$, as shown in Figure 3-6. In the case of Proposed-Mean (Figure 3-7k), the exploration behavior when using the Shifted $f^A$ is similar to that when using the Original $f^A$ – Proposed-Mean does not explore as large a range of $f^A$ values as compared to Proposed-Covariance and Proposed-Combined. As a result, this also limits the best $f$ value it can find to above 1.

Working with the Shifted-Inverted $f^A$ model (last column of Figure 3-7) results in similar exploration behavior as when using the Inverted $f^A$ model for the 4 different GP priors. Proposed-Covariance and Proposed-Combined (Figures 3-7h and 3-7p respectively) continue to explore a much larger range of $f^A$ values as compared to Standard and Proposed-Mean (Figures 3-7d and 3-7l). However, as the maximum of Shifted-Inverted $f^A$ does not coincide with the global minimum of $f$, the global minimum has an $f^A$ value of -10 which is an $f^A$ value that does not occur at a unique point. This thus implies that simply exploring the $f^A$-space has a lower chance of finding the minimum of $f$. In the case of Proposed-Mean, the inversed relationship of the Shifted-Inverted $f^A$ and $f$ again resulted in Proposed-Mean exploring regions with smaller (more negative) $f^A$ values, along with a smaller range of $f^A$ values compared to Original $f^A$ and Shifted $f^A$.

In general, the experiments in this section show that the use of $k^A$ as the covariance function encourages exploration in the $f^A$-space regardless of biases in $f^A$ when modeling the objective function. To efficiently identify the minimum in the objective function, the optimum in $f^A$ should ideally coincide with the minimum in the objective function, and the $f^A$ value at the objective function minimum should occur only at a unique point in the feasible region. In the case of inversion bias, the results suggest that using $f^A$ in the prior mean function may not be advisable. However, as the Shifted and Shifted-Inverted cases show, the correlation between $f^A$ and $f$ can still improve high-dimensional BO performance when using the $k^A$ covariance function, compared to a general-purpose squared exponential covariance function, highlighting the robustness of the proposed method.
3.4 Case Study: Midtown Manhattan Traffic Signal Control

In this section, we test our proposed method in a case study involving a high-dimensional traffic signal optimization problem for Midtown Manhattan. We first describe the traffic signal optimization problem in Section 3.4.1. The analytical surrogate model used for the case study is described in Section 3.4.2. The results are then presented in Section 3.4.3.

3.4.1 Traffic Signal Optimization Problem

In this case study, we apply our proposed method to a high-dimensional fixed time traffic signal optimization problem for the large-scale area of Midtown Manhattan (MTM) in New York City. For a review of traffic signal optimization terminology, see Osorio (2010, Appendix A, pages 119-121). In fixed time signal control, the signal plan is cyclic (i.e. periodic) with a fixed cycle time (i.e. the time required to complete one sequence of signals). The decision variables in this problem are the green splits (i.e. normalized green times) of each signal phase of each intersection in the network. Other control variables, such as the offsets, stage structure, cycle times, etc., are predetermined and kept constant. The notation used for formulation of the traffic signal optimization problem is given below (also summarized in Appendix B.1):

\[ f \]  SO objective function (expected travel time of vehicles in the road network);
\[ F \]  random variable denoting the travel time of vehicles in the road network;
\[ x_j \]  green split of signal phase \( j \) (decision variable);
\[ x \]  vector of all green splits (decision vector);
\[ z \]  vector of endogenous simulation variables.

**Exogenous problem parameters:**

\[ c_\ell \]  cycle time of intersection \( \ell \);
\[ d_\ell \]  fixed cycle time of intersection \( \ell \);
\[ x^{LB}_\ell \]  vector of minimal green splits;
\[ p \]  vector of exogenous simulation parameters;
\[ I \]  set of signal controlled intersection indices;
\[ P(\ell) \]  set of signal phase indices of intersection \( \ell \).
The formulation of the problem is then given by:

$$\min_x f(x, z; p) = \mathbb{E}[F(x, z; p)]$$  \hspace{1cm} (3.24)$$

subject to

$$\sum_{j \in \mathcal{P}(\ell)} x_j = \frac{c_\ell - d_\ell}{c_\ell}, \hspace{0.2cm} \forall \ell \in \mathcal{I}$$  \hspace{1cm} (3.25)$$

$$x \geq x^{LB}.$$  \hspace{1cm} (3.26)$$

The objective function for this case study (Eq. (3.24)) is the expected travel time of vehicles, as evaluated by a stochastic traffic simulator whose output is represented by the random variable $F$. The objective function depends on a vector of exogenous parameters $p$, which accounts for the road network topology and fixed lane attributes (e.g. lane length, maximum speed, grade), for instance. The endogenous simulation variables $z$ represents, for example, route choice decisions, as well as link-level and network-level performance metrics like travel times, speeds, densities, delays, etc. For more details about how the objective function is computed from the stochastic traffic simulator, we refer the reader to Appendix B.3.

The feasible region of $x$ is defined by the constraints (3.25) and (3.26). Eq. (3.25) represents the cycle time constraint, which states that for a given intersection, the sum of green splits must be equal to the proportion of cycle time that can be optimized (i.e. not fixed). In practice, it is common to assign fixed amounts of time to certain traffic phases, typically for safety considerations and to comply with local transportation regulations. For instance, a fixed amount of the cycle time is typically assigned to all-red periods, where the signal is red for all traffic movements between some signal phases for safety reasons. Eq. (3.26) represents the lower bounds of the green splits, where the minimum green splits are typically determined by the local transportation authorities based on safety considerations.

The MTM area being simulated in this case study is demarcated by a rectangle in the map shown in Figure 3-8. In this problem, we simulate traffic from 3pm - 6pm, and optimize the signal plans for the peak hour of 5pm - 6pm. We control a total of 97 signalized intersections, with 259 green splits (i.e. decision variables). Due to
the linear cycle time equality constraint (Eq. (3.25)), this can be considered a 162-D problem (i.e. $259 - 97 = 162$), making it high-dimensional in the field of BO.

The MTM simulation model is implemented using the Aimsun software (TSS-Transport Simulation Systems 2015). It consists of a total of 698 roads, 2756 lanes and 444 intersection. The complete network topology of the simulation model is illustrated in Figure 3-9. During the simulated interval of 5pm - 6pm, the expected demand is over 29,000 trips per hour, distributed across more than 3500 origin-destination pairs. In this simulation model, the minimum green time that can be assigned to each signal phase is 6 seconds. Hence, the corresponding elements in the vector of minimum green splits $x^{LB}$ (Eq. (3.26)) are the ratio of 6 seconds to the cycle time of the intersection that the green split belongs to.

In this case study, BO is implemented as described in Algorithm 3, with 10 initial observations provided to fit the GP posteriors. The computational budget is taken as 55 iterations (i.e. 45 optimization iterations). As with the Griewank function experiments in Section 3.3, the objective function estimate is obtained by taking the mean of 4 simulations, with 2 additional simulations of the best solution at the end of each optimization iteration. This leads to a total of 310 simulation runs, 40 of which are spent on the 10 initial observations.

For each GP prior, we consider 3 distinct initial sets, each containing the 10 initial observations used to fit the GP posteriors at the start of the BO run. The 10 initial observations in each set are obtained by sampling within the feasible region (as defined by Eq. (3.25) and (3.26)) uniformly at random using the code of Stafford (2006), and taking the mean of 4 simulation runs for each point. In addition, one of the uniformly random points in Initial Set 3 is replaced by an existing signal plan, which was previously used by the New York City Department of Transportation (NYCDOT) for the MTM area. This existing signal plan is known to perform well, and hence is included in Initial Set 3 to investigate how BO will be affected when the initial set contains a solution with good performance. We also use the existing plan as a benchmark for the performance of the signal plans proposed by BO with the 4 different GP priors. For more implementation details, we refer the reader to
Figure 3-8: Map of Midtown Manhattan with the simulated area demarcated by a rectangle (MapQuest.com, Inc 2018).

Figure 3-9: Midtown Manhattan model in Aimsun.
Appendix B.3.

### 3.4.2 Queueing Network Model

For the traffic signal optimization problem, the underlying road network of the MTM area is known beforehand. Hence, this problem-specific prior information can be used to inform the GP prior through the prior mean function and/or the covariance function. To do this, we model the road network using a finite capacity queueing network model (Osorio and Chong 2015, Eq. 6). The queueing network model accounts for vehicular spillbacks (when downstream lanes are full, thus blocking traffic flow from upstream lanes) through the queueing theoretic concept of blocking. In this queueing network model, each lane in the network is represented by a finite space capacity $M/M/1/k$ queue, where $k$ denotes the finite space capacity of that lane.

The green splits of the traffic signal optimization problem are then related to the queueing network model, through their effect on the service rates of the queues at the corresponding intersections (see Eq. 18 of Osorio and Chong (2015)). Based on the queueing network model, the analytical approximation $f^A$ of the objective function (i.e. expected travel time) can be derived (Osorio and Chong 2015, Eq. 11).

We use the notation of Osorio and Chong (2015), where the index $i$ refers to a given queue:

- $\gamma_i$: external arrival rate;
- $\lambda_i^{\text{eff}}$: effective arrival rate;
- $\mu_i$: service rate;
- $\rho_i^{\text{eff}}$: effective traffic intensity;
- $k_i$: upper bound of the queue length;
- $N_i$: total number of vehicles in queue $i$;
- $P(N_i = k_i)$: probability of queue $i$ being full;
- $p_{ij}$: transition probability from queue $i$ to queue $j$;
- $\mathcal{D}_i$: set of downstream queues of queue $i$. 

111
The queuing network model is given by the following system of equations:

\[ \lambda_i^{\text{eff}} = \gamma_i (1 - P(N_i = k_i)) + \sum_j p_{ij} \lambda_j \quad (3.27a) \]

\[ \rho_i^{\text{eff}} = \frac{\lambda_i^{\text{eff}}}{\mu_i} + \left( \sum_{j \in \mathcal{D}_i} p_{ij} P(N_j = k_j) \right) \left( \sum_{j \in \mathcal{D}_i} \rho_j^{\text{eff}} \right) \quad (3.27b) \]

\[ P(N_i = k_i) = \frac{1 - \rho_i^{\text{eff}}}{1 - (\rho_i^{\text{eff}})^{k_i+1}} (\rho_i^{\text{eff}})^{k_i}. \quad (3.27c) \]

Eq. (3.27a) is a flow conservation equation relating the demand rate of queue \( i \) (left hand side of the equality) to the sum of the demand rate of vehicle trips that start in queue \( i \) (first term of the right hand side) and of the demand rate of vehicle trips that arise from upstream queues (second term of the right hand side). More specifically, the demand rate for trips that start in queue \( i \) is represented by \( \gamma_i \), and the term \( (1 - P(N_i = k_i)) \) enforces that trips can only start in queue \( i \) if it is not full. Eq. (3.27b) defines the traffic intensity. The first term of the right hand side is the traffic intensity when the queue is not full (i.e., when there is no spillback). The second term accounts for the impact in queue \( i \) due to spillbacks from its downstream queues. Eq. (3.27c) gives the expression of the spillback probability (i.e., the blocking probability) as defined for an \( M/M/1/k_i \) queue.

The endogenous variables of the above system of equations are related to the decision vector (the green split vector \( \mathbf{x} \)) by the following linear equations:

\[ \mu_i = s \left( e_i + \sum_{j \in \mathcal{P}_j(i)} x_j \right) \quad \forall i \in \mathcal{L}, \quad (3.28) \]

where \( s \) denotes an exogenous scalar that represents the saturation flow rate (i.e., maximum queue discharge rate), and \( e_i \) is an exogenous parameter that represents the ratio of fixed green time to cycle time for signalized queue \( i \). Eq. (3.28) states that the service rate of a signalized queue \( i \) is given by the saturation rate scaled by the proportion of cycle time the queue has a green phase. This proportion is given by the term in parenthesis, which depends on the fixed (i.e., not optimized) time (term
\( e_i \) and the variable time (summation term, which represents the sum of the green splits of the signal phases of queue \( i \)).

The expected travel time (i.e. the objective function) can be approximated by the queueing network model. This is done by applying Little’s law (Little 1961) to the entire network:

\[
f^A(x) = \frac{\sum_i \mathbb{E}[N_i]}{\sum_i \gamma_i(1 - P(N_i = k_i))}
\]  

(3.29)

where \( \mathbb{E}[N_i] \) represents the expected number of vehicles in lane \( i \). The numerator of Eq. (3.29) represents the expected number of vehicles in the entire network, while the denominator represents the expected arrival rate to the network. The ratio of these two quantities provide the expected travel time of the network according to Little’s law.

The expected number of vehicles in lane \( i \) can be computed approximately as such:

\[
\mathbb{E}[N_i] = \rho_i^{\text{eff}} \left( \frac{1}{1 - \rho_i^{\text{eff}}} - \frac{(k_i + 1)(\rho_i^{\text{eff}})^{k_i}}{1 - (\rho_i^{\text{eff}})^{k_i+1}} \right)
\]  

(3.30)

The derivation of Eq. (3.30) can be found in Appendix A of Osorio and Chong (2015). For more details about the queueing network model and the derivation of the analytical approximation of the objective function, we refer the reader to Osorio and Chong (2015, Sections 3, 4 and Appendix A).

### 3.4.3 Results

We compare the optimization performance of the 4 different GP priors in Figure 3-10. Each plot in Figure 3-10 shows a comparison of the optimization performance for a different initial set. The x-axis represents the optimization iteration (i.e. does not count the 10 initial observations), while the y-axis shows the expected travel time (objective function value). In each plot, each line depicts the mean of the best solution at a given iteration for 5 BO runs, with the shaded regions showing the values \( \pm 1 \) standard error away from the mean. The mean travel time and its standard error
based on 50 simulations using the existing plan is also indicated in each plot by the thin red dashed line and the red shaded region. This allows for comparison with the performance of BO with the 4 different GP priors.

We further tested if the differences in optimization performance of the different GP priors are statistically significant, by performing a one-sided paired $t$-test based on the mean performance of the 5 BO runs for each initial set. The results of the $t$-test are given in Table 3.2. For each row in the table, the null hypothesis assumes that the first GP prior (in Column 1) obtained a mean performance that is worse than or equal to the mean performance of the second GP prior (in Column 1). In contrast, the alternative hypothesis states that the first GP prior obtained a mean performance that is better than the second GP prior. For instance, the last row of Table 3.2 tests the alternative hypothesis that Proposed-Covariance obtained a better
Table 3.2: One-sided paired t-test results

<table>
<thead>
<tr>
<th>Test</th>
<th>Initial Set 1</th>
<th></th>
<th>Initial Set 2</th>
<th></th>
<th>Initial Set 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>t-stat</td>
<td>p-value</td>
<td>t-stat</td>
<td>p-value</td>
<td>t-stat</td>
<td>p-value</td>
</tr>
<tr>
<td>Proposed-Covariance vs. Standard</td>
<td>-2.546</td>
<td>0.0318</td>
<td>-3.625</td>
<td>0.0111</td>
<td>-0.851</td>
<td>0.221</td>
</tr>
<tr>
<td>Proposed-Mean vs. Standard</td>
<td>-0.369</td>
<td>0.365</td>
<td>-1.956</td>
<td>0.0611</td>
<td>0.577</td>
<td>0.702</td>
</tr>
<tr>
<td>Proposed-Combined vs. Standard</td>
<td>-1.675</td>
<td>0.0846</td>
<td>-3.024</td>
<td>0.0195</td>
<td>0.521</td>
<td>0.685</td>
</tr>
<tr>
<td>Proposed-Covariance vs. Proposed-Mean</td>
<td>-2.943</td>
<td>0.0211</td>
<td>-0.448</td>
<td>0.339</td>
<td>-1.020</td>
<td>0.183</td>
</tr>
<tr>
<td>Proposed-Combined vs. Proposed-Mean</td>
<td>-2.687</td>
<td>0.0274</td>
<td>-0.690</td>
<td>0.264</td>
<td>0.176</td>
<td>0.565</td>
</tr>
<tr>
<td>Proposed-Covariance vs. Proposed-Combined</td>
<td>-1.738</td>
<td>0.0786</td>
<td>0.095</td>
<td>0.536</td>
<td>1.759</td>
<td>0.0767</td>
</tr>
</tbody>
</table>

mean performance than Proposed-Combined. Both the t-statistics and p-values are shown for each test and for all 3 initial sets. Each t-test is considered at the 10% level of significance. With 4 degrees of freedom, the corresponding critical value of the t-statistic is -1.533. The t-tests with t-statistics smaller than the critical value, which are displayed in bold, have their null hypotheses rejected.

The difference in performance between the existing plan and the BO solutions were also tested for statistical significance using a one-sided two-sample t-test. The results of the t-test are shown in Table 3.3. The null hypothesis assumes that the GP prior has a mean performance that is worse than or equal to the mean performance of the existing plan. The alternative hypothesis states that the GP prior has a better mean performance than the existing plan. Similar to Table 3.2, the t-tests with t-statistics smaller than the critical value at 10% level of significance are displayed in bold.

We first consider Initial Set 1 (Figure 3-10a). Of all the GP priors, Proposed-Covariance (blue dashed line) is able to identify the best point on average after expending the computational budget. In fact, Proposed-Covariance is statistically significantly better than all of the other 3 GP priors as seen in Table 3.2. It identified solutions that reduced the expected travel time by 3.2% on average compared to
the existing plan. It is also the only GP prior that has a mean performance which is statistically better than the existing plan. Proposed-Combined (green solid line) obtained solutions with the next best performance, and is statistically significantly better than Proposed-Mean and Standard. This is followed by Proposed-Mean (magenta dotted line) and then Standard (black dash-dot line). However, the mean performance of Proposed-Mean is not statistically better than that of Standard. While Proposed-Covariance is the best GP prior based on results at the end of the BO runs, Figure 3-10a also shows that it is the slowest in finding solutions better than the existing plan. Proposed-Mean and Proposed-Combined are actually able to identify solutions better than the existing plan much more quickly (5 and 2 iterations respectively) than both Proposed-Covariance (15 iterations). Standard is unable to identify solutions that are significantly better than the existing plan. This suggests that $f^A$ in the prior mean function is being exploited to find good solutions quickly.

For Initial Set 2 (Figure 3-10b), Proposed-Covariance, Proposed-Mean and Proposed-Combined are all able to outperform Standard at the end of the BO runs. The differences in performance compared to Standard are statistically significant at the 10% level of significance as seen in Table 3.2, highlighting the usefulness of incorporating problem-specific information in the GP prior. The performance of Proposed-Covariance, Proposed-Mean and Proposed-Combined were also statistically better than the existing plan, with Proposed-Covariance registering a 3.3% reduction in expected travel time on average relative to the existing plan. However, there is no

<table>
<thead>
<tr>
<th>Test</th>
<th>Initial Set 1</th>
<th></th>
<th>Initial Set 2</th>
<th></th>
<th>Initial Set 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t$-stat</td>
<td>$p$-value</td>
<td>$t$-stat</td>
<td>$p$-value</td>
<td>$t$-stat</td>
<td>$p$-value</td>
</tr>
<tr>
<td>Standard vs. Existing Plan</td>
<td>-0.483</td>
<td>0.316</td>
<td>-0.407</td>
<td>0.343</td>
<td>-1.258</td>
<td>0.107</td>
</tr>
<tr>
<td>Proposed-Covariance vs. Existing Plan</td>
<td>-1.914</td>
<td>0.0304</td>
<td>-1.986</td>
<td>0.0260</td>
<td>-1.742</td>
<td>0.0435</td>
</tr>
<tr>
<td>Proposed-Mean vs. Existing Plan</td>
<td>-0.581</td>
<td>0.282</td>
<td>-1.575</td>
<td>0.0605</td>
<td>-1.119</td>
<td>0.134</td>
</tr>
<tr>
<td>Proposed-Combined vs. Existing Plan</td>
<td>-1.163</td>
<td>0.125</td>
<td>-2.037</td>
<td>0.0232</td>
<td>-1.054</td>
<td>0.148</td>
</tr>
</tbody>
</table>
significant difference between Proposed-Covariance, Proposed-Mean and Proposed-Combined. Furthermore, as with Initial Set 1, Proposed-Mean and Proposed-Combined are again able to quickly identify solutions better than the existing plan (1 and 3 iterations respectively), compared to Proposed-Covariance (18 iterations). Standard is again unable to identify solutions that are significantly better than the existing plan.

Initial Set 3 (Figure 3-10c) contains the existing plan as one of the initial points. It is the point with the best performance in the initial set. The reason the starting mean performance of all the GP priors is better than that of the existing plan (i.e. the red dashed line) is due to chance. With a signal plan with good performance already in the initial set, Standard is able to perform well in this case, as it simply explored around the vicinity of the existing plan in the feasible region to find better solutions (more evidence of this behavior is shown later in Figures 3-11 and 3-12). As a result, Standard is able to find solutions by the end of the BO runs which are statistically on par with those obtained by Proposed-Covariance, Proposed-Mean and Proposed-Combined, as shown by the results of the $t$-tests in Table 3.2. However, the solutions found by Standard, Proposed-Mean and Proposed-Combined were not statistically better than the existing plan. Hence, this shows that if the initial set contains an observation with good performance, the importance of access to problem-specific prior information may be diminished. In this case, Proposed-Covariance is the only GP prior that performed better than the existing plan based on the $t$-test. It reduced the expected travel time by 2.9% on average, compared to the existing plan.

In general, Figure 3-10 showed that there is value in using the $k^A$ covariance function, as it is able to consistently identify solutions with better or equal performance than GP priors using the squared exponential covariance function. In addition, the results also show that placing $f^A$ in the prior mean function allows for the problem-specific information to be exploited, such that BO can quickly find good solutions within a few iterations. However, as Figures 3-10a and 3-10c show, placing $f^A$ in the prior mean function could limit the best solution found in the long run, indicating that more exploration may be helpful.

Figure 3-11 shows the $f^A$ values and the corresponding $f$ values for all evaluated
points. It provides insights into the amount of exploration done in the $f^A$-space by each GP prior. In each plot of Figure 3-11, the $f^A$ and $f$ values of every simulated point is illustrated, along with the 2-D ecdf to provide a visualization of the distribution of $f^A$ and $f$ values explored. The 2-D ecdf here is computed based on Eq. (3.22). Each row of plots represent one of the 4 different GP priors being tested, while each column shows the results for a different initial set.

Focusing first on Initial Set 1 (first column of Figure 3-11), we see that Standard explored quite a large range of $f^A$ values (2.79 - 6.11) compared to Proposed-Mean and Proposed-Combined. However, the $f$ values that it found are not as good, with the best $f$ value being 11.22. In contrast, Proposed-Covariance also explored a large range of $f^A$ values (2.80 - 5.96), but the solutions it identified included more with smaller $f$ values (around 11), particularly around the region with $f^A$ values between 3 and 4.5. This shows that the minimum of $f^A$ may not necessarily coincide with the minimum of $f$, but the $k^A$ covariance function can still help to identify solutions with smaller $f$ values. Proposed-Mean and Proposed-Combined explored a smaller range of $f^A$ values than Standard and Proposed-Covariance. In fact, most of the evaluated points have $f^A$ values less than 3.5, indicating that the Proposed-Mean and Proposed-Combined are exploiting the use of $f^A$ in the prior mean function.

Moving on to Initial Set 2 (second column of Figure 3-11), Standard seemed to have found a point with good performance (with $f^A$ value of 2.79) and kept exploring around the vicinity, which explains the cluster of evaluated points with $f^A$ values of about 2.8. Further evidence of this behavior is shown later in Figure 3-12. Proposed-Covariance again explored a large range of $f^A$ values (3.36 - 7.85). Furthermore, it identified many points with large $f^A$ but small $f$ values, providing further evidence that the $k^A$ covariance function is helpful in the search for small $f$ values. From Figure 3-11h, Proposed-Mean mainly focused on finding points with smaller $f^A$ again. On the other hand, Proposed-Combined explored a larger range of $f^A$ values compared to Proposed-Mean this time. However, the main focus is still on the regions with smaller $f^A$ values.

For Initial Set 3 (third column of Figure 3-11), which contains the existing plan
as one of the initial observations, Standard spent most of the computational budget exploring points close to the existing plan with the hope of finding better solutions. However, it also explored other regions with larger $f^A$ values, but the points explored with large $f^A$ values also have large $f$ values. Proposed-Covariance, Proposed-Mean
and Proposed-Combined similarly explored only in the vicinity of the existing plan (see Figure 3-12 for more evidence). As a result, their exploration is limited to just a small range of $f^A$ values. This could be the result of having an observation with good performance in the initial set, which could affect the hyperparameter values obtained through maximum likelihood estimation, such that the effect of $f^A$ in the covariance function on exploration in the $f^A$-space is diminished. Hence, this shows the sensitivity of BO to the initial set.

To visualize the distribution of the observations in the feasible region (i.e. the space of feasible signal plans), we made use of multidimensional scaling (MDS, see e.g. Cox and Cox (2008)) to project the points onto a 2-D space. Simply put, MDS is a projection technique that retains the original pairwise distances between points as much as possible. Here, we used the Matlab function \texttt{cmdscale} to compute the projection based on the pairwise Euclidean distances between the observations. Figure 3-12 illustrates the positions of the observations relative to one another in this 2-D space. The contour lines in the plots indicate the objective function value at a given point. Each column represents a different initial set, while each row shows a different GP prior. Note that the observations from all the GP priors and all 3 initial sets are used to define the MDS 2-D space, hence the 2-D space and contour line layout in each plot are the same. This allows for direct comparison across the plots for different initial sets. Furthermore, we note that the relative errors of representing the observations in just 2 dimensions are naturally large (92%, 81% and 82% for Initial Sets 1,2 and 3 respectively). Hence, we do not use the MDS plots to draw independent conclusions, but merely use it to substantiate other claims.

Comparing across the 3 columns in Figure 3-12, we see that the regions explored in the MDS 2-D projected space can be quite different for each initial set. This supports the notion that BO is sensitive to the initial set used. The first and second columns of Figure 3-12 even show that the Proposed-Covariance explores significantly different regions of the 2-D space compared to Proposed-Mean and Proposed-Combined, highlighting that the different GP priors react differently to the initial sets as expected. The first column of Figure 3-12 also supports the explanation for
Figure 3-12: Multidimensional scaling 2-D projections of all observations in the feasible region.

the cluster of observations by Standard with $f^A$ values around 2.8 as seen in Figure 3-11b – there is a cluster of observations by Standard around $(−0.25, −0.3)$ in Figure 3-12b, suggesting that Standard got stuck in a local minimum, and hence ex-
plored only in the vicinity. For Initial Set 3, we previously observed that all 4 GP priors explored only an area with small $f^A$ values, suggesting that they are exploring only in the vicinity of the existing plan. The third column of Figure 3-12 reinforces this claim, as it shows most of the observations surrounding the initial observation at \((-0.30, -0.07)\) which corresponds to the existing plan.

As mentioned in Section 3.2.2, there is a trade-off in terms of computational runtime when using an analytical surrogate model in the GP. The additional runtime occurs mostly during the optimization of the acquisition function at every iteration. Figure 3-13 plots the ecdfs of computational runtimes for all 4 GP priors. The ecdfs are each based on 135 observations (45 iterations × 3 initial sets). The ecdfs in Figure 3-13 represent the fraction of observations with a corresponding computational runtime or less. In other words, an ecdf curve that is shifted more to the right indicates a greater computational runtime on average. Figure 3-13a plots the distribution of runtimes for optimizing the acquisition function. It shows that Standard requires the least computational runtime (24 s on average) to optimize the acquisition function. This is followed by Proposed-Mean (197 s), Proposed-Covariance (604 s) and Proposed-Combined (793 s). This indicates that using the analytical surrogate model in the covariance function results in greater computational runtime when optimizing the acquisition function, compared to using it in the prior mean function. Embedding the surrogate model in both the prior mean function and covariance function (as in Proposed-Combined) resulted in an even larger runtime. It should be noted that the increase in computational runtime depends the analytical surrogate model used. A surrogate model that can be evaluated more quickly would result in a smaller increase in computational runtime. Hence, when choosing incorporate an analytical surrogate model in the GP, the trade-off between improvement in optimization performance and computational runtime should be considered.

Figure 3-13b plots the distribution of total computational runtime per iteration for the 4 GP priors. In each iteration, most of the runtime is spent on evaluating the traffic simulator, in addition to optimizing the acquisition function. Based on Figure 3-13b, Standard requires the least runtime per iteration (1560 s on average),
Figure 3-13: Empirical cdf illustrating distribution of (a) computational runtimes for optimizing the acquisition function at each iteration, and (b) total computational runtime per iteration.

followed by Proposed-Mean (1750 s), Proposed-Covariance (2200 s) and Proposed-Combined (2340 s). The average runtime per iteration shows that most of the time is spent evaluating the simulations. It also shows that the additional runtime required when using the analytical surrogate model in the GP prior is small compared to the runtime required for the simulation evaluations.

3.5 Conclusion

In this chapter, we showed how problem-specific surrogate models can be used in BO to tackle high-dimensional problems efficiently. Incorporating the surrogate model in the GP allows the problem-specific information to be used for both exploration and exploitation. More specifically, exploitation of the problem-specific information can be achieved by using the surrogate model in the prior mean function of the GP, allowing the algorithm to identify good solutions more quickly. On the other hand, incorporating a surrogate model in the GP covariance function can encourage exploration in areas of the feasible region with different surrogate model values from points which have already been evaluated. This allows better solutions to be found even if the surrogate model is not an accurate representation of the objective function. The results show that having the surrogate model in the GP covariance function
leads to better optimization performance than not providing any problem-specific information in the covariance function. In addition, using the surrogate model in the prior mean function allows BO to quickly find good solutions within a few iterations. However, it could limit the best solution found in the long run. This suggests that if the computational budget is very limited, it might be better to include the surrogate model in the prior mean function.

At the same time, our results also show that incorporating the surrogate model in the covariance function (as opposed to the prior mean function only) helped the optimization be more robust to biases in the surrogate model. This includes both inversion and translation (i.e. shifted) biases. This opens up the possibility of using models which are less accurate but are still (anti-)correlated to the objective function. For instance, the surrogate model does not even have to be modeling the objective function itself, but instead it could be modeling another quantity which is (anti-)correlated to the objective function (e.g. network throughput can be anti-correlated with the expected travel time). On the other hand, if there is an inversion bias in the surrogate model (i.e. anti-correlation), using the surrogate model in the prior mean function can lead to worse performance.

It is also worth noting the importance of choosing a surrogate model that is computationally tractable, while still being accurate. The runtime required to evaluate the surrogate model can be a major contributor to the overall BO runtime. As such, this is especially important if computational runtime is a major concern.

Also, while the proposed analytical model-based covariance function was demonstrated on a transportation case study problem, we would like to emphasize the generality of the method, as it is not limited to just transportation optimization or SO problems. The proposed covariance function can be applied to any problem type in general where the objective function is expensive to evaluate, and where there is problem-specific prior information available in the form of an analytical model.

As part of our ongoing research, we are investigating possible ways to extend the use of surrogate model-based covariance functions and BO to high-dimensional dynamic problems. In transportation, it is common to encounter dynamic optimization
problems, where the objective function and decision variables are time-dependent, as practitioners try to account for the spatiotemporal dynamics of travelers in the system. To tackle dynamic problems, the GP posterior model would have to account for time variations of the objective function. Due to the difficulties of modeling a high-dimensional dynamic problem accurately, we believe that BO, together with the use of problem-specific prior information, can be a good option for solving such simulation-based optimization problems.

It is also notable that using the surrogate model in the prior mean function allows good solutions to be found quickly, but limits longer term performance. This suggests that using the surrogate model in the prior mean function for a given number of iterations, before switching to a constant prior mean function and placing the surrogate model in the covariance function might lead to better performance within the limited computational budget. This sequential approach could quickly find good solutions, without sacrificing the long term performance.

Another area that can be studied further is the choice of functional forms for the surrogate model-based covariance function. For the purpose of this chapter, we focused on using a squared exponential form of the difference between analytical model values of two points in the proposed covariance function. However, other functional forms might provide a better description of the covariance between points, and hence may lead to more efficient optimization. For instance, if the objective function is known to not be smooth (i.e. infinitely differentiable), a functional form based on the Matérn covariance function may be more appropriate.

Despite the effectiveness of the proposed covariance function for high-dimensional BO, we note that it can be difficult to find a good set of hyperparameter values. This is especially so with limited observations in high-dimensional settings. Furthermore, the set of initial points used to fit the GP posterior at the start of the BO run can affect the estimated hyperparameter values, and has a strong impact on the eventual outcome. As such, more studies can be done to identify ways to better estimate hyperparameter values that can allow for more efficient optimization to take place.

We also found that BO is sensitive to the set of initial points used to fit the GP at
the start of the optimization process. This is regardless of whether problem-specific prior information is incorporated in the GP prior mean function and/or the covariance function or not. Hence, work could be done to better sample for initial points while keeping computational costs in check, especially for high-dimensional BO problems.
Chapter 4

Bayesian Optimization for High-Dimensional Dynamic Urban Transportation Problems: A Traffic Model-Based Approach

In this chapter, we present an extension of the Bayesian optimization (BO) method proposed in Chapter 3 to tackle the class of dynamic problems, which include time-dependent decision variables. The Gaussian process (GP) priors of the BO approach of Chapter 3 relied on the use of an analytical stationary traffic model. In this chapter, problem-specific prior information is incorporated into the GP prior using an analytical surrogate model that accounts for the temporal dynamics of traffic in the transportation network. The proposed surrogate model is a computationally efficient dynamic traffic network model that is suitable for optimization purposes, while being able to model the temporal dynamics for large-scale networks. The use of a GP prior based on a dynamic, rather than a stationary traffic model, enhances the ability of the BO framework to tackle dynamic SO problems (i.e. SO problems with time-dependent decision variables).
4.1 Introduction

Dynamic optimization problems, i.e., optimization problems with time-dependent decision variables, are naturally ubiquitous in the field of urban transportation. The underlying systems oftentimes vary as a function of time. For instance, the level of demand, as well as demand pattern (e.g. direction of travel, route choice, etc.), for the urban road network or public transit network changes depending on the time of day. A common approach used in transportation optimization is to decompose the dynamic problem into a set of independent stationary problems, with one problem per time interval. However, accounting for the dynamics, as well as for the temporal dependency across the time intervals is a more realistic representation of the underlying dynamic problems and can yield solutions with enhanced performance.

In this chapter, we focus on offline dynamic transportation SO problems of the following form:

$$\min_{x_1, \ldots, x_L \in \chi} f(x, z; p) \equiv \sum_{l=1}^{L} \alpha_l \mathbb{E}[F_l(x_l, z_l; p)].$$ (4.1)

The SO problem considers $L$ sequential time intervals. In each time interval $l \in \{1, \ldots, L\}$, we consider a continuous decision vector $x_l$ (e.g. traffic signal plan for time interval $l$) that resides in the feasible region $\chi$, where the constraints are assumed to be analytical and differentiable. In this work, the constraints are also convex. More specifically, we use bound and linear equality constraints. The objective function $f$ is the weighted sum of the expectation of a simulation-evaluated network performance metric $F_l$ during time interval $l$ (e.g. average trip travel time, network throughput, etc.), where $\alpha_l$ denotes the weight coefficient for time interval $l$. The network performance metric $F_l$ depends on the interval-specific decision vector $x_l$ and the vector of endogenous variables $z_l$ (e.g. traffic assignment, link densities, etc.), as well as the vector of exogenous parameters $p$ (e.g. network topology). The decision vector containing all time intervals is denoted $x = (x_1, \ldots, x_L)$. Similarly, the vector of endogenous variables (e.g. traffic assignment, link densities, etc.) for all time inter-
The network performance metric $F_l$ depends on the decision vector $x$ and the vector of endogenous variables $z$, as well as the vector of exogenous parameters $b$ (e.g., network topology). Note that $F_l$ can depend the decision vectors and endogenous variables of past time intervals. The objective function as formulated in Eq. (4.1) tries to capture the temporal dependencies between time intervals, such that the decision variables of an earlier time can have an impact on subsequent time intervals. This encourages the selection of solutions that account for the temporal dependencies between time intervals.

Due to the heavy computational burden of having to evaluate a simulation every time we want an estimate of the objective function in Eq. (4.1), we only have access to a small data set of observations when optimizing the objective function. Since we are considering high-dimensional problems, this means that we are working in the regime where we have more decision variables than observations. As a result, BO is chosen as the optimization framework, as it is known to be highly data-efficient (i.e. works well with limited data) and suitable for optimization problems with expensive-to-evaluate objective functions. A review of BO can be found in Section 3.1.1.

In the previous chapter, we showed that BO is capable of high-dimensional optimization for stationary problems (i.e. problems with no time-dependent decision variables) when combined with analytical models containing problem-specific prior information. This chapter extends the classes of problems that the BO framework of Chapter 3 is able to tackle to include offline dynamic optimization problems.

**Literature Review**

In the literature, dynamic transportation optimization problems are most commonly tackled with analytical models (e.g., Köster et al. 2018, Han et al. 2016, Lou, Yin, and Laval 2011). However, as mentioned in Section 3.1, traffic simulation models are better able to model the underlying traffic networks. Many SO methods have been proposed for tackling stationary transportation problems (see Section 3.1 for a review). Increasingly, work is also being done on SO for offline dynamic transportation problems, including OD matrix calibration and estimation (Ros-Roca, Montero,
and Barceló 2021, Osorio 2019), traffic signal control (Baldi et al. 2019, Chong and Osorio 2018) and toll optimization (Gu, Waller, and Saberi 2019, Zhang et al. 2019). However, most of the past work on dynamic transportation SO problems tend to treat the simulator as a black box – they do not embed information about the underlying problem dynamics when solving the optimization problem, other than using simulation observations which are time-dependent (Ros-Roca, Montero, and Barceló 2021, Baldi et al. 2019, Gu, Waller, and Saberi 2019, Zhang et al. 2019). Metamodel-based SO approaches, which incorporate problem-specific prior information, have been proposed for tackling dynamic transportation problems (Osorio 2019, Chong and Osorio 2018). However, the metamodel-based SO approaches do not explicitly balance exploration and exploitation. Furthermore, the chosen metamodel in Osorio (2019) is time-independent, and instead relies on the temporally correlated simulation observations to capture the temporal dependencies between time intervals.

Recently, work has also been done to tackle general dynamic optimization problems using BO (Nyikosa, Osborne, and Roberts 2018, Baheri et al. 2017, Bogunovic, Scarlett, and Cevher 2016, Marchant et al. 2014). Most of these efforts focus on sequential decision (i.e. online) problems. While these methods are not directly applicable to an offline dynamic transportation optimization problem, they do highlight the need for spatiotemporal GP priors when modeling a time-dependent objective function (Nyikosa, Osborne, and Roberts 2018, Bogunovic, Scarlett, and Cevher 2016, Marchant et al. 2014, Roberts et al. 2013, Wang, Fleet, and Hertzmann 2005). This means that the chosen covariance function has to be able to capture both spatial and temporal covariance between points in the feasible region. A popular method is the contextual bandit (Krause and Ong 2011), where the context refers to endogenous state information (e.g. time, state of network). For dynamic BO problems, the context can be chosen to be time (Nyikosa, Osborne, and Roberts 2018, Baheri et al. 2017). However, in many cases, the covariance function is chosen such that the spatial and temporal covariances are separable (e.g. Marchant et al. (2014)). Separable spatiotemporal covariance functions do not model space-time interactions (Singh et al. 2010, Cressie and Huang 1999). A separable spatiotemporal covariance functions im-
plies that the temporal component of the covariance between two points in the feasible region has the same functional shape, regardless of the relative displacements of the spatial components of the two points (Cressie and Huang 1999). Hence, to properly model a time-dependent objective function, it is necessary to use a non-separable spatiotemporal covariance function. Singh et al. (2010) provides a way to construct non-separable covariance functions using the spectral density, as obtained through Fourier transform. However, this requires additional simulation observations, which may be undesirable when working with computationally expensive simulators.

Another possible method to construct a non-separable spatiotemporal covariance function involves incorporating a dynamic analytical surrogate model of the objective function. The dynamic analytical surrogate model can be constructed by incorporating problem-specific prior information. For instance, in the field of transportation, traffic flow models can provide some insights into the traffic dynamics in the network, which can be used to obtain an analytical approximation of the objective function. In the previous chapter, we showed that incorporating a stationary queueing network model (Osorio and Bierlaire 2009) approximating expected trip travel time in the covariance function can lead to improvements in the solutions found by BO for a stationary high-dimensional problem. By extension, placing a dynamic analytical model in the covariance would result in a non-separable spatiotemporal covariance function that could help BO identify good solutions for a high-dimensional dynamic optimization problem.

When selecting a dynamic analytical model for use in the GP prior (i.e. prior mean function and covariance function), it has to be computationally efficient as used to solve an analytical optimization at every iteration of the BO approach. The role of the analytical model is to approximately describe traffic dynamics. It need not be highly accurate since it is used to define the GP prior of the BO approach and hence is combined with detailed dynamic simulation information within the BO approach. In this chapter, we focus on modelling the underlying road network of a given dynamic transportation optimization problem. Traffic flow models are able to capture some of the traffic dynamics in the network. In particular, macroscopic models hold the most
potential for use in the GP prior due to their computational efficiency, even though accuracy is sacrificed when compared to microscopic or mesoscopic models. Many different types of macroscopic traffic models for urban traffic networks have been proposed (see Section 2.2.2 of Li (2016) for a review), including a revised Lighthill-Whitham-Richards (LWR) model (Lighthill and Whitham 1955, Richards 1956, Jiang et al. 2019), as well as extensions to the cell transmission model (Daganzo 1994, 1995) to better model signalized intersections and urban traffic networks (Adacher and Tiriolo 2018, Hao, Boel, and Li 2018, Zhang, Wolshon, and Dixit 2015, Tiriolo, Adacher, and Cipriani 2014, Flötteröd and Rohde 2011). Link transmission models (Lu and Osorio 2018, Osorio and Flötteröd 2015) have also been used to model link boundary conditions in the network setting. Other dynamic macroscopic models for urban traffic networks include the model of Lin et al. (2012), and a model based on nonlinear partial differential equations (Thonhofer et al. 2018). However, many of the existing models were proposed with accuracy and consistency with traffic flow theory as their main goal. As a result, many of them are suitable for network analysis, but not sufficiently computationally efficient for use in the GP prior when working with a large-scale network.

This chapter proposes a dynamic analytical traffic network model that is able to model large-scale traffic networks, while remaining computationally efficient for use in the GP prior for addressing high-dimensional dynamic SO problems. The proposed analytical model has an arithmetic (resp. memory) complexity that is quadratic (resp. linear) in the number of links in the network, and is independent of the link space capacities. Hence, the model is suitable for use with large-scale networks.

In the following section, we formulate the dynamic analytical traffic network model, and discuss how the BO framework is extended to tackle dynamic optimization problems of the form in Eq. (4.1). We then validate the dynamic analytical traffic network model in Section 4.3. In Section 4.4, we test the use of the dynamic analytical traffic model in BO in a case study of a synthetic road network under different demand scenarios. The dynamic analytical traffic network model is then used in conjunction with BO to address a traffic signal control problem for Midtown
Manhattan, with the results of the case study presented in Section 4.4.3. Lastly, our conclusions are provided in Section 4.5.

4.2 Methodology

The analytical surrogate model used to specify the GP prior of the BO approach is a dynamic analytical traffic model, which is formulated in Section 4.2.1. In Section 4.2.3, we describe how the time-dependent SO problem is then framed for use within the BO framework.

4.2.1 Dynamic Analytical Traffic Model

In the case of the dynamic, time-dependent problem, the analytical model capturing the problem-specific prior information should provide information on the temporal dynamics as well, as this would allow the temporal correlation between variables to be modeled. For instance, in the case of traffic signal optimization where we are searching for the best set of signal plans for multiple time intervals, the performance of a signal plan for a given time interval would depend on the choice of signal plan for the previous time interval.

The proposed analytical model is a dynamic traffic network model that focuses on achieving computational efficiency and scalability, while maintaining accuracy as much as possible. In the proposed model, each link (defined here as a lane in a road section) in the network is modeled as a reservoir of vehicles, with inflow from upstream links and outflow to downstream links. A summary of the notation used is given below, and also in Appendix C.1:

\[
\begin{align*}
Q_{ij,l} & \quad \text{flow from link } i \text{ to link } j \text{ in time interval } l; \\
Q_{in,i,l} & \quad \text{inflow to link } i \text{ in time interval } l; \\
Q_{out,i,l} & \quad \text{outflow to link } i \text{ in time interval } l; \\
Q_{FD,i,l} & \quad \text{flow (as a function of vehicle density) as determined by a fundamental diagram; } \\
\gamma_{i,l} & \quad \text{external arrival rate to link } i \text{ in time interval } l; \\
\mu_{i,l} & \quad \text{service rate of link } i \text{ in time interval } l; \\
\rho_{i,l} & \quad \text{vehicle density of link } i \text{ in time interval } l;
\end{align*}
\]
maximum vehicle density (i.e. jam density); s
saturation flow rate; p_{ij}
probability of vehicles turning from link i to link j; d_i
length of link i; \Delta t
time step length; M
number of time steps in an interval; N_{i,l}^{DQ}
counter for number of vehicle that have reached the downstream end of link i in time interval l; t_i^{FF}
free-flow travel time of link i; \mathcal{D}_i
set of downstream links of link i; \mathcal{U}_i
set of upstream links of link i; \mathcal{L}
set of all links.

The proposed dynamic analytical traffic model in time interval l at time t = 0, \Delta t, \ldots, (M - 1)\Delta t is given by the following system of equations:

\begin{align*}
Q_{ij,l}(t) &= \frac{\mu_{i,l}(t)}{s} \cdot Q^{FD}(\rho_{i,l}(t)) \cdot 1(N_{i,l}^{DQ}(t) > 0) \cdot p_{ij} \left(1 - \frac{\rho_{i,l}(t)}{\rho_{\text{max}}}ight), \quad (4.2a) \\
Q_{i,l}^{\text{in}}(t) &= \gamma_{i,l} \left(1 - \frac{\rho_{i,l}(t)}{\rho_{\text{max}}}ight) + \sum_j Q_{j,i,l}(t), \quad (4.2b) \\
Q_{i,l}^{\text{out}}(t) &= \sum_j Q_{j,i,l}(t) + Q^{FD}(\rho_{i,l}(t)) \cdot 1(N_{i,l}^{DQ}(t) > 0) \cdot \left(1 - \sum_j p_{ij}\right), \quad (4.2c) \\
N_{i,l}^{DQ}(t) &= N_{i,l}^{DQ}(t - \Delta t) + \left[Q_{i,l}^{\text{in}}(t - t_i^{FF}) \left(1 + \frac{\rho_{i,l}(t)}{\rho_{\text{max}}}ight) - Q_{i,l}^{\text{out}}(t)\right] \cdot \Delta t, \quad (4.2d) \\
\rho_{i,l}(t + \Delta t) &= \rho_{i,l}(t) + \frac{1}{d_i} \left(Q_{i,l}^{\text{in}}(t) - Q_{i,l}^{\text{out}}(t)\right) \Delta t. \quad (4.2e)
\end{align*}

Eq (4.2a) describes the flow from link i to a downstream link j at a given time t \in [0, (M - 1)\Delta t] in time interval l. The expression (4.2a) can be broken down as follows. The service rate \mu_{i,l} is divided by the saturation flow rate s to provide the fraction of the saturation flow rate that can flow out of the link at a given time t. This service rate fraction is used to model the traffic light controlling traffic on that link i. The maximum outflow of the link i is determined by \( Q^{FD}(\cdot) \), which represents the flow as a function of vehicle density, as determined by a fundamental diagram. In this work, we make use of a triangular fundamental diagram (Treiber and Kesting 2013, Section 8.5). The indicator function 1(N_{i,l}^{DQ}(t) > 0) implies that flow from link i
to a downstream link $j$ can only be non-zero at time $t$ if there are vehicles at the downstream end of link $i$ at time $t$. This helps to account for link travel time in the model. The proportion of flow going from link $i$ to link $j$ is determined by the turning probabilities $p_{ij}$. Finally, the term $\left(1 - \frac{\rho_{i,l}(t)}{\rho_{\text{max}}}ight)$ acts to inversely scale the amount of outflow from link $i$ to link $j$ by the density on link $j$, so as to model spillback when the downstream link is congested. As a result, when $\rho_{j,i}(t)$ is large, $Q_{i,j,l}(t)$ will be small. As seen from Eq. (4.2a), the flows within the network can computed based only on the link densities $\rho_{i,l}$ and the counters $N_{i,l}^{DQ}$, thus helping with computational efficiency while staying frugal in memory requirements.

The inflows $Q_{i,l}^{\text{in}}$ to and outflows $Q_{i,l}^{\text{out}}$ from each link $i$ at time $t$ in time interval $l$ are defined in Eq. (4.2b) and (4.2c) respectively. In Eq. (4.2b), the inflow to a link $i$ is the sum of the external arrival rate to link $i$ and the flows from all upstream links of link $i$ to link $i$. Note that both flow and external arrival rate have units of vehicles per hour. Similarly, the outflow from a link $i$, as described in Eq. (4.2c), is the sum of flows from link $i$ to all downstream links of link $i$ and the flow leaving the network from link $i$. Note that the sum $\sum_{j \in D_i} p_{ij}$ in Eq. (4.2c) represents the total proportion of flow out of link $i$ into downstream links of link $i$. However, the sum is not necessarily equal to 1. Hence, the remainder (i.e. $1 - \sum_{j \in D_i} p_{ij}$) represents the proportion of flow leaving the network from link $i$. The flow leaving the network from link $i$ is also controlled by $1(N_{i,l}^{DQ}(t) > 0)$, such that the flow leaving the network from link $i$ can only be non-zero when there are vehicles at the downstream end of link $i$ to account for link travel time.

The counter $N_{i,l}^{DQ}$ is defined in Eq. (4.2d). It approximates the number of vehicles at the downstream end of the link by adding the number of vehicles that enter the downstream end of the link and subtracting the number of vehicles that leave the link at each time step, based on the inflow $Q_{i,l}^{\text{in}}$ and outflow $Q_{i,l}^{\text{out}}$ respectively. The discrete time approximation is used as there is no closed-form expression for the exact sum over time. The inflow to the downstream end of the link is approximated by the inflow to the (upstream end of the) link which occurred the link travel time before the current time. The link travel time is approximated by the free-flow link travel time.
scaled by the factor \( 1 + \frac{\rho_{i,l}(t)}{\rho_{\text{max}}} \), such that the link travel time is longer when there are more vehicles (i.e. higher density) on that link. Under free-flow conditions, the link travel time approximation should be a fairly good approximation. Under congested conditions, the approximation may start to break down. The inclusion of the counter \( N_{i,l}^{\text{DQ}} \) in the model (4.2) helps to prevent an instantaneous inflow of vehicles at the upstream end of an empty link from causing instantaneous outflow from that link. Hence, it accounts for non-instantaneous travel time.

The vehicle density of link \( i \) is updated at every time step according to Eq. (4.2e). A discrete sum over time steps is used as there is no closed-form expression for the exact sum (i.e. integral over time \( t \)). The second term represents the approximate change in density from time step \( t \) to \( t + \Delta t \). Note that the initial link density in time interval \( l \) corresponds to the link density at the end of time interval \( l - 1 \) (i.e. \( \rho_{i,l}(0) = \rho_{i,l-1}(M\Delta t) \) for \( l \geq 2 \).

The triangular fundamental diagram (Treiber and Kesting 2013, Section 8.5) chosen to represent \( Q_{\text{FD}} \) in Eq. (4.2) is formulated as such:

\[
Q_{\text{FD}}(\rho_{i,l}(t)) = \begin{cases} 
V_0 \rho_{i,l}(t) & \text{if } \rho_{i,l}(t) \leq \rho_C, \\
\frac{1}{T_G} \left(1 - \frac{\rho_{i,l}(t)}{\rho_{\text{max}}}\right) & \text{if } \rho_C < \rho_{i,l}(t) \leq \rho_{\text{max}}.
\end{cases}
\]  
(4.3a)

\[
\rho_C = \frac{1}{V_0 T_G + (1/\rho_{\text{max}})}.
\]  
(4.3b)

The additional model parameters in the triangular fundamental diagram are the desired speed \( V_0 \) and the time gap between vehicles \( T_G \). The congestion density is represented by \( \rho_C \), and is defined in Eq. (4.3b). Eq. (4.3a) shows how the flow is related to the vehicle density of the link when there is free-flowing traffic (i.e. \( \rho_{i,l}(t) \leq \rho_C \)) and when the link is congested (i.e. \( \rho_{i,l}(t) > \rho_C \)). The triangular fundamental diagram was chosen to represent \( Q_{\text{FD}} \) due to its simplicity. Yet, it is able to describe flow under free-flow and congested conditions. While the triangular fundamental diagram is able to describe phenomena such as moving jam fronts and their velocity, we do not explicitly model this in Eq. (4.2). For a more in-depth discussion about the triangular fundamental diagram, we refer the reader to Treiber
and Kesting (2013, Section 8.5).

In the model (Eq. (4.2)), the exogenous variables are $\mu_{i,l}$, $s$, $p_{ij}$ and $\rho^{\max}$. All the other variables are considered endogenous. In this model, traffic assignment is assumed to be exogenous, hence the turning probabilities $p_{ij}$ are exogenous. In the case studies, when this model is used to address signal control problems, $\mu_{i,l}$ becomes endogenous. In those cases, $\mu_{i,l}$ is a function of the green splits $x_l$ in time interval $l$ (see Eq. (4.8)).

### 4.2.2 Objective Function Example

With the time-dependent link densities $\rho_{i,l}(t)$, it is possible to obtain an approximation to the objective function that is being optimized. The analytical approximation, denoted $f^A$, of the simulation-based objective function ($f$ of Eq. (4.1)) can be written as the weighted mean of the analytical model approximation for each time interval $f_l^A$. In other words,

$$f^A = \sum_{l=1}^{L} \alpha_l f_l^A,$$

(4.4)

where $\alpha_l$ is the weight coefficient assigned to time interval $l$, and is the same as $\alpha_l$ as defined in Eq. (4.1).

For instance, in the case study of Section 4.4, we consider a dynamic traffic signal control problem, where the objective is to jointly minimize the expected travel time for multiple time intervals. Hence, $\mathbb{E}[F_l]$ of Eq. (4.1) represents the expected travel time during time interval $l$. In this case, the objective function approximation for each time interval $f_l^A$ would then be the expected travel time for each time interval $l$. The expected travel time for time interval $l$ can be approximated by applying Little’s
law (Little 2011, 1961) to the entire network for time interval $l$:

$$f_{l}^{A} = \frac{\frac{1}{M} \sum_{i \in \mathcal{L}} d_{i} \sum_{m=0}^{M-1} \rho_{i,l}(m\Delta t)}{\frac{1}{M} \sum_{i \in \mathcal{L}} \sum_{m=0}^{M-1} \gamma_{i,l} \left(1 - \frac{\rho_{i,l}(m\Delta t)}{\rho_{\text{max}}}ight)}$$  \hfill (4.5)

$$= \frac{\sum_{i \in \mathcal{L}} d_{i} \sum_{m=0}^{M-1} \rho_{i,l}(m\Delta t)}{\sum_{i \in \mathcal{L}} \sum_{m=0}^{M-1} \gamma_{i,l} \left(1 - \frac{\rho_{i,l}(m\Delta t)}{\rho_{\text{max}}}ight)}. \hfill (4.6)$$

The numerator in Eq. (4.5) represents the time-averaged number of vehicles in the whole network in time interval $l$, while the denominator is the time-average effective external arrival rate to the network during time interval $l$. The ratio of these two quantities is the expected time spent in the network in time interval $l$ according to Little’s law.

### 4.2.3 Dynamic Bayesian Optimization

The BO framework employed in Chapter 3 was used to solve time-independent optimization problems. However, it can be extended to tackle dynamic, time-dependent optimization problems of the form in Eq. (4.1). In order to do so, we first have to frame the dynamic problem in a suitable form for BO. This is done by jointly solving for the decision variables for all time intervals (i.e. $x_{1}, \ldots, x_{L}$). In other words, the decision variables for all time intervals are taken simultaneously as the input to the simulation-based objective function; BO then solves for the decision variables for all time intervals concurrently. For a problem considering $L$ time intervals, this means that the dimensionality of the optimization problem would be $L$ times that of the stationary problem.

The increase in the dimensionality of the optimization problem points to the increased importance of problem-specific prior information when attempting to solve the dynamic problem efficiently. In Chapter 3, a time-independent analytical queueing network model was used to provide problem-specific prior information on the
covariance between points in the time-independent feasible region. For the dynamic problem, it is essential to include temporal dependencies in the analytical model. Hence, we make use of the dynamic analytical traffic model (Eq. (4.2)), which is able to capture temporal correlations between points, to provide the problem-specific prior information. As with Chapter 3, the dynamic analytical traffic model is incorporated into the GP model by using the modified squared exponential covariance function of Eq. (3.16).

4.3 Validation

In this section, we first introduce the toy networks used for the purpose of validating the proposed dynamic analytical traffic model in Section 4.3.1. The validation results are presented in Section 4.3.2.

4.3.1 Toy Networks

For the purpose of validation, we make use of toy networks consisting of links connected in tandem to analyze the temporal dynamics of the vehicle densities $\rho_{i,t}(t)$ for each link $i$. More specifically, we consider two toy networks – a 2-link network and a 4-link network. Illustrations of the toy networks are shown in Figure 4-1.

The 2-link network (Figure 4-1a) consists of two links (single lanes) connected in tandem, with a traffic light between links 1 and 2. The traffic signal is set up such...
that each cycle is 60s, with a green time of 36s. This 2-link network provides insights into how vehicle density flows from an upstream link to a downstream link according to the dynamic analytical traffic model, in the presence of a traffic light.

Similarly, the 4-link network (Figure 4-1b) consists of four links (single lanes) connected in tandem. As shown in Figure 4-1b, there are traffic lights between links 2 and 3, as well as between links 3 and 4. The first traffic signal (between links 2 and 3) has a cycle time of 60s and a green time of 36s. The second traffic signal (between links 3 and 4) has a cycle time of 60s and a green time of 48s. An offset value of zero is used between the start of the green signals of each traffic signal. Note that there is no traffic light between links 1 and 2. This can occur in scenarios where longer links may be modeled as multiple short links. Hence, this allows us to see if the analytical model provides good estimates of the flow of vehicle density across the links with no traffic light. The larger 4-link network also provides insight on how estimation errors can build up as the network gets larger.

The dynamic analytical traffic model is set up using the parameter values as shown in Table 4.2. The parameters are defined in the units as shown in the parentheses of the first column of Table 4.2. The chosen maximum vehicle density $\rho_{\text{max}}$ of 200veh/km corresponds to an effective vehicle length of 5m (i.e. $\frac{1}{\rho_{\text{max}}}$), which is typical of the inter-vehicle distance for an average passenger car. The time step $\Delta t$ is chosen to be 1.0s, as it provides a good trade-off between computational efficiency and accuracy of the model. The saturation flow rate $s$ of 1800veh/h per lane is within the common range of values (1500 - 2000veh/h) as stated in the Highway Capacity Manual (Transportation Research Board 2000, page 8-26).

For the validation of the dynamic analytical traffic model, we consider a single time interval (i.e. $L = 1$) with 3 different demand scenarios. Each demand scenario is defined by the external arrival rate to link 1 $\gamma_{1,1}$ for each toy network. The 3 chosen values of $\gamma_{1,1}$ are shown in Table 4.2, and represent a low and a high level of congestion in the network respectively.

We also consider 3 different link lengths. In each case, all of the links in the network have the same length. The choice of link lengths being considered is motivated by the
Table 4.2: Dynamic analytical traffic model parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum vehicle density, $\rho_{\text{max}}$ (veh/km)</td>
<td>200</td>
</tr>
<tr>
<td>Desired speed, $V_0$ (km/h)</td>
<td>50</td>
</tr>
<tr>
<td>Time gap, $T_G$ (s)</td>
<td>1.2</td>
</tr>
<tr>
<td>Time step, $\Delta t$ (s)</td>
<td>1.0</td>
</tr>
<tr>
<td>Saturation flow rate, $s$ (veh/h)</td>
<td>1800</td>
</tr>
<tr>
<td>Demand, $\gamma_{1,1}$ (veh/h)</td>
<td>{360, 720}</td>
</tr>
<tr>
<td>Link lengths, $d_i$ (m)</td>
<td>{50, 100, 200}</td>
</tr>
</tbody>
</table>

Figure 4-2: Empirical cumulative distribution function illustrating the distribution of link lengths in the Midtown Manhattan network.

distribution of link lengths in the Midtown Manhattan network used in the case study (Section 4.4.3). This distribution of link lengths is plotted as an empirical cumulative distribution function (ecdf) in Figure 4-2. The ecdf represents the fraction of links in the network with a corresponding length or less. Based on Figure 4-2, more than 60% of the links have lengths of 50m or less. Around 35% of the links are between 50m and 150m long, while the rest of the links (around 5%) are between 150m and 200m in length.

4.3.2 Validation Results

We validate the dynamic analytical traffic model by comparing the vehicle density of each link at every second to the mean vehicle densities of 100 microscopic simulation
replications. The microscopic simulations used for validation are implemented using the PTV Vissim software (PTV AG 2020). For each simulation, an empty network is used as the initial state, and a warm-up period of 2 minutes is provided. Observations are then collected every second for the next 8 minutes. This is sufficient time to observe multiple traffic signal cycles. If there are high levels of congestion, this would also allow for observations of the congestion build-up in the networks. A similar set-up is used when evaluating the dynamic analytical traffic model.

Tables 4.3 and 4.4 provide simulation-based statistics that describe the level of congestion in the 2-link and 4-link networks respectively. More specifically, the tables show the time-average vehicle densities and peak vehicle densities for each link for different link lengths and demand. The vehicle density values are computed based on the mean of 100 simulation runs. From Tables 4.3 and 4.4, it can be seen that the scenario with the most congestion is the case of the 50m links and 720veh/h demand. In this scenario, the highest peak vehicle density occurs at the link with the first traffic light (i.e. 101.34veh/h on link 1 for the 2-link network, and 100.77veh/h on link 2 for the 4-link network). This represents a significant level of congestion. Conversely, the scenario with the least congestion is the case of the 200m links and 360veh/h demand.

Figures 4-3 and 4-4 show some examples of the dynamic analytical traffic model capturing the dynamics of the link vehicle densities in the 2-link and 4-link networks respectively, during the 8 minutes evaluation period. Each figure plots the vehicle density (y-axis) for each link in the network as a function of time (x-axis). For each link, the mean density of 100 microscopic simulations is represented by the black dotted line, while the shaded region represents the 95% confidence interval of the vehicle density on that link. The red solid line depicts the vehicle density predictions of the dynamic analytical traffic model for each link.

Figure 4-3 illustrates the dynamics for the two-link network using the 100m links with 720veh/h demand scenario. In this example, it can be seen that the vehicle densities as predicted by the dynamic analytical traffic model matches well with the vehicle densities as evaluated by microscopic simulation. This is true for both links 1
### Table 4.3: Simulation-based Level of Congestion Statistics (2-link Network)

<table>
<thead>
<tr>
<th>Link Length (m)</th>
<th>Demand (veh/h)</th>
<th>Time-Average Vehicle Density (veh/km)</th>
<th>Peak Vehicle Density (veh/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Link 1</td>
<td>Link 2</td>
</tr>
<tr>
<td>50</td>
<td>360</td>
<td>17.70</td>
<td>7.93</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>38.71</td>
<td>15.10</td>
</tr>
<tr>
<td>100</td>
<td>360</td>
<td>12.06</td>
<td>7.14</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>28.50</td>
<td>14.63</td>
</tr>
<tr>
<td>200</td>
<td>360</td>
<td>9.84</td>
<td>7.24</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>21.39</td>
<td>14.46</td>
</tr>
</tbody>
</table>

### Table 4.4: Simulation-based Level of Congestion Statistics (4-link Network)

<table>
<thead>
<tr>
<th>Link Length (m)</th>
<th>Demand (veh/h)</th>
<th>Time-Average Vehicle Density (veh/km)</th>
<th>Peak Vehicle Density (veh/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Link 1</td>
<td>Link 2</td>
</tr>
<tr>
<td>50</td>
<td>360</td>
<td>8.00</td>
<td>16.96</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>18.44</td>
<td>40.31</td>
</tr>
<tr>
<td>100</td>
<td>360</td>
<td>6.95</td>
<td>12.75</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>14.24</td>
<td>28.86</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>13.94</td>
<td>21.49</td>
</tr>
</tbody>
</table>
and 2. In particular, the dynamic analytical traffic model is able to capture the build-up of vehicle density in link 1 (Figure 4-3a) when the traffic signal is red (e.g. from 36s to 60s and periodically). When the traffic signal turns green (e.g. at every 60s), the vehicle density in link 1 drops, before leveling off as the outflow from link 1 starts to match the inflow. Similarly, the analytical model is able to capture the dynamics of link 2. When the traffic signal turns green (e.g. at every 60s), the vehicle density in link 2 (Figure 4-3b) increases sharply as vehicles flow into link 2 from link 1. The vehicle density in link 2 then starts dropping as vehicles reach the end of the link and exit the network.

In the case of the 4-link network (Figure 4-4), the example shown is from the 50-m links with 720veh/h demand scenario. This corresponds to the scenario with the most congestion out of all the scenarios tested. In this example, the dynamic analytical traffic model shows that it is able to capture most of the vehicle density dynamics. The analytical model predicts similar peak shapes as the microscopic simulations, although the magnitudes of the peaks as predicted by the dynamic analytical traffic model may be smaller than that obtained by simulation (Figures 4-4a, 4-4c and 4-4d). This can be due to the fact that the dynamic analytical traffic model relies on length-averaged link vehicle density to compute flow into and out of the link, even though there may not be an even distribution of vehicles along the link. While the
use of the counter $N_{i,l}^{DQ}$ (number of vehicle that have “built up" at the downstream end of the link) helps to alleviate this issue by accounting of the time needed to travel the length of link, it does not completely solve the problem. As a result, the outflow (resp. inflow) may be larger (resp. smaller) than expected, leading to smaller peaks in the vehicle density on that link.

We further validate the dynamic analytical traffic model by looking at the mean absolute error (MAE) and normalized MAE of the vehicle density for each lane in the networks as estimated by the dynamic analytical traffic model. The MAE is computed by considering the absolute difference between the vehicle density of the dynamic analytical traffic model and the simulation-based vehicle density for all links at every time step. The simulation-based vehicle density is obtained by taking the
mean of 100 simulation runs. The lane-specific MAE is then obtained by taking the mean over all time steps. The lane-specific normalized MAE is computed by multiplying the respective MAE by the link length (i.e. normalized by link length). Hence, the normalized MAE represents the absolute difference in number of vehicles on the link as estimated by the dynamic analytical traffic model and the simulations. The MAE and normalized MAE values for the 2-link and 4-link networks are presented in Tables 4.5 and 4.6 respectively.

As seen from both Tables 4.5 and 4.6, the MAE is generally larger when demand is greater for both the 2-link and 4-link networks. This suggests that the dynamic analytical traffic model can become less accurate at higher levels of congestion. It can also be seen that the normalized MAE values in Tables 4.5 and 4.6 for the 200m links for a given demand scenario are greater when compared to the same demand scenario for the 50m and 100m links. This indicates that the accuracy of the dynamic analytical traffic model may decrease if longer link lengths are used. A reason for this is that longer links are more likely to have an uneven distribution of vehicles along the link. Since the vehicle density is computed by averaging the number of vehicles along the length of the link, this can lead to inaccuracies when computing the link outflow, as well as blocking at the upstream end of the link. Hence, the modeling accuracy of the link dynamics may decrease with longer links. On the other hand, the normalized MAE values for all link lengths and demand scenarios are generally less than 1. In fact, the normalized MAE for the 50m and 100m links cases are all less than 0.5 vehicles, including the highly congested link 1 in the 2-link network, and link 2 in the 4-link network (as seen from Tables 4.3 and 4.4 respectively). Hence, this shows that the difference in number of vehicles on each link as predicted by the dynamic analytical traffic model and the simulations is small.

4.4 Case Study: Traffic Signal Optimization

In this section, we refer to the use of the dynamic analytical traffic model as a surrogate model in BO as dynamic BO. We test dynamic BO on a synthetic network,
### Table 4.5: Mean Absolute Errors (2-link Network)

<table>
<thead>
<tr>
<th>Link Length (m)</th>
<th>Demand (veh/h)</th>
<th>MAE (veh/km)</th>
<th>Normalized MAE (veh)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Link 1</td>
<td>Link 2</td>
</tr>
<tr>
<td>50</td>
<td>360</td>
<td>4.47</td>
<td>2.46</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>7.95</td>
<td>4.37</td>
</tr>
<tr>
<td>100</td>
<td>360</td>
<td>2.35</td>
<td>1.93</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>4.45</td>
<td>2.73</td>
</tr>
<tr>
<td>200</td>
<td>360</td>
<td>2.78</td>
<td>2.44</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>3.69</td>
<td>3.32</td>
</tr>
</tbody>
</table>

### Table 4.6: Mean Absolute Errors (4-link Network)

<table>
<thead>
<tr>
<th>Link Length (m)</th>
<th>Demand (veh/h)</th>
<th>MAE (veh/km)</th>
<th>Normalized MAE (veh)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Link 1</td>
<td>Link 2</td>
</tr>
<tr>
<td>50</td>
<td>360</td>
<td>1.11</td>
<td>3.85</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>2.91</td>
<td>4.50</td>
</tr>
<tr>
<td>100</td>
<td>360</td>
<td>0.754</td>
<td>2.21</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>1.58</td>
<td>4.18</td>
</tr>
<tr>
<td>200</td>
<td>360</td>
<td>0.637</td>
<td>3.05</td>
</tr>
<tr>
<td></td>
<td>720</td>
<td>1.40</td>
<td>4.97</td>
</tr>
</tbody>
</table>
where we show that dynamic BO is competitive with other benchmark methods under different scenarios. The dynamic traffic signal optimization problem is described in Section 4.4.1. In Section 4.4.2, we introduce the synthetic arterial network used for the case study. We also explain the different scenarios considered and the benchmark methods used for comparison. The large-scale case study involving Midtown Manhattan is presented in Section 4.4.3.

4.4.1 Dynamic Traffic Signal Optimization Problem

We first define the dynamic traffic signal optimization problem, which is used to benchmark our proposed dynamic BO approach against other methods. This dynamic traffic signal optimization problem is also used in the large-scale New York City case study in Section 4.4.3. For a review of traffic signal control optimization terminology, we refer the reader to Osorio (2010, Appendix A, pages 119-121).

We consider fixed time signal control, where the signal plan is periodic, with a fixed cycle time (i.e. the time required to complete a sequence of signals). The signal plans are determined offline, and do not respond to real-time conditions in the traffic network. For the dynamic setting, the simulation period (e.g. 5pm - 6pm) is divided up into $L$ time intervals. The goal is then to determine a fixed time signal plan for each time interval. The decision variables are the green splits (i.e. normalized green times) of each signal phase of the intersections in the network, for each time interval during the simulation period. All the other control variables, such as the offsets, cycle times, stage structure, etc., are predetermined and kept constant. The notation used to formulate the dynamic traffic signal optimization problems summarized below and in Appendix C.1:

- $f$: SO objective function;
- $L$: total number of time intervals;
- $F_l$: random variable denoting the travel time of vehicles in the road network during time interval $l$;
- $F_{L=1}$: random variable denoting the travel time of vehicles in the road network during the second half of the simulation period (when $L = 1$);
- $x_{l,j}$: green split of signal phase $j$ in time interval $l$;
- $\mathbf{x}_l$: vector of green splits for time interval $l$;
vector of endogenous simulation variables for time interval $l$.

**Exogenous problem parameters:**
- $c_i$: cycle time of intersection $i$;
- $e_i$: fixed cycle time of intersection $i$;
- $x^{LB}$: vector of minimal green splits;
- $b$: vector of exogenous simulation parameters;
- $\mathcal{I}$: set of signal controlled intersection indices;
- $\mathcal{P}(i)$: set of signal phase indices of intersection $i$.

The problem is formulated as such:

$$
\min_{x_1, \ldots, x_L} f(x, z; b) \equiv \begin{cases} 
\mathbb{E}[F_{L=1}(x, z; b)], & \text{if } L = 1 \\
\frac{1}{[L/2]} \sum_{l=[(L/2)+1]}^{L} \mathbb{E}[F_l(x, z; b)], & \text{otherwise} 
\end{cases}
$$

subject to

$$
\sum_{j \in \mathcal{P}(i)} x_{l,j} = \frac{c_i - e_i}{c_i}, \quad \forall i \in \mathcal{I}, \ l = 1, \ldots, L, 
$$

and

$$
x_l \geq x^{LB}, \quad l = 1, \ldots, L.
$$

The objective function (Eq. (4.7a)) for this case study is the expected travel time of vehicles that end their trips during the second half of the simulation period. For instance, if we simulate from 5pm to 6pm, the objective function is the expected travel time during 5:30pm to 6pm. For $L > 1$, this is equivalent to the evaluation period defined by time intervals $l = [(L/2) + 1]$ to $l = L$. In these expressions, $[\cdot]$ denotes rounding down to the smaller integer and $\lceil \cdot \rceil$ denotes rounding up to the larger integer.

For this case study, we consider different values of $L \in \{1, 2, 4\}$. Hence the expression in Eq. (4.7a) allows us to directly compare the objective function values for $L = 1$ and $L = 2$, since they represent the travel time measured during the same time period.

This objective function was chosen to emphasize the effect of temporal dynamics in the traffic network. More specifically, when $L > 1$, there can be time intervals $l < [(L/2) + 1]$ (and hence signal plans) that are not within the evaluation period. However, they can have an effect on the way congestion builds up within the network, which can eventually affect the expected travel time during the evaluation period.

The endogenous simulation variables are denoted by $z$. They represent link-level and network-level metrics like travel times, densities, delays (which are a function of the
signal plans in the previous time intervals). The exogenous simulation parameters \( b \) includes the road network topology and fixed lane attributes (e.g. lane length, speed limits, grade).

The linear constraints Eq. (4.7b) represent the cycle time constraints. This set of constraints ensure that the sum of all the green splits for a given intersection add up to the proportion of available cycle time that can be optimized (i.e. not fixed). Eq. (4.7c) represents the lower bound of the green splits. The minimum green splits are normally determined by the local transportation authorities based on safety considerations.

In the dynamic analytical traffic model, the service rates for signal-controlled links \( \mu_{i,l} \) (Eq. (4.2a)) are functions of the green splits \( x_l \). For this case study, the green times computed from the green splits \( x_l \) and the cycle times are rounded to the nearest second. Since the time step \( \Delta t \) is 1.0s, the service rate \( \mu_{i,l}(m\Delta t) \) for a signal-controlled link \( i \) at time step \( m\Delta t \) in time interval \( l \) can be defined as:

\[
\mu_{i,l}(m\Delta t) = \begin{cases} 
  s & \text{if the light is green,} \\
  0 & \text{otherwise,}
\end{cases} \quad l = 1, \ldots, L, \quad m = 1, \ldots, M. \quad (4.8)
\]

The service rates for non-signal-controlled links are assigned the saturation flow rate \( s \) for all time steps.

### 4.4.2 Synthetic Arterial Network

In this case study, we make use of a synthetic network (Osorio and Yamani 2017, Osorio and Wang 2017) which we refer to as the Synthetic Arterial Network (SAN). The SAN is depicted in Figure 4-5. It consists of 20 single-lane and 4 intersections, making up a main arterial (horizontal) and 4 side roads (vertical). The main arterial (denoted by AB) and the second side road from the left (denoted by CD) are bidirectional. The rest of the side roads are unidirectional. The travel demand is defined such that vehicles only have straight paths (i.e. no left or right turns at the intersections). As such, the signal plans of each intersections consists of only two phases.
(since there are no turning movements). Taking into account the linear cycle time equality constraint (Eq. (4.7b)), and $L$ time intervals, this would be a $4L$-dimensional problem.

For this case study, we optimize the signal plans for a 1-hour period, with a 15-minute warm-up period. As such, the objective function is the expected travel time of vehicles that end their trips in the last 30 minutes. The minimum green times for each signal phase in the SAN simulation model are set to 4 seconds. Hence, the elements of the vector $\mathbf{x}^{LB}$ (Eq. (4.7c)) are the ratio of 4 seconds to the cycle time of the corresponding intersection. The SAN simulation model is implemented using the Aimsun software (TSS-Transport Simulation Systems 2015, Aimsun Next Postgraduate License). For more implementation details, we refer the reader to Appendix C.2.

**Scenarios**

The purpose of performing a case study of the SAN is to identify if there are scenarios where dynamic BO may be better or worse than other BO methods. For this purpose, we consider the demand profile shown in Table 4.8. The first column of Table 4.8 denotes the origin-destination (OD) pair. The second column shows the demand profile used in the SAN. In this case, the demand for AB and BA drops significantly after 30 minutes, while the demand for CD and DC increases dramatically at the 30-
Table 4.8: Demand

<table>
<thead>
<tr>
<th>OD Pair</th>
<th>Demand (veh/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB/BA</td>
<td>1050 → 350</td>
</tr>
<tr>
<td>CD/DC</td>
<td>300 → 900</td>
</tr>
<tr>
<td>EF</td>
<td>100</td>
</tr>
<tr>
<td>HG</td>
<td>100</td>
</tr>
<tr>
<td>JK</td>
<td>100</td>
</tr>
</tbody>
</table>

minute mark. This results in the major direction (i.e. direction with more demand) at the second intersection switching at the 30-minute mark.

With this set-up, we consider the following scenarios:

- **Level of Demand**: We consider how the level of demand can affect the performance of the different BO methods. The values given in Table 4.8 represent the 100% case, and has a total time-averaged network demand of 2900 veh/h. For comparison, we evaluate the cases with 25%, 50%, 75%, 100% and 125% of the demand, where the demand values are scaled by the respective percentages.

- **Number of Time Intervals**: We also look whether having more time intervals with different signal plans can lead to better network performance. For each of the different levels of demand, we compare the cases with 2 and 4 time intervals (i.e. \( L = 2 \) and 4 respectively). These correspond to 8-D and 16-D problems respectively.

We also look at how the different BO methods respond to different demand profiles. The set-up and results are presented in Appendix C.3.

**Benchmark Methods**

The results of Chapter 3 show that having the analytical surrogate model in the prior mean function of the GP has a limited effect on optimization performance (see Figure 3-10 and Table 3.2). Hence, we chose to use a constant prior mean function (Eq. (3.14)) when performing BO on the dynamic problem. Instead, the analytical surrogate model is used in the covariance function of the GP. This is equivalent to the Proposed-Covariance GP prior of Chapter 3. Similar to Chapter 3, the acquisition
function used is the expected improvement (EI) acquisition function (Eq. (3.10)-(3.12)).

To evaluate the performance of using the dynamic analytical traffic model as the analytical surrogate model when performing BO for a dynamic problem, we compare several benchmark methods. These benchmark methods are used in the case study in Section 4.4.3 as well. The benchmark methods are as follows:

- **Vanilla BO**: Vanilla BO does not make use of any problem-specific prior information. The GP prior consists of the constant prior mean function (Eq. (3.14)), and the standard squared exponential covariance function (Eq. (3.3)). Vanilla BO corresponds to the Standard GP prior of Chapter 3.

- **Dynamic BO**: Dynamic BO uses the proposed dynamic analytical traffic model (Eq. (4.2)) as the analytical surrogate model in the analytical model-based covariance function (Eq. (3.16)). The prior mean function used in this GP prior is also the constant prior mean function (Eq. (3.14)). Hence, Dynamic BO corresponds to the Proposed-Covariance GP prior of Chapter 3, but with the dynamic analytical traffic model as the surrogate model.

- **Stationary BO**: Stationary BO uses the stationary traffic model (Eq. (3.27)) as the analytical surrogate model in the covariance function (Eq. (3.16)). Similar to the other methods, the prior mean function used is the constant prior mean function (Eq. (3.14)). Stationary BO corresponds exactly to the Proposed-Covariance GP prior of Chapter 3. Note that when working with scenarios with multiple time intervals (i.e. \( L > 1 \)), time-averaged values for the external arrival rates are used for evaluating the stationary traffic model. For each interval-specific signal plan \( \mathbf{x}_l \) for \( l = [(L/2) + 1], \ldots, L \), the interval-specific stationary traffic model estimate is computed. The stationary traffic model estimate of the objective function is then taken as the mean of the interval-specific stationary traffic model estimates.

For each BO method and scenario, we consider 3 different initial sets, each containing 4 initial points used to fit the GP posteriors at the beginning of the BO run.
These initial points are drawn from the feasible region uniformly at random using the code of Stafford (2006). Similar to Section 3.4, we compute the objective function estimate by taking the mean of 4 simulations, with 2 additional simulations of the best solution at the end of each optimization iteration. For each BO run, we consider a computational budget of 30 iterations in total (i.e. 26 optimization iterations).

Results

The optimization performance of the different BO methods under 2 and 4 time intervals are plotted in Figures 4-6 and 4-7 respectively. In each plot of Figures 4-6 and 4-7, the x-axis represents the optimization iteration (i.e. does not count the 4 initial observations). The y-axis shows the objective function estimate of the best solution at a given iteration. Each BO method is run 3 times for 3 different initial sets (i.e. 9 runs in total). Each line in each plot of Figure C-1 denotes the mean of the best solution (i.e. the solution with the smallest average travel time) at a given iteration for 9 BO runs. The shaded regions denote the values ±1 standard error away from the mean. The black dash-dotted line corresponds to Vanilla BO, the blue dashed line to Dynamic BO, and the green solid line to Stationary BO. In these plots, the smaller the y-value reached by the curve, the better the solution is. Similarly, the sooner the curve reaches a small y-value, the more efficient the BO method is deemed to be.

We also test the differences in the mean of the best solution proposed by the different BO methods for statistical significance. The results of the one-sided paired $t$-tests are shown in 4.9 for the different levels of demand and number of time intervals. Each $t$-test considers whether Dynamic BO performs better than Vanilla BO (resp. Stationary BO) for a given level of demand and number of time intervals. The null hypothesis assumes that Dynamic BO obtained a mean performance that is worse than or equal to the mean performance of Vanilla BO (resp. Stationary BO). The alternative hypothesis states that Dynamic BO obtained a mean performance that is better than Vanilla BO (resp. Stationary BO). Both the $t$-statistics and $p$-values are shown for each test. We consider each $t$-test at the 10% level of significance.
Figure 4-6: Mean performance at a given iteration for different levels of demand with 2 time intervals.
Figure 4-7: Mean performance at a given iteration for different levels of demand with 4 time intervals.
With 8 degrees of freedom, the corresponding critical value of the \( t \)-statistic is -1.397. The \( t \)-tests with \( t \)-statistics smaller than the critical value have their null hypotheses rejected, and are displayed in bold in Table 4.9.

We first consider the case of 2 time intervals (Figure 4-6). When using 2 time intervals, Dynamic BO is competitive against Vanilla BO and Stationary BO. Across most levels of demand (i.e. 50%, 75%, 100% and 125%), there is little difference between the relative performance of the best solutions obtained by the 3 BO methods. However, as seen from Table 4.9, Dynamic BO was statistically better than Vanilla BO at the 75% and 125% levels of demand. In Figure 4-6e, we also see that Dynamic BO was able to identify good solutions more quickly than Vanilla BO. This indicates the usefulness of the dynamic analytical traffic model in finding good solutions quickly. Furthermore, the 25% demand case (Figure 4-6a) shows that Dynamic BO is able to obtain solutions that are statistically better than those obtained by Stationary BO. This suggests that in conditions with less congestion, there are more temporal dynamics in the network which need to be accounted for. Conversely, a congested network can be modeled relatively well with a stationary analytical model.

In the case of 4 time intervals (Figure 4-7), we see that Dynamic BO identifies good solutions faster than Vanilla BO. This is especially true for the higher levels of demand (i.e. 50%, 75%, 100% and 125%). In addition, Dynamic BO outperforms Stationary BO at the end of the BO runs for all the different levels of demand considered. The differences in performance between Dynamic BO and Stationary BO, while small, are statistically significant at the 10% level of significance, as seen in Table 4.9. However,
across the different levels of demand, Figure 4-7 shows that there is no significant difference in the relative performance of the 3 BO methods – Dynamic BO does not perform much better for a given level of demand.

Comparing Figures 4-6 and 4-7, the results suggest that Dynamic BO might be more effective than Vanilla BO and Stationary BO as the number of time intervals increases. This could be due to the problem dimensionality increasing as the number of time intervals increase. With a higher problem dimensionality, the more accurate prior information provided by the dynamic analytical traffic model would then become more valuable for solving the optimization problem.

We also examine the average travel time for each OD pair in the SAN. The results are plotted in Figures 4-8 and 4-9 for the 2 and 4 time interval cases respectively. In each plot of Figures 4-8 and 4-9, the x-axis labels all the OD pairs of the network. Each bar represents the average travel time of vehicles which have completed their journeys for a given OD pair (see Figure 4-5). The average travel time is computed from the best solutions of 9 optimization runs (3 initial sets of 3 runs each). The error bars denote the 95% confidence interval. Within each group of bars, the grey (left) bar corresponds to Vanilla BO, the blue (center) bar to Dynamic BO, and the green (right) bar to Stationary BO.

We further tested if the differences in average travel time along the main arterial (i.e. OD pairs AB and BA) obtained by Dynamic BO and Stationary BO are statistically significant. This is done by performing a one-sided paired $t$-test based on the average travel time of the 9 optimization runs. The results of the $t$-tests are tabulated in Table 4.10 for the different levels of demand and number of time intervals. Each $t$-test considers whether Dynamic BO obtained a smaller travel time along AB and BA than Stationary BO. The null hypothesis assumes that Dynamic BO obtained a travel time that is worse than or equal to the travel time of Stationary BO, while the alternative hypothesis states that Dynamic BO obtained a travel time smaller than Stationary BO. Similar to Table 4.9, we consider each $t$-test at the 10% level of significance. The $t$-test which have their null hypotheses rejected are displayed in bold in Table 4.10.
Figure 4-8: Average travel time by OD pair for different levels of demand with 2 time intervals.
Figure 4-9: Average travel time by OD pair for different levels of demand with 4 time intervals.
Table 4.10: One-sided paired $t$-test results (average travel time by OD pair in SAN)

<table>
<thead>
<tr>
<th>Demand</th>
<th>AB</th>
<th>BA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L = 2</td>
<td>L = 4</td>
</tr>
<tr>
<td></td>
<td>$t$-stat</td>
<td>$p$-value</td>
</tr>
<tr>
<td>25%</td>
<td>-0.335</td>
<td>0.373</td>
</tr>
<tr>
<td>50%</td>
<td>0.275</td>
<td>0.605</td>
</tr>
<tr>
<td>75%</td>
<td>-1.165</td>
<td>0.139</td>
</tr>
<tr>
<td>100%</td>
<td>-2.163</td>
<td>0.0313</td>
</tr>
<tr>
<td>125%</td>
<td>-0.802</td>
<td>0.223</td>
</tr>
</tbody>
</table>

For the case of 2 time intervals (Figure 4-8), we see that Dynamic BO can find solutions that result in shorter travel times along the main arterial (i.e. OD pairs AB and BA). This is especially true as the level of demand increase. As seen from Table 4.10, the travel times along both AB and BA obtained by Dynamic BO is statistically smaller than those of Stationary BO at 100% level of demand. At 125% level of demand, the travel times along BA as obtained by Dynamic BO is also statistically better than that of Stationary BO. At the same time, the travel times along the side roads are not significantly increased compared to Vanilla BO and Stationary BO.

In the case of 4 time intervals (Figure 4-9), the reduction in travel times along the main arterial obtained by Dynamic BO compared to Vanilla BO and Stationary BO is even more pronounced. With the exception of the 25% level of demand, Dynamic BO finds solutions that have statistically smaller travel times along AB and BA compared to Stationary BO for all the other levels of demand (Table 4.10). This reinforces the notion that Dynamic BO is better able to find solutions with shorter travel times on the main arterial when the level of demand increases, compared to Stationary BO. Similar to the 2 time intervals case, the travel times along the side roads for Dynamic BO are not significantly longer than those for Vanilla BO and Stationary BO.

Comparing Figures 4-8 and 4-9, we see that the reduction in travel times along the main arterial for Dynamic BO is more noticeable when considering more time intervals. For instance, the travel time along BA for Dynamic BO is 55% shorter than that of Stationary BO for the 125% level of demand case with 4 time intervals.
(Figure 4-9e), compared to a 23% reduction in the 125% level of demand case with 2 time intervals (Figure 4-8e). This indicates the usefulness of the dynamic analytical traffic model in helping to identify more equitable solutions when more time intervals are involved. The longer travel times along the main arterial indicate that Vanilla BO and Stationary BO are identifying solutions that prioritize the side roads.

In general, Dynamic BO is competitive with Vanilla BO and Stationary BO. As the number of time intervals being considered increases, Dynamic BO is able to identify solutions better than Stationary BO. Across the different levels of demand, there is little difference in the relative performance of the solutions identified by the different BO methods. However, as the level of demand increases, Dynamic BO is able to find solutions with shorter travel times along the main arterial of the SAN, suggesting that Dynamic BO is better able to find more equitable solutions than the other two methods.

4.4.3 Case Study: Midtown Manhattan

In this section, we test the proposed Dynamic BO method in a large-scale case study involving a fixed time traffic signal optimization problem for Midtown Manhattan. We first describe the details of the implementation, followed by the results of the case study.

Implementation Details

In this case study, we compare how Dynamic BO fares against Vanilla BO and Stationary BO when applied to a high-dimensional dynamic traffic signal optimization problem involving the large-scale network of Midtown Manhattan (MTM) in New York City. The traffic signal optimization problem being solved is formulated in the same way as the SAN case study (see Section 4.4.1, Eq. (4.7a)-(4.7c)), where the objective function is also to minimize the expected travel time for vehicles that complete their trips in the evaluation period.

The MTM area being studied is demarcated by the rectangle in the map shown
in Figure 4-10. Note that the MTM network used here is the same network used for the case studies in Chapters 2 and 3. For this MTM case study, we simulate the peak hour of 5pm - 6pm. Specifically, we optimize the traffic signals for the peak hour of 5pm - 6pm. A 15-minute warm-up period is provided at the beginning of each simulation run. Note that this implementation differs slightly from Sections 2.4 and 3.4, where traffic is simulated from 3pm - 6pm, with the traffic signal plans being optimized for 5pm - 6pm.

As with Chapters 2 and 3, the MTM simulation model (Figure 4-11) is implemented using the Aimsun software (TSS-Transport Simulation Systems 2015, Aimsun Next Postgraduate License). The simulation model consists of 698 roads, 2756 lanes and 444 intersections. During the hour of 5pm - 6pm, the expected demand for the whole network is over 29,000 trips per hour, distributed across more than 3500 origin-destination pairs. The external arrival rates for the MTM network do not change during the simulated hour of 5pm - 6pm. However, due to congestion build-up in the network, it can still be useful to consider a dynamic traffic signal plan. The minimum green times for each signal phase is taken as 6 seconds. This means that the elements in the vector of minimum green splits $x^L_B$ (Eq. (4.7c)) are the ratio of 6 seconds to the cycle time of the corresponding intersection.

We compare the performance of the traffic signal plans proposed by the 3 different BO methods explained in Section 4.4.2. Taking into account the trade-off between the number of time intervals being considered and the problem dimensionality, we decide to tackle a dynamic problem with 2 time intervals (i.e. $L = 2$) for the Vanilla BO and Dynamic BO methods,. In other words, each time interval has a 30-minute duration. Given that we control a total of 97 signalized intersections with 259 green splits, this means that the dynamic problem corresponds to a 324-D problem (i.e. $2 \times (259-97) = 324$), after accounting for the linear cycle time equality constraint (Eq. (4.7b)). On the other hand, the Stationary BO method is used to tackle a stationary problem consisting of just 1 time interval of 1 hour (i.e. a 162-D problem). This ensures that Stationary BO is not placed at a disadvantage, since its analytical surrogate model (i.e. the stationary analytical model) does not account for temporal dynamics. It also
Figure 4-10: Map of Midtown Manhattan with the simulated area demarcated by a rectangle (MapQuest.com, Inc 2018).

Figure 4-11: Midtown Manhattan model in Aimsun.
allows us to see if solving a dynamic problem can lead to solutions that are better than solutions to the stationary problem. Note that both the stationary and dynamic problems are high-dimensional in the field of BO, where problems are usually limited to less than 10 dimensions (Wang et al. 2016, Kandasamy, Schneider, and Póczos 2015).

We consider 3 distinct initial sets for each BO method. Each initial set consists of 20 initial observations, which are drawn uniformly at random using the code of Stafford (2006). These initial observations are used to fit the GP posteriors at the start of the BO run. Since the dynamic and stationary problems are different, the initial sets used for Stationary BO is different from the initial sets used for the other 2 methods. This also ensures that Dynamic BO and Vanilla BO are not disadvantaged by using initial observations which consist of a single signal plan for both time intervals. For the BO runs, we use a computational budget of 60 iterations (i.e. 40 optimization iterations). Similar to the SAN case study, the objective function value is estimated by taking the mean of 4 simulations, with 2 additional simulations of the best solution at the end of each optimization iteration. Including the 80 simulation runs used to evaluate the 20 initial observations, the total number of simulation runs is then 320.

Furthermore, we also use an existing signal plan as a performance benchmark for the signal plans proposed by the different BO methods. This existing signal plan was previously used by the New York City Department of Transportation (NYCDOT) for the MTM area, and is known to perform well. For more implementation details, we refer the reader to Appendix C.2.

Results

The optimization performance of the 3 different BO methods are compared in Figure 4-12. Each plot in Figure 4-12 compares the optimization performance for a different initial set. Note that Vanilla BO and Dynamic BO have the same initial observations for each initial set, but the initial observations used for Stationary BO are different. This is due to the fact that Vanilla BO and Dynamic BO solve a dynamic problem with 2 time intervals (i.e. a 324-D problem), while Stationary BO solves a
stationary problem with 1 time interval (i.e. a 162-D problem).

In each plot of Figure 4-12, the x-axis represents the optimization iteration, while the y-axis denotes the expected travel time in the last 30 minutes of the simulated hour (objective function value). Each line depicts the mean of the best solution at a given iteration for 3 BO runs, with the shaded region representing the values ±1 standard error away from the mean. In addition, the mean travel time and its standard error (based on 50 simulation runs) of the existing signal plan is also illustrated in each plot by the thin red dashed line and the red shaded region.

To facilitate the interpretation of the results of Figure 4-12, we test the differences in the mean of the best signal plans proposed by the different BO methods for statistical significance. The results of the one-sided paired $t$-tests comparing the performance of the different BO methods are shown in Table 4.11. The difference in performance

Figure 4-12: Mean performance at a given iteration achieved by the different BO methods.
Table 4.11: One-sided paired $t$-test results (MTM optimization performance)

<table>
<thead>
<tr>
<th>Test</th>
<th>Initial Set 1</th>
<th>Initial Set 2</th>
<th>Initial Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t$-stat</td>
<td>p-value</td>
<td>$t$-stat</td>
</tr>
<tr>
<td>Dynamic BO vs. Vanilla BO</td>
<td>-1.376</td>
<td>0.151</td>
<td>-0.893</td>
</tr>
<tr>
<td>Stationary BO vs. Vanilla BO</td>
<td>-3.238</td>
<td>0.0418</td>
<td>-2.407</td>
</tr>
<tr>
<td>Stationary BO vs. Dynamic BO</td>
<td>-0.686</td>
<td>0.282</td>
<td>-1.201</td>
</tr>
</tbody>
</table>

Table 4.12: One-sided two-sample $t$-test results (MTM optimization performance)

<table>
<thead>
<tr>
<th>Test</th>
<th>Initial Set 1</th>
<th>Initial Set 2</th>
<th>Initial Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t$-stat</td>
<td>p-value</td>
<td>$t$-stat</td>
</tr>
<tr>
<td>Vanilla BO vs. Existing Plan</td>
<td>-0.567</td>
<td>0.287</td>
<td>-0.618</td>
</tr>
<tr>
<td>Dynamic BO vs. Existing Plan</td>
<td>-1.333</td>
<td>0.0941</td>
<td>-0.918</td>
</tr>
<tr>
<td>Stationary BO vs. Existing Plan</td>
<td>-1.198</td>
<td>0.0304</td>
<td>-1.499</td>
</tr>
</tbody>
</table>

between the existing plan and the signal plans proposed by the BO methods are tested for statistical significance using a one-sided two-sample $t$-test, with results given in Table 4.12. In each row of the Tables 4.11 and 4.12, the null hypothesis assumes that the first method (in Column 1) obtained a mean performance that is worse than or equal to the mean performance of the second method (in Column 1). The alternative hypothesis states that the first method obtained a mean performance that is better than the second method. For example, in the last row of Table 4.11, we test the alternative hypothesis that Stationary BO has a mean performance that is better than Dynamic BO. The $t$-tests are conducted for all 3 initial sets, and both the $t$-statistics and $p$-values are recorded in Tables 4.11 and 4.12. Each $t$-test is considered at the 10% level of significance. With 2 degrees of freedom in each test, the corresponding critical value for the $t$-statistic is -1.886. The $t$-tests with $t$-statistics smaller than the critical value have their null hypotheses rejected, and are displayed in bold in the tables.

Focusing first on Initial Set 1 (Figure 4-12a), we see that Stationary BO has the
best mean performance at the end of the optimization runs, followed by Dynamic BO and then Vanilla BO. The \( t \)-test results (Table 4.11) show that while Stationary BO is statistically better than Vanilla BO, it is not statistically better than Dynamic BO. The \( t \)-test results in Table 4.11 show that Dynamic BO is not statistically better than Vanilla BO. The results in Table 4.12 also show that both Dynamic BO and Stationary BO have mean performance that are statistically better than the existing plan. Dynamic BO registered a 3.0% reduction in the expected travel time compared to the existing plan, while Stationary BO recorded a 4.3% improvement over the existing plan. Figure 4-12a also shows that Stationary BO was the fastest in finding solutions better than the existing plan (7 iterations). In comparison, Vanilla BO (resp. Dynamic BO) took 10 iterations (resp. 18 iterations).

For Initial Set 2 (Figure 4-12b), Stationary BO again outperforms Vanilla BO and Dynamic BO. While the difference in performance compared to Vanilla BO is statistically significant at the 10% level of significance (Table 4.11), it is not statistically significant compared to Dynamic BO. Also, although the mean performance of Dynamic BO is slightly better than Vanilla BO as seen in Figure 4-12b, this difference is not statistically significant. Stationary BO is also the only BO method to have mean performance that is statistically better than the existing plan for this initial set, achieving a 3.4% improvement compared to the existing plan. As with Initial Set 1, Stationary BO was the fastest in finding solutions better than the existing plan (4 iterations).

For Initial Set 3 (Figure 4-12c), the ranking of the BO methods remains the same – Stationary BO has the best mean performance again, followed by Dynamic BO and Vanilla BO respectively. In fact, both Dynamic BO and Stationary BO are statistically better than Vanilla BO as seen from Tables 4.11. However, there is no statistical difference between the mean performances of Dynamic BO and Stationary BO. Furthermore, Stationary BO is the only BO method that has mean performance statistically better than the existing plan, registering a 3.6% improvement compared to the existing plan. In this initial set, Dynamic BO is the fastest in finding solutions better than the existing plan (13 iterations), compared to Vanilla BO and Stationary
BO (17 and 20 iterations respectively).

We also consider the queue lengths in the network for every 10 minute interval. For each BO run, we record the time-average total queue lengths at every 10 minute interval, for the best signal plan. The total queue lengths is the sum of the actual queue lengths and the virtual queue lengths. The actual queue lengths are composed of vehicles that are physically in the network. In the simulation, vehicles that are trying to enter a link in the network, but are unable to due to the link being full, are placed in the virtual queues. Hence, tracking both actual and virtual queue lengths allows us to fully understand the levels of congestion in the network. Figure 4-13 plots the mean queue lengths for every 10 minute interval, as achieved by the best signal plans proposed by each BO method. Figure 4-13a plots the total queue lengths (i.e. sum of actual and virtual queues), while Figures 4-13b and 4-13c show the breakdown of the length of the actual queues and the virtual queues respectively. For each plot in Figure 4-13, the x-axis represents time, while the y-axis denotes queue length aggregated over all links. Each bar shows the mean queue length for a given 10-minute interval, and the error bars represent the 95% confidence interval. Within each group of bars, the grey (left) bar corresponds to Vanilla BO, the blue (center) bar to Dynamic BO, and the green (right) bar to Stationary BO.

As can be seen from Figure 4-13a, there is a continuous build-up of the total mean queue length throughout the hour of simulation. Figures 4-13b and 4-13c also show increases in the actual and virtual queue lengths over time respectively. This indicates significant congestion in the MTM network. From Section 4.4.2, we saw that Dynamic BO is competitive but has no significant advantage over the other methods at high levels of congestion, especially with fewer time intervals. Hence, this may be a reason why Dynamic BO does not perform better than the other BO methods.

However, we also see from Figure 4-13a that the total mean queue length for the signal plans identified by Vanilla BO and Dynamic BO are smaller than Stationary BO. This is especially true in the later parts of the hour (e.g. at the 50-minute and 60-minute marks), where the difference in queue lengths of Stationary BO and the other 2 methods are more pronounced. This implies that solving for dynamic signal
plans can lead to shorter queue lengths. The breakdown of the queue lengths show
that while actual queue lengths (Figure 4-13b) are slightly smaller for Stationary BO,
the virtual queue lengths (Figure 4-13c) are significantly larger. An implication of
this observation is that the signal plans of Stationary BO are preventing vehicles
from entering the network. As a result, since the travel time of vehicles in the virtual
queues are not tracked in the simulation, their time spent in the virtual queues are
not counted in the objective function. Hence, even though Stationary BO was able
to find signal plans with better objective function values, the signal plans may not be
that equitable. This also further supports the observations from the SAN case study
that Dynamic BO identifies more equitable signal plans than Stationary BO.

Figure 4-13: (a) Total mean queue length, (b) total actual queue length, and (c) total
virtual queue length at a given time interval.
4.5 Conclusion

In this chapter, we proposed a dynamic analytical traffic model for use as a surrogate model in the GP prior when performing BO. While the model does not prioritize accuracy, it is able to provide useful prior information with sufficient computational efficiency to be used for optimization purposes. At the same time, it has a computational complexity that is quadratic in the number of links being modeled, making it suitable for use in large-scale networks. We also showed how a dynamic SO problem can be formulated so that it is suitable for BO. This allows us to extend the BO techniques proposed in Chapter 3 to tackle dynamic SO problems. By incorporating the dynamic analytical traffic model in the GP prior, we provide a BO framework for tackling dynamic transportation SO problems.

Our results show that using the dynamic analytical traffic model as the surrogate model in the GP covariance function can lead to similar or better solutions than not using any problem-specific information. It is also competitive with using the stationary traffic model as the surrogate model in the GP covariance function. This is especially true when optimizing for a greater number of time intervals. Hence, this suggests that there is potential in using the dynamic analytical traffic model to tackle dynamic transportation SO problems with BO. In particular, using the dynamic analytical traffic model as the surrogate model may also be better at finding more equitable solutions, compared to using the stationary traffic model.

In this work, we have laid the groundwork for a new dynamic analytical traffic model. Further improvements to the model can be investigated. One such improvement involves finding ways to account for the uneven distribution of vehicle density along the links. Based on the current formulation, the link dynamics are modeled using the average link densities. However, this can lead to inaccuracies in the link outflow and blocking at the upstream end of the link if vehicle density is unevenly distributed along the link. Hence, this can lead to errors when modeling the link dynamics. At the same time, further improvements to the computational efficiency of the dynamic analytical traffic model will also enhance its added value as a surrogate
model in BO.

When evaluating the different BO methods under different scenarios, we used a toy network to try to better understand the differences between the BO methods. However, the single arterial network used in the case study in Section 4.4.2 may have been too simple. In that case study, even the method which did not make use of any problem-specific information is able to solve the optimization problem relatively efficiently. Hence, this makes it difficult to separate the methods in terms of their performance under the different scenarios tested. Future studies can be done using slightly larger, more complex networks to test the different BO methods under various scenarios.
Chapter 5

Conclusion

This thesis introduces two different methods for balancing exploration and exploitation when tackling high-dimensional simulation-based optimization problems through the use of problem-specific prior information. The aim of finding a good balance between exploration and exploitation is to achieve efficient optimization. This is particularly important when working with a limited computational budget. The proposed methods were applied to transportation problems, with the results demonstrating the effectiveness of the methods in achieving efficient optimization under limited computational budgets.

In Chapter 2, we provide an alternative approach to using problem-specific prior information for optimization purposes. This approach involves using the problem-specific prior information for sampling points that are fed into the optimization algorithm. This is done by constructing a sampling distribution based on problem-specific information in the form of an analytical model. The resulting sampling distribution assigns greater probability of sampling points with good performance as predicted by the analytical model. As a result, the proposed inverse cumulative distribution function (cdf) sampling mechanism is able to efficiently sample for points with good performance. Unlike sampling points using uniform sampling, which is purely exploration, the inverse cdf sampling mechanism introduces exploitation components through the use of the problem-specific prior information in the analytical model. Furthermore, Chapter 2 also provides insight into whether the problem-specific infor-
formation should be used in the sampling mechanism and/or optimization algorithm. The results show that using the inverse cdf sampling mechanism can lead to similar or better performance compared to the traditional method of using the problem-specific information only in the optimization algorithm. This is in addition to reducing the overall computational runtime required.

Chapter 3 proposes approaches to enable Bayesian optimization (BO) for tackling high-dimensional optimization problems. More specifically, we identify approaches for using problem-specific prior information for both exploration and exploitation when performing BO. The problem-specific information is provided in the form of an analytical surrogate model. We show that exploitation can be achieved when the analytical surrogate model is incorporated in the prior mean function of the Gaussian process (GP) prior. On the other hand, more efficient exploration can be induced when the analytical surrogate model is incorporated in the covariance function of the GP prior. When used this way, the problem-specific information is able to help alleviate some of the problems that plague traditional BO when it comes to high-dimensional problems. In Chapter 3, we also evaluate the effect of different biases in the analytical surrogate model on BO performance. The results reveal that the proposed approach of incorporating the problem-specific information in the GP covariance function is relatively robust to biases in the surrogate model. Hence, this opens up the possibility of using models which are less accurate, as long as they are (anti-)correlated to the objective function. While we may have applied the proposed approaches to a transportation problem, the proposed approaches can easily be generalized to other types of optimization problems with expensive-to-evaluate objective functions. This is true as long as there is a suitable surrogate model for approximating the objective function.

In Chapter 4, we extend the BO framework of Chapter 3 to tackle high-dimensional dynamic transportation problems. This allows us to take advantage of the ability of the proposed approaches in Chapter 3 to balance exploration and exploitation. However, in order to effectively perform BO on a dynamic problem, the GP covariance function should be ideally be non-separable, so as to properly model space-time in-
teractions. We achieve this by developing a scalable dynamic analytical traffic model that is also sufficiently computationally efficient for use in BO. This dynamic analytical traffic model functions as the analytical surrogate model that is incorporated into the GP covariance function when performing BO. We also further tested several scenarios to see when it is advantageous to use the dynamic analytical traffic model in BO. The results show that the dynamic analytical traffic model can be used in BO to tackle large-scale SO problems. It also suggests that the use of the dynamic analytical traffic model can be advantageous under certain scenarios.

The methods proposed in this thesis have shown the importance of balancing exploration and exploitation when tackling high-dimensional simulation-based optimization problems. Moving forward, further work can be done to better understand how better optimization performance can be achieved.

The BO approaches proposed in Chapter 3 were shown to be sensitive to the initial points used to fit the GP models at the start of the optimization runs. These initial points were sampled uniformly at random. However, there could be better ways to sample for the initial points. For instance, it might be worth considering how initial points sampled using the inverse cdf sampling mechanism of Chapter 2 will affect the BO performance.

Additionally, the results of Chapter 3 imply that using the analytical surrogate model in the prior mean function of the GP allows good solution to be found quickly, but may potentially limit longer term performance. Hence, this suggests that it might be possible to fine-tune the balance between exploitation and exploration when performing BO. This can be done by switching between a constant prior mean function and the surrogate model-based prior mean function at different iterations. A systematic method switching between prior mean functions could potentially lead to better optimization performance within the limited computational budget.

Lastly, the methods proposed in this thesis can naturally be extended to other problem types and even fields beyond transportation. In this thesis, the methods were all demonstrated using a traffic signal control optimization problem. However, the inverse cdf sampling mechanism of Chapter 2 and the dynamic analytical traffic model
of Chapter 4 can also be used to address other types of transportation optimization problems. Furthermore, the BO framework for tackling high-dimensional problems (Chapter 3) can easily be applied to any problem, as long as there exists a surrogate model which can approximate the objective function.

As the use of expensive-to-evaluate simulation models become increasingly prevalent in the field of transportation, we believe that optimization methods with good short-term performance will become more relevant. This thesis helps to lay the groundwork for such techniques, so that future research may continue to build on it and bring about further improvements.
Appendix A

Appendices of Chapter 2

A.1 List of Notation

- \( f \) objective function (expected number of vehicles in the road network);
- \( f^A \) approximate analytical expression of the objective function;
- \( F \) random variable denoting the number of vehicles in the road network;
- \( x_j \) green split of signal phase \( j \) (decision variable);
- \( z \) vector of endogenous simulation variables (e.g., queue-lengths, vehicular speeds per lane);

**Exogenous problem parameters:**

- \( c_\ell \) cycle time of intersection \( \ell \);
- \( d_\ell \) fixed cycle time of intersection \( \ell \);
- \( e_i \) ratio of fixed green time to cycle time of signalized link \( i \);
- \( x^{L,B} \) vector of minimal green splits;
- \( \theta \) vector of exogenous parameters;
- \( n \) total number of signal phases to be optimized (i.e., dimension of the decision vector \( x \));
- \( I \) set of intersection indices;
- \( P_1(\ell) \) set of phase indices of intersection \( \ell \);

**Exogenous parameters of the analytical model:**

- \( \gamma_i \) external arrival rate at queue \( i \);
- \( m \) total number of queues in the network;
- \( k_i \) space capacity of queue \( i \) in terms of number of vehicles;
- \( p_{ij} \) transition probability from queue \( i \) to queue \( j \);
- \( s \) saturation flow rate [veh/h];
- \( M \) set of all queues;
- \( L \) set of indices of the signalized lanes;
\[ U_i \] set of upstream queues of queue \( i \);
\[ D_i \] set of downstream queues of queue \( i \);
\( \mathcal{P}_2(i) \) set of phase indices of lane \( i \);
\( \mathcal{P}_3(i) \) index of intersection that queue \( i \) leads to;
\( \mathcal{P}_4(i) \) set of phase indices at the intersection that queue \( i \) leads to, but not including the indices of phases which are green in favor of vehicles in queue \( i \);

Endogenous parameters of the analytical model:
\( \rho_i \) traffic intensity;
\( \lambda_i \) arrival rate;
\( N_i \) total number of vehicles in queue \( i \);
\( P(N_i = k_i) \) probability of queue \( i \) being full, also known as the blocking or spill-back probability;
\( \mu_i \) service rate of link \( i \);

Analytical sampling distribution notation:
\( \hat{\rho}_i \) upper bound on the possible values of \( \rho_i \);
\( g_{\rho_1,\ldots,\rho_j} \) joint pdf of \( \rho_i \) to \( \rho_j \);
\( G_{\rho_1,\ldots,\rho_j} \) joint cdf of \( \rho_i \) to \( \rho_j \);
\( G_{\rho_i | \rho_j} \) conditional cdf of \( \rho_i \) conditional on \( \rho_j \);
\( \kappa_0 \) normalization constant to ensure \( g \) integrates to 1;
\( \kappa_1 \) upper bound of \( f^A \);
\( u_i \) realization of a univariate standard uniform random variable.

A.2 Derivation of Sampling Distributions and Transformation of \( \rho \) to \( x \) Based on the Analytical Network Model (2.9)

This appendix details the derivations of the analytical expressions for the marginal and conditional cdf’s in Eq. (2.16)-(2.24), as well as the transformation of \( \rho \) to \( x \) (Section 2.2.4).

A.2.1 Derivation of the Joint cdf \( G_{\rho_1,\ldots,\rho_m}(\rho_1, \ldots, \rho_m) \)

In order to obtain the analytical expressions of the marginal and conditional cdf’s (Eq. (2.16)-(2.24)), we first evaluate the integral in Eq. (2.12) to get the analytical
expression of the joint cdf. From Eq. (2.12), we have:

\[
G_{\rho_1,\ldots,\rho_m}(\rho_1,\ldots,\rho_m) = \frac{1}{\kappa_0} \int_0^\rho \kappa_1 - F^A(\tilde{\rho}_1,\ldots,\tilde{\rho}_m) \, d\tilde{\rho} \tag{A.2.1}
\]

\[
= \frac{1}{\kappa_0} \left[ \int_0^\rho \kappa_1 d\tilde{\rho} - \int_0^\rho f^A(\tilde{\rho}_1,\ldots,\tilde{\rho}_m) d\tilde{\rho} \right] \tag{A.2.2}
\]

Computing the first integral in Eq. (A.2.2) gives:

\[
\int_0^\rho \kappa_1 d\tilde{\rho} = \kappa_1 \rho_1 \ldots \rho_m \tag{A.2.3}
\]

\[
= \kappa_1 \prod_{i=1}^m \rho_i \tag{A.2.4}
\]

Using the expressions for \( f^A \) in Eq. (2.7), the second integral in Eq. (A.2.2) can be computed as:

\[
\int_0^\rho f^A(\tilde{\rho}_1,\ldots,\tilde{\rho}_m) d\tilde{\rho} = \int_0^\rho \sum_{i=1}^m \mathbb{E}[N_i(\tilde{\rho}_i)] d\tilde{\rho} \tag{A.2.5}
\]

\[
= \sum_{i=1}^m \int_0^\rho \mathbb{E}[N_i(\tilde{\rho}_i)] d\tilde{\rho} \tag{A.2.6}
\]

\[
= \sum_{i=1}^m \left[ \int_0^{\rho_i} \mathbb{E}[N_i(\tilde{\rho}_i)] d\tilde{\rho}_i \right] \left( \prod_{j \neq i} \rho_j \right) \tag{A.2.7}
\]

The transition from Eq. (A.2.5) to Eq. (A.2.6) makes use of the fact that \( N_i(\rho_i) \) is a function of \( \rho_i \) only. Similarly, since \( N_i(\rho_i) \) does not depend on \( \rho_j, j \neq i \), integrating with respect to \( \rho_j, j \neq i \), produces Eq. (A.2.7). This simplifies the integral to a univariate integral. We define \( h(\rho_i) \) to be the integral in Eq. (A.2.7) and evaluate it:

\[
h(\rho_i) = \int_0^{\rho_i} \mathbb{E}[N_i(\tilde{\rho}_i)] d\tilde{\rho}_i \tag{A.2.8}
\]

\[
= \int_0^{\rho_i} \tilde{\rho}_i \frac{(k_i + 1)\tilde{\rho}_i^{k_i+1}}{1 - \tilde{\rho}_i^{k_i+1}} d\tilde{\rho}_i \tag{A.2.9}
\]

\[
= -\rho_i - \log(1 - \rho_i) - \int_0^{\rho_i} \frac{(k_i + 1)\tilde{\rho}_i^{k_i+1}}{1 - \tilde{\rho}_i^{k_i+1}} d\tilde{\rho}_i \tag{A.2.10}
\]

\[
= -\rho_i - \log(1 - \rho_i) + \rho_i \log \left( 1 - \tilde{\rho}_i^{k_i+1} \right) - \int_0^{\rho_i} \log \left( 1 - \tilde{\rho}_i^{k_i+1} \right) d\tilde{\rho}_i \tag{A.2.11}
\]
where Eq. (A.2.9) uses the expression for \( f^A \) in Eq. (2.8). Integrating the first term in Eq. (A.2.9) leads to Eq. (A.2.10). Finally, Eq. (A.2.11) is obtained by evaluating the integral in Eq. (A.2.10) by parts. Note that the term in the integral in Eq. (A.2.11) can be Taylor expanded before being integrated, giving:

\[
\int_0^{\rho_i} \log \left(1 - \rho_i^k+1\right) d\tilde{\rho}_i = \int_0^{\rho_i} \left[ - \sum_{\alpha=1}^{\infty} \frac{\tilde{\rho}_i^{\alpha k_i+1}}{\alpha} \right] d\tilde{\rho}_i \quad (A.2.12)
\]

\[
= - \sum_{\alpha=1}^{\infty} \frac{1}{\alpha} \int_0^{\rho_i} \tilde{\rho}_i^{\alpha k_i+1} d\tilde{\rho}_i \quad (A.2.13)
\]

\[
= - \sum_{\alpha=1}^{\infty} \frac{\rho_i^{\alpha k_i+1}}{\alpha(\alpha k_i + \alpha + 1)} \quad (A.2.14)
\]

where Eq. (A.2.12) uses the Taylor expansion of the term in the integral. The independence of \( \alpha \) and \( \rho_i \) allows the order of the summation and integral to be interchanged in Eq. (A.2.13). Finally, the evaluation of the integral leads to the expression in Eq. (A.2.14). Inserting Eq. (A.2.14) into Eq. (A.2.11) gives the final expression of \( h(\rho_i) \) as shown in Eq. (2.23):

\[
h(\rho_i) = -\rho_i - \log(1 - \rho_i) + \rho_i \log(1 - \rho_i^{k_i+1}) + \sum_{\alpha=1}^{\infty} \frac{\rho_i^{\alpha k_i+1}}{\alpha(\alpha k_i + \alpha + 1)} \quad \forall i \in \mathcal{M}
\]

\[(A.2.15)\]

Putting together Eq. (A.2.2), (A.2.4), (A.2.7) and (A.2.15), the analytical expression for the joint cdf is:

\[
G_{\rho_1, \ldots, \rho_m}(\rho_1, \ldots, \rho_m) = \frac{1}{\kappa_0} \left[ \kappa_1 \prod_{i=1}^{m} \rho_i - \sum_{i=1}^{m} h(\rho_i) \left( \prod_{j \neq i}^{m} \rho_j \right) \right]
\]

\[(A.2.16)\]

**A.2.2 Derivation of the Marginal cdf \( G_{\rho_1}(\rho_1) \) (Eq. (2.17))**

The marginal cdf \( G_{\rho_1}(\rho_1) \) is used in Step 1 of Algorithm 1 to obtain the sample value for \( \rho_1 \). It can be derived from the joint cdf (Eq. (A.2.16)) by marginalizing the variables \( \rho_i \) for \( i = 2, \ldots, m \) out of the joint cdf. Since the support of the joint cdf is taken to be \([0, \hat{\rho}]\) (Section 2.2.4), this can be done by setting \( \rho_i = \hat{\rho}_i \)
for \( i = 2, \ldots, m \) (see Evans and Rosenthal (2004, Theorem 2.7.3)). The resulting
equation for \( G_{\rho_i}(\rho_1) \) is then derived to give Eq. (2.17), as follows:

\[
G_{\rho_i}(\rho_1) = G_{\rho_1, \rho_2, \ldots, \rho_m}(\rho_1, \hat{\rho}_2, \ldots, \hat{\rho}_m) \\
= \frac{1}{\kappa_0} \left[ \kappa_1 \rho_1 \prod_{i=2}^{m} \hat{\rho}_i - h(\rho_1) \prod_{i=2}^{m} \hat{\rho}_i - \sum_{i=2}^{m} h(\hat{\rho}_i) \left( \rho_1 \prod_{i=2 \atop j \neq i}^{m} \hat{\rho}_j \right) \right] \\
= \frac{1}{\kappa_0} \left[ (\kappa_1 \rho_1 - h(\rho_1)) \prod_{i=2}^{m} \hat{\rho}_i - \rho_1 \sum_{i=2}^{m} h(\hat{\rho}_i) \left( \prod_{i=2 \atop j \neq i}^{m} \hat{\rho}_j \right) \right] \\
= \frac{1}{\kappa_0} \left[ (\kappa_1 \rho_1 - h(\rho_1)) \prod_{i=2}^{m} \hat{\rho}_i - \rho_1 r_1 \right] \\
\tag{A.2.18}
\tag{A.2.19}
\tag{A.2.20}
\]

where Eq. (A.2.18) is derived from Eq. (A.2.16) by setting \( \rho_i = \hat{\rho}_i \) for \( i = 2, \ldots, m \). Eq. (A.2.19) and (A.2.20) are then obtained by factorizing and tidying up Eq. (A.2.18). The constant \( r_1 \) is defined in Eq. (2.21) and (A.2.32), and helps to simplify the expression (A.2.20).

A.2.3 Derivation of Conditional cdf \( G_{\rho_i|\rho_1, \ldots, \rho_{i-1}}(\rho_i|\rho_1, \ldots, \rho_{i-1}) \) (Eq. (2.19))

The conditional cdf \( G_{\rho_i|\rho_1, \ldots, \rho_{i-1}}(\rho_i|\rho_1, \ldots, \rho_{i-1}) \) is used in Step 2c in Algorithm 1 to get the sample value for \( \rho_i \), conditioned on the components that have been sampled. The expression for the conditional cdf given by Eq. (2.19) can be derived by first
noting that

\[ G_{\rho_1, \ldots, \rho_{i-1}}(\rho_i | \rho_1, \ldots, \rho_{i-1}) \]

\[ = g_{\rho_i | \rho_1, \ldots, \rho_{i-1}}(P_i \leq \rho_i | \rho_1, \ldots, \rho_{i-1}) \tag{A.2.21} \]

\[ = \frac{g_{\rho_i, \rho_1, \ldots, \rho_{i-1}}(P_i \leq \rho_i, \rho_1, \ldots, \rho_{i-1})}{g_{\rho_1, \ldots, \rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})} \tag{A.2.22} \]

\[ = \frac{1}{g_{\rho_1, \ldots, \rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})} \int_0^{\rho_i} g_{\rho_i, \rho_1, \ldots, \rho_{i-1}}(\tilde{\rho}_i, \rho_1, \ldots, \rho_{i-1}) \, d\tilde{\rho}_i \tag{A.2.23} \]

\[ = \frac{1}{g_{\rho_1, \ldots, \rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})} \left[ \frac{d}{d\rho_1} \int_0^{\rho_i} \ldots \frac{d}{d\rho_1} \int_0^{\rho_1} g_{\rho_1, \rho_1, \ldots, \rho_{i-1}}(\tilde{\rho}_1, \tilde{\rho}_1, \ldots, \tilde{\rho}_{i-1}) \, d\tilde{\rho}_1 \ldots d\tilde{\rho}_{i-1} \right] \tag{A.2.24} \]

\[ = \frac{1}{g_{\rho_1, \ldots, \rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})} \left[ \frac{d^{i-1}}{d\rho_1 \ldots d\rho_{i-1}} \int_0^{\rho_i} \ldots \int_0^{\rho_1} g_{\rho_1, \rho_1, \ldots, \rho_{i-1}}(\tilde{\rho}_1, \tilde{\rho}_1, \ldots, \tilde{\rho}_{i-1}) \, d\tilde{\rho}_1 \ldots d\tilde{\rho}_{i-1} \right] \tag{A.2.25} \]

\[ = \frac{1}{g_{\rho_1, \ldots, \rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})} \left[ \frac{d^{i-1} G_{\rho_1, \rho_2, \ldots, \rho_{i-1}}(\rho_i, \rho_1, \ldots, \rho_{i-1})}{d\rho_1 \ldots d\rho_{i-1}} \right] \tag{A.2.26} \]

where \( P_i \) is the random variable representing the sampled values of \( \rho_i \). Eq. (A.2.21) makes use of the definition of cdf’s (Evans and Rosenthal 2004, Definition 2.5.1) to represent the conditional cdf in terms of the conditional pdf. The transition from Eq. (A.2.21) to (A.2.22) uses the definition of conditional pdf’s (Evans and Rosenthal 2004, Definition 2.8.3). The integral in Eq. (A.2.23) can be integrated with respect to \( \rho_1 \) and then differentiated with respect to \( \rho_1 \) again with no net effect, because of the Leibniz’s integral rule (Abramowitz and Stegun 1964, Eq. (3.3.7)):

\[ \frac{d}{dc} \int_{a(c)}^{b(c)} f(x, c) \, dx = \int_{a(c)}^{b(c)} \frac{\partial}{\partial c} f(x, c) \, dx + f(b, c) \frac{db}{dc} - f(a, c) \frac{da}{dc} \tag{A.2.27} \]

The first and third term on the right hand side of Eq. (A.2.27) would be zero, leaving just the second term. This is repeated with \( \rho_j \) for \( j = 2, \ldots, i-1 \) to give Eq. (A.2.24). The order of the integrals and derivatives can be switched since \( \rho_1, \ldots, \rho_{i-1} \) are as-
sumed independent for the purpose of sampling. Finally, Eq. (A.2.26) is obtained by making use of the fact that the integral of the joint pdf is the joint cdf of $\rho_1, \ldots, \rho_i$.

Based on the expression (A.2.26) for the conditional cdf, we require the analytical expressions for the joint pdf $g_{\rho_1, \ldots, \rho_i}(\rho_1, \ldots, \rho_i)$ and the joint cdf $G_{\rho_1, \ldots, \rho_i}(\rho_i, \rho_1, \ldots, \rho_i)$. We start with the joint cdf first; similar to deriving the marginal cdf in Section A.2.2, we marginalize the variables $\rho_j$ for $j = i+1, \ldots, m$ out of the full joint cdf $G_{\rho_1, \ldots, \rho_m}(\rho_i, \rho_1, \ldots, \rho_m)$:

$$G_{\rho_i, \rho_1, \ldots, \rho_{i-1}}(\rho_i, \rho_1, \ldots, \rho_{i-1}) = G_{\rho_1, \ldots, \rho_{m-1}}(\rho_1, \rho_2, \ldots, \rho_{m-1})$$

(A.2.28)

The derivative in Eq. (A.2.26) is then given by differentiating Eq. (A.2.29):

$$\frac{d^{i-1}G_{\rho_i, \rho_1, \ldots, \rho_{i-1}}(\rho_i, \rho_1, \ldots, \rho_{i-1})}{d\rho_1 \ldots d\rho_{i-1}} = \frac{1}{\kappa_0} \left[ \kappa_1 \prod_{j=1}^i \mu_j \prod_{k=i+1}^m \hat{\rho}_k - \sum_{j=1}^i h(\rho_j) \left( \prod_{k=1, k \neq j}^i \mu_k \right) \left( \prod_{k=i+1}^m \hat{\rho}_k \right) \right]$$

(A.2.29)

In order to simplify the expression in Eq. (A.2.30), we define the variable $q_i$ and constant $r_i$, as given in Eq. (2.20) and (2.21):

$$q_i = \left( \sum_{j=1}^{i-1} \frac{dh(\rho_j)}{d\rho_j} \right) \left( \prod_{k=i+1}^m \hat{\rho}_k \right) \text{ for } i = 2, \ldots, m,$$

(A.2.31)

$$r_i = \sum_{j=i+1}^m \left( h(\hat{\rho}_j) \prod_{k=i+1, k \neq j}^m \hat{\rho}_k \right) \forall i \in \mathcal{M}$$

(A.2.32)

The derivative $\frac{dh(\rho_i)}{d\rho_i}$ in Eq. (A.2.31) can be evaluated as such:

$$\frac{dh(\rho_i)}{d\rho_i} = \frac{d}{d\rho_i} \int_{\rho_i}^{\rho_i} \mathbb{E} [N_i(\hat{\rho}_i)] \ d\hat{\rho}_i = \mathbb{E} [N_i(\rho_i)]$$

(A.2.33)
Inserting Eq. (A.2.31) and (A.2.32) into Eq. (A.2.30),

\[
\frac{d^{i-1}G_{\rho_1,\rho_2,\ldots,\rho_{i-1}}(\rho_i, \rho_1, \ldots, \rho_{i-1})}{d\rho_1 \ldots d\rho_{i-1}} = \frac{1}{\kappa_0} \left[ \kappa_i \rho_i \prod_{k+1}^m \hat{\rho}_k - q_i \rho_i - r_i \rho_i - h(\rho_i) \prod_{k=i+1}^m \hat{\rho}_k \right]
\]

(A.2.34)

We now consider the joint pdf \(g_{\rho_1,\ldots,\rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})\). The joint pdf can be represented as the derivative of the joint cdf (Evans and Rosenthal 2004, Corollary 2.5.1). Then, by observing that Eq. (A.2.35) is the same as setting \(\rho_i = \hat{\rho}_i\) in Eq. (A.2.34), we get the expression for the joint pdf in Eq. (A.2.37):

\[
g_{\rho_1,\ldots,\rho_{i-1}}(\rho_1, \ldots, \rho_{i-1}) = \frac{d^{i-1}G_{\rho_1,\rho_2,\ldots,\rho_{i-1}}(\rho_1, \ldots, \rho_{i-1})}{d\rho_1 \ldots d\rho_{i-1}}
\]

(A.2.35)

\[
= \frac{d^{i-1}G_{\rho_i,\rho_1,\ldots,\rho_{i-1}}(\hat{\rho}_i, \rho_1, \ldots, \rho_{i-1})}{d\rho_1 \ldots d\rho_{i-1}}
\]

(A.2.36)

\[
= \frac{1}{\kappa_0} \left[ \kappa_i \hat{\rho}_i \prod_{k+1}^m \hat{\rho}_k - q_i \hat{\rho}_i - r_i \hat{\rho}_i - h(\hat{\rho}_i) \prod_{k=i+1}^m \hat{\rho}_k \right]
\]

(A.2.37)

By inserting Eq. (A.2.34) and (A.2.37) into Eq. (A.2.26), we get the expression for the conditional cdf as given in Eq. (2.19):

\[
G_{\rho_i|\rho_1,\ldots,\rho_{i-1}}(\rho_i|\rho_1, \ldots, \rho_{i-1}) = \frac{\kappa_i \rho_i \prod_{k=i+1}^m \hat{\rho}_k - q_i \rho_i - r_i \rho_i - h(\rho_i) \prod_{k=i+1}^m \hat{\rho}_k}{\kappa_i \hat{\rho}_i \prod_{k=i+1}^m \hat{\rho}_k - q_i \hat{\rho}_i - r_i \hat{\rho}_i - h(\hat{\rho}_i) \prod_{k=i+1}^m \hat{\rho}_k} \quad \text{for } i = 2, \ldots, m
\]

(A.2.38)

### A.2.4 Transformation from \(\rho\) to \(x\)

Given the sampled values \(\rho\), we first transform them to \(\mu\), and subsequently to \(x\). The transformation of \(\rho\) to \(\mu\) involves using the analytical model (2.9). Rearranging Eq. (2.9b), we get an expression for \(\mu_i\) as a function of the sampled \(\rho_i\):

\[
\mu_i = \lambda_i \left[ \rho_i - \left( \sum_{j \in D_i} p_{ij} P(N_j = k_j) \right) \left( \sum_{j \in D_i} p_j \right)^{-1} \right]
\]

(A.2.39)

From the values of \(\mu_i\), a set of values for the decision variables \(x_j\) can be obtained.
From Eq. (2.10), we have

$$\mu_i = s \left( e_i + \sum_{j \in P_2(i)} x_j \right) \quad \forall i \in \mathcal{L}$$  \hspace{1cm} (A.2.40)

$$\implies \sum_{j \in P_2(i)} x_j = \frac{\mu_i}{s} - e_i$$  \hspace{1cm} (A.2.41)

We can replace $P_2(i)$ with the matrix elements

$$\xi_{ij} = \begin{cases} 1 & \text{if } j \in P_2(i) \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (A.2.42)

Rewriting Eq. (A.2.41) with $\xi_{ij}$,

$$\sum_j \xi_{ij} x_j = \frac{\mu_i}{s} - e_i \quad \forall i \in \mathcal{L}$$  \hspace{1cm} (A.2.43)

In matrix form, this can be written as

$$\Xi \mathbf{x} = \frac{1}{s} \mathbf{\mu} - \frac{1}{c} \mathbf{e}$$  \hspace{1cm} (A.2.44)

$$\mathbf{x} = \Xi^{-1} \left[ \frac{1}{s} \mathbf{\mu} - \frac{1}{c} \mathbf{e} \right]$$  \hspace{1cm} (A.2.45)

where $\Xi \in \{0,1\}^{\mathcal{L} \times |\mathcal{L}|}$ is made up of the elements $\xi_{ij}$, and is known from the road network. The system of linear equations (Eq. (A.2.45)) can be solved using a pseudoinverse of $\Xi$. A pseudoinverse is used because the linear system of equations is overdetermined, since the number of queues leading to a given signalized intersection is greater than the number of signal phases at that intersection. A popular pseudoinverse that can be used is the Moore-Penrose pseudoinverse (see e.g. Campbell and Meyer (1991, Chapter 1)), which provides a unique inverse matrix. To compute the pseudoinverse, the code of Luong (2009) was used, as it is able to handle sparse matrices. The resulting green splits $\mathbf{x}$ is unique, since the code of Luong (2009) uses the Moore-Penrose pseudoinverse, which is unique. In cases where the sum of the green
splits for a given intersection does not add up to the cycle time of that intersection, the green splits are scaled proportionally so that they add up correctly.

### A.3 Details on Validation Plot Generation

In each plot of Figure 2-4, the phases that are not plotted have green split values that are fixed at the histogram bin center value where the inverse cdf sampling assigns the highest marginal probability as shown in Figure 2-5a. As an example, for the top left plot, \(x_5\) and \(x_7\) were both fixed at 0.777778. For the plot based on the queueing model, the plots were generated by computing the expected number of vehicles in the network using the expression in Eq. (2.8), with the green splits for Phases 1, 3, 5 and 7 used to compute the corresponding \(\rho\) values using Eq. (2.9) and (2.10). For the plots based on traffic simulation results, 10 traffic simulation replications were used to compute the average objective function estimates at each point.

To generate the marginal distributions in Figure 2-5b, we first note that the toy network has 4 degrees of freedom due to the linear cycle time constraint on the green split at each intersection (Eq. (2.4)). The joint green split distribution based on traffic simulation estimates can be represented by

\[
g_{x_1,x_3,x_5,x_7}(x_1, x_3, x_5, x_7) = \frac{1}{\kappa_0} \left( \kappa'_1 - \mathbb{E}[F(x_1, x_3, x_5, x_7)] \right)
\]  

(A.3.1)

where \(g_{x_1,x_3,x_5,x_7}(x_1, x_3, x_5, x_7)\) is the joint green split distribution, \(\kappa_0\) is the normalization constant as computed according to Eq. (2.30), and \(\kappa'_1\) is a constant satisfying the constraint \(\kappa'_1 \geq \mathbb{E}[F(x_1, x_3, x_5, x_7)]\) to ensure non-negativity. Here, the phases are numbered such that Phases 1 and 2 correspond to the first intersection on the left, as displayed in Figure 2-2, while Phases 3 and 4 correspond to the second intersection from the left and so on. The odd numbered phases are the phases that are green in favor of the main arterial (i.e., OD pairs (1) \(\rightarrow\) (2) and (3) \(\rightarrow\) (4) of Figure 2-3), while the even numbered phases are green in favor of the side roads (i.e., OD pairs (5) \(\rightarrow\) (6), (7) \(\rightarrow\) (8), (9) \(\rightarrow\) (10), (11) \(\rightarrow\) (12) and (13) \(\rightarrow\) (14) of Figure 2-3). Then, treating
g_{x_1,x_3,x_5,x_7}(x_1, x_3, x_5, x_7) like a joint pdf, we compute the marginal distributions by summing out all but the relevant variable. For instance, to obtain the marginal distribution for $x_1$, we sum out $x_3$, $x_5$ and $x_7$ (see Eq. (A.3.2)). We first identify 6 green splits that are uniformly distributed in the feasible range of green split values for each phase, sort them in increasing order, and represent them by the indices $i, j, k, l$ for Phase 1, 3, 5 and 7 respectively (i.e., $(i, j, k, l) \in [1, 2, 3, 4, 5, 6]^4$). The choice of 6 green splits was used as a compromise between having as many green splits as possible and computational runtime. This leads to $6^4 = 1296$ possible combinations of green splits for the 4 intersections. Each of these combinations are simulated 10 times to obtain an average of the objective function estimate. The marginal distributions are then approximated using a discrete sum. Eq. (A.3.2) shows the case for calculating the marginal distribution for Phase 1, with the rest of the phases (i.e., Phases 3, 5 and 7) taking on a similar form:

$$g_{x_1}(x_{1,i}) \approx \frac{1}{k_0} \sum_{j=1}^{6} \sum_{k=1}^{6} \sum_{l=1}^{6} \left[ k'_j - f(x_{ijkl}) \right] \Delta x_3 \Delta x_5 \Delta x_7$$  \hspace{1cm} (A.3.2)

where $x_{1,i}$ denotes the $i^{th}$ green split value for Phase 1. $x_{ijkl}$ refers to the green split combination with $i, j, k, l$ representing the index of the green split of Phase 1, 3, 5 and 7 respectively. Each of $\Delta x_p$ is the interval between each successive green split for phase $p$.

### A.4 Case Study Implementation Details

The simulator’s objective function (Eq. (2.3)), which is the expected number of vehicles in the network, accounts for all vehicles that have started their trip. During simulation, when a given queue is full, additional vehicles entering that particular queue are placed in a virtual queue. In order to counter this in our experiments, the total demand of the network (i.e., inflow) is scaled to 75% of the original demand (28900 trips per hour), so that the number of vehicles in the virtual queues is small at any point in time during simulation. However, any vehicles that enter the virtual
queues due to congestion are still included in the count of the number of vehicles in the network (i.e. the objective function).

### A.5 Simulation-based Optimization Algorithm

The simulation optimization (SO) framework used for this study is based on the metamodel-based optimization method proposed by Osorio and Bierlaire (2013). A metamodel is an analytical function which attempts to approximate the underlying objective function. The notation to be used in the algorithm is defined at a given iteration $k$ as such:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k$</td>
<td>current iterate</td>
</tr>
<tr>
<td>$\Delta_k$</td>
<td>trust region radius</td>
</tr>
<tr>
<td>$m_k/\phi_k$</td>
<td>metamodel</td>
</tr>
<tr>
<td>$\nu_k = (\alpha_k, \beta_k)$</td>
<td>vector of metamodel parameters</td>
</tr>
<tr>
<td>$n_k$</td>
<td>total number of simulation runs conducted up to and including iteration $k$</td>
</tr>
<tr>
<td>$u_k$</td>
<td>total number of successive trial points rejected</td>
</tr>
<tr>
<td>$\hat{f}$</td>
<td>simulation estimate of objective function</td>
</tr>
<tr>
<td>$\tau_k$</td>
<td>number of optimization iterations since the last sampling iteration</td>
</tr>
<tr>
<td>$\bar{r}$</td>
<td>number of optimization iterations to run before switching to a sampling iteration, as defined by the simulation budget for sampling (see Table 2.6)</td>
</tr>
<tr>
<td>$n_{max}$</td>
<td>total number of simulation run permitted</td>
</tr>
<tr>
<td>$\bar{r}$</td>
<td>number of simulation replications per iteration</td>
</tr>
<tr>
<td>$d$</td>
<td>number of decision variables</td>
</tr>
<tr>
<td>$z$</td>
<td>endogenous variables</td>
</tr>
</tbody>
</table>

The metamodels used in Section 2.4 are denoted $m$ and $\phi$ (see Table 2.5). The $m$ metamodel makes use of the analytical network model estimate of the objective function (Eq. (2.8)), in addition to a functional component, as shown in Eq. (A.5.1). The $\phi$ metamodel is a general-purpose metamodel, consisting of just a quadratic polynomial, as shown in Eq. (A.5.2).

$$m_k(x_k, z; \alpha_k, \beta_k; p) = \alpha_k f^A(x_k, z; p) + \phi_k(x_k; \beta_k) \quad (A.5.1)$$
Before running the SO algorithm (Algorithm 4), the SO method (Table 2.5), along with the proportion of simulation budget to be used for sampling (Table 2.6), have to first be selected. The initial point \( x_0 \) is generated using uniform sampling, and evaluated by simulation to obtain the simulation estimate \( \hat{f}(x_0) \). The simulation estimate is then used to fit the metamodel parameters by solving a least squares problem (Osorio and Bierlaire 2013, Eq. 5). The initialization step (Step 0) is the same as Step 0 in the algorithm proposed by Osorio and Bierlaire (2013, Section 4.2). The notation used is the same, except for \( \bar{\gamma} \) and \( \bar{r} \), which are represented by \( \gamma \) and \( r \) in Osorio and Bierlaire (2013).

Subsequently, at each iteration, the algorithm either (i) samples a new point to evaluate (Step 1), or (ii) it optimizes the fitted metamodel (Steps 2-4) based on the proportion of simulation budget allocated to sampling. In the sampling step, a new point is generated by the chosen sampling strategy, and evaluated by simulation. The simulation observation is then used to update the metamodel parameters. The optimization steps (Steps 2-4) are the same as Steps 2, 3 and 5 in Osorio and Bierlaire (2013, Section 4.2). The notation used is the same, except for \( \psi_k \), which is represented by \( \rho_k \) in Osorio and Bierlaire (2013). The main difference between Algorithm 4 and the algorithm proposed in Osorio and Bierlaire (2013) is that the model improvement step (i.e., Step 3 in Osorio and Bierlaire (2013)) is replaced by the sampling step (Step 1 in Algorithm 4). The sampling step of Algorithm 4 is triggered after a deterministic number of optimization iterations, whereas the model improvement step of Osorio and Bierlaire (2013) is triggered based on the change in metamodel parameters. Also, the model improvement step of Osorio and Bierlaire (2013) uses uniform sampling to sample for a new point. For additional details about the metamodel-based optimization steps (Steps 2-4 in Algorithm 4), we refer the reader to Osorio and Bierlaire (2013, Section 4.2, Steps 2, 3 and 5).
Algorithm 4: Simulation-based optimization algorithm

Let $\eta_1, \bar{\gamma}, \gamma_{\text{inc}}, \bar{d}, \bar{u}, \Delta_{\text{max}}$ be constants which satisfy the following constraints:

$0 < \eta_1 < 1$, $0 < \bar{\gamma} < 1 < \gamma_{\text{inc}}$, $0 < \bar{d} < \Delta_{\text{max}}$, $\bar{u} \in \mathbb{N}^*$ (see Osorio and Bierlaire (2013, Section 4.3) for values used)

0. Initialization

(a) Select metamodel for use: $m$ (Eq. (A.5.1)) or $\phi$ (Eq. (A.5.2))

(b) Select sampling strategy: uniform sampling (i.e., Unif) or inverse cdf sampling (i.e., Invcdf; see Algorithm 1)

(c) Select proportion of simulation budget for sampling and set $\bar{\tau}$: 10% ($\bar{\tau} = 9$), 50% ($\bar{\tau} = 1$) or 100% ($\bar{\tau} = 0$)

(d) Set $k = 0$, $n_0 = 0$, $u_0 = 0$, $\tau_0 = 0$

(e) Sample an initial point $x_0$ using uniform sampling

(f) Determine $\Delta_0$ ($\Delta_0 \in (0, \Delta_{\text{max}})$) (see Osorio and Bierlaire (2013, Section 4.3) for values used)

(g) Run $\bar{r}$ simulations to get $\hat{f}(x_0)$ and set $n_0 = n_0 + \bar{r}$

(h) Compute $\nu_0$ according to Osorio and Bierlaire (2013, Eq. (5)) and fit an initial metamodel

(i) Set $n_{k+1} = n_k$, $k = k + 1$

1. Sampling
   if $\tau_k = \bar{\tau}$ do

   (a) Sample for a new point $x_k$ using the chosen sampling strategy

   (b) Run $\bar{r}$ simulations to get $\hat{f}(x_k)$ and set $n_k = n_k + \bar{r}$

   (c) Include the new observation in the set of point evaluated by simulation.

   Compute $\nu_k$ according to Osorio and Bierlaire (2013, Eq. (5)) and fit the new metamodel $m_{k+1}$ or $\phi_{k+1}$.

   (d) Set $n_{k+1} = n_k$, $\tau_{k+1} = 0$, $k = k + 1$

   (e) if $n_k < n_{\text{max}}$ do repeat Step 1, else stop

   else

   Go to Step 2

end

2. Optimization: step calculation

Compute a step $s_k$ that minimizes the metamodel $m_k$ or $\phi_k$, such that the trial point $x_k + s_k$ is within the trust region.
Algorithm 5: Simulation-based optimization algorithm (continued)

4. Optimization: acceptance or rejection of the trial point

(a) Run $\bar{r}$ simulations to get $\hat{f}(x_k + s_k)$ and set $n_k = n_k + \bar{r}$
(b) Compute the ratio $\psi_k = \frac{\hat{f}(x_k) - \hat{f}(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$ or $\psi_k = \frac{\hat{f}(x_k) - \hat{f}(x_k + s_k)}{\phi_k(x_k) - \phi_k(x_k + s_k)}$ depending on the chosen metamodel
(c) if $\psi_k \geq \eta_1$ do
   Accept the trial point: $x_{k+1} = x_k + s_k$, $u_k = 0$, $\tau_k = \tau_k + 1$
   else
   Reject the trial point: $x_{k+1} = x_k$, $u_k = u_k + 1$, $\tau_k = \tau_k + 1$
   end
(d) Include the new observation in the set of points evaluated by simulation. Compute $\nu_k$ according to Osorio and Bierlaire (2013, Eq. (5)) and fit the new metamodel $m_{k+1}$ or $\phi_{k+1}$

5. Optimization: trust region radius update

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

(a) if $\psi_k \leq \eta_1$ and $u_k \geq \bar{u}$, do set $u_k = 0$
(b) Set $n_{k+1} = n_k$, $u_{k+1} = u_k$, $k = k + 1$
(c) if $n_k < n_{max}$ do go to Step 1, else stop
A.6 Comparing Invcdf-$\phi$ and Unif-$\phi$

Figure A-1 plots the cdf’s for the 50% and 10% exploration settings, comparing the Invcdf-$\phi$ and Unif-$\phi$ methods. The left column of Figure A-1 shows the results for 50% exploration, while the right column displays the results for 10% exploration. Each row of plots considers a given initial point.

Figure A-1 shows that the cdf’s of Invcdf-$\phi$ are all positioned to the left of those of Unif-$\phi$ for both the 50% and 10% exploration settings. This indicates that Invcdf-$\phi$ outperforms Unif-$\phi$ for all initial points, which shows again that using problem-specific structural information in the exploration mechanism leads to better solutions. Given that Unif-$\phi$ performs much worse than Invcdf-$\phi$, we will not compare Unif-$\phi$ with the other methods in the subsequent figures.
Figure A-1: Cdf’s of the average number of vehicles in the network for the Invcdf-ϕ and Unif-ϕ methods using 50% and 10% exploration settings and considering 3 random initial points.
Appendix B

Appendices of Chapter 3

B.1 List of Notation

- $f$: objective function;
- $f^A$: approximate analytical model of the objective function;
- $F$: random variable denoting the stochastic output of a simulation run;
- $x$: vector of decision variables;
- $z$: vector of endogenous simulation variables;
- $p$: vector of deterministic exogenous parameters;
- $\chi$: feasible region;
- $D$: number of dimensions of the feasible region;
- $\tau$: i.i.d. Gaussian noise;

**GP notation:**

- $t$: number of observations;
- $t_0$: number of initial observations;
- $T$: budget for number of objective function evaluations;
- $m$: prior mean function of a GP;
- $m$: vector of prior mean function values;
- $\beta$: prior mean constant;
- $\alpha$: analytical model scaling constant;
- $k$: covariance function of a GP;
- $k$: vector of covariance function values;
- $K$: covariance matrix;
- $k_{SE}$: squared exponential covariance function;
- $k_{fA}$: analytical model-based covariance function;
- $\mu$: posterior (predictive) mean function of a GP;
- $\sigma^2$: posterior (predictive) variance of a GP;
- $\sigma_0^2$: covariance amplitude;
\( \ell \) covariance characteristic length-scale;
\( \ell_{\text{f.a}} \) analytical model length-scale;
\( \tau^2 \) variance of Gaussian noise;
\( I \) identity matrix;

Signal control problem exogenous parameters:
\( c_\ell \) cycle time of intersection \( \ell \);
\( d_\ell \) fixed cycle time of intersection \( \ell \);
\( \mathbf{x}^{L,B} \) vector of minimal green splits;
\( \mathcal{I} \) set of signal controlled intersection indices;
\( \mathcal{P}(\ell) \) set of signal phase indices of intersection \( \ell \).

## B.2 Implementation Details for Validation with 100-D Griewank Function

### B.2.1 Optimizing the EI Acquisition Function

The optimization of the EI acquisition function is traditionally carried out using DIRECT (Jones, Perttunen, and Stuckman 1993), which is a deterministic, derivative-free optimizer. Given that we are considering stochastic problems, this means that DIRECT is not a suitable optimizer. Since the analytical expression of the gradient of EI is available (Frean and Boyle 2008, Section 3), we use a multistart gradient ascent approach to maximize the acquisition function (Hutter, Hoos, and Leyton-Brown 2011, Shahriari et al. 2015). More specifically, we use the \textit{MultiStart} routine in Matlab, along with \textit{fmincon} as the solver.

### B.2.2 GP Implementation

In this work, the GP models were implemented using the GPML package for Matlab (Rasmussen and Nickisch 2018). The fitting of the GP posterior is done by updating the GP hyperparameters through maximum likelihood estimation (see Step 2a of Algorithm 3), using the \textit{minimize} function (which minimizes the negative log marginal likelihood) found in the GPML package for Matlab (Rasmussen and Nickisch 2018).
B.2.3 Validation with 100-D Griewank Function

To identify the fixed set of hyperparameter values for use in Section 3.3.4, we executed a grid search to find the best set of hyperparameter values. The hyperparameter values tested can be found in Table B.2. The first column of the table identifies the GP prior, while the second and third columns show the values tested for the mean function hyperparameter ($\beta$ for Standard and Proposed-Covariance, and $\alpha$ for Proposed-Mean and Proposed-Combined). The fourth and fifth columns show the values tested for the covariance amplitude and characteristic length-scale respectively. The last column represents the analytical model length-scale values, which is applicable only to the GP priors with $k_{fA}$ as the covariance function (i.e. Proposed-Covariance and Proposed-Combined). The values in bold represent the best set of fixed hyperparameter values for each GP prior. This same set of fixed hyperparameter values were used in Section 3.3.5 as well.

The tested values shown in Table B.2 were chosen to represent different orders of magnitude, with the goal of finding hyperparameter values of the right order of magnitude. As such, the best set of hyperparameter values found may not be the optimal set, as no further fine-tuning of the hyperparameter values was done. For each GP prior, the best set of hyperparameters was chosen by comparing the mean of the best objective function estimate of 3 optimization runs (as in Figure 3-3) for every set. The set that required the fewest number of iterations to reach within 0.05 of the global minimum value of 0 is taken as the best.

B.3 Case Study Implementation Details

The case study objective function (expected travel time) in Eq. (3.24) is evaluated by running simulations of the MTM model in Aimsun. In each simulation run, the travel time of each vehicle is taken as the total amount of time that vehicle spent in the network. For vehicles which manage to complete their trips between 5pm - 6pm, the total amount of time spent in the network is the difference between the time they enter the network and the time they exit the network. For vehicles which do
Table B.2: Fixed Hyperparameter Values Tested

<table>
<thead>
<tr>
<th>GP Prior</th>
<th>( \mathbf{\nu} )</th>
<th>( \mathbf{\nu} )</th>
<th>( \mathbf{\phi} )</th>
<th>( \mathbf{\phi} )</th>
<th>( \mathbf{\phi} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed-Combined</td>
<td>{0, 1}</td>
<td>{0, 100, 1000, 10000}</td>
<td>{0.1, 0.01, 0.001}</td>
<td>{0.01, 0.001}</td>
<td>{0.001, 0.0001}</td>
</tr>
<tr>
<td>Proposed-Mean</td>
<td>{0, 1}</td>
<td>{0.1, 1, 10}</td>
<td>{0.01, 0.001}</td>
<td>{0.001, 0.0001}</td>
<td>{0.0001, 0.00001}</td>
</tr>
<tr>
<td>Proposed-Covariance</td>
<td>{0, 1}</td>
<td>{0.1, 1, 10}</td>
<td>{0.01, 0.001}</td>
<td>{0.001, 0.0001}</td>
<td>{0.0001, 0.00001}</td>
</tr>
<tr>
<td>Standard</td>
<td>{0, 1}</td>
<td>{0.1, 1, 10}</td>
<td>{0.01, 0.001}</td>
<td>{0.001, 0.0001}</td>
<td>{0.0001, 0.00001}</td>
</tr>
</tbody>
</table>
Table B.3: Pre-selected hyperparameter values

<table>
<thead>
<tr>
<th>GP Prior</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\sigma_0^2$</th>
<th>$\ell$</th>
<th>$\ell^A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>-</td>
<td>10</td>
<td>15</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>Proposed-Covariance</td>
<td>-</td>
<td>10</td>
<td>15</td>
<td>0.5</td>
<td>100</td>
</tr>
<tr>
<td>Proposed-Mean</td>
<td>3</td>
<td>-</td>
<td>15</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>Proposed-Combined</td>
<td>3</td>
<td>-</td>
<td>15</td>
<td>0.5</td>
<td>100</td>
</tr>
</tbody>
</table>

not complete their trips by the end of the simulation (i.e. 6:00pm), their total time spent in the network are counted as the difference between the time they entered the network and 6:00pm. The reason for counting the time spent in the network by vehicles that have not completed their trips is to ensure that gridlocks in the network are penalized. The expected travel time is then taken to be the mean amount of time that each vehicle spends in the network.

Unlike the optimization of the 100-D Griewank function in Section 3.3.4, we decided to optimize the hyperparameters at every iteration, due to the difficulties involved in identifying a good set of fixed hyperparameters when working with a computational-expensive simulator. To prevent the hyperparameter values from diverging, we optimize the hyperparameters by performing maximum likelihood estimation using a fixed, pre-selected set of hyperparameter values (Table B.3) as the initial point at every 5th iteration (i.e. 5th, 10th, 15th, etc.). For all other iterations, the maximum likelihood estimation initial point was taken as the optimized hyperparameter values from the previous iteration.
Appendix C

Appendix of Chapter 4

C.1 List of Notation

\( f \) objective function;
\( f^A \) approximate analytical model of the objective function;
\( F_l \) random variable denoting the stochastic output of a simulation run for time interval \( l \);
\( \alpha_l \) weight coefficient for time interval \( l \);
\( x_l \) vector of decision variables for time interval \( l \);
\( z_l \) vector of endogenous simulation variables for time interval \( l \);
\( b \) vector of deterministic exogenous parameters;
\( \chi \) feasible region;
\( D \) number of dimensions of the feasible region;
\( L \) total number of time intervals;

**Dynamic traffic network model notation:**

\( Q_{ij,l} \) flow from link \( i \) to link \( j \) in time interval \( l \);
\( Q_{in}^{in} \) inflow to link \( i \) in time interval \( l \);
\( Q_{out}^{out} \) outflow to link \( i \) in time interval \( l \);
\( Q_{FD}^{FD} \) flow (as a function of vehicle density) as determined by a fundamental diagram;
\( \gamma_{i,l} \) external arrival rate to link \( i \) in time interval \( l \);
\( \mu_{i,l} \) service rate of link \( i \) in time interval \( l \);
\( \rho_{i,l} \) vehicle density of link \( i \) in time interval \( l \);
\( \rho_{i,\text{max}} \) maximum vehicle density (i.e. jam density) of link \( i \);
\( s \) saturation flow rate;
\( p_{ij} \) probability of vehicles turning from link \( i \) to link \( j \);
\( \Delta t \) time step length;
\( M \) number of time steps in an interval;
$N_{i,l}^{DQ}$ \hspace{1cm} counter for number of vehicle that have reached the downstream end of link $i$ in time interval $l$;

$t_i^{FF}$ \hspace{1cm} free-flow travel time of link $i$;

**Signal control problem notation:**

$F_{L=1}$ \hspace{1cm} random variable denoting the travel time of vehicles in the road network during the second half of the simulation period (when $L = 1$);

$x_{l,j}$ \hspace{1cm} green split of signal phase $j$ in time interval $l$;

$c_i$ \hspace{1cm} cycle time of intersection $i$;

$e_i$ \hspace{1cm} fixed cycle time of intersection $i$;

$x^{LB}$ \hspace{1cm} vector of minimal green splits;

$I$ \hspace{1cm} set of signal controlled intersection indices;

$P(i)$ \hspace{1cm} set of signal phase indices of intersection $i$.

### C.2 Case Study Implementation Details

In Sections 4.4 and 4.4.3, the case study objective function (expected travel time in the last 30 minutes) of Eq. (4.7a) is evaluated by running simulations in Aimsun. In each simulation run, the travel time of each vehicle is taken as the total amount of time that vehicle spent in the network. For vehicles which manage to complete their trips in the last 30 minutes, the total amount of time spent in the network is taken to be the difference between the time they enter the network and the time they exit the network. For vehicles which do not complete their trips by the end of the simulation, their total time spent in the network are counted as the difference between the time they entered the network and the end of the hour of observation. This implementation of counting the time spent in the network by vehicles that have not completed their trips helps to ensure that gridlocks in the network are penalized. The expected travel time is then taken to be the mean amount of time that each vehicle spends in the network. This method of computing the expected travel time from the simulation is the same for both the dynamic and stationary problems.

Similar to the case study in Chapter 3, the hyperparameter optimization is done by performing maximum likelihood estimation with a fixed, pre-selected set of hyperparameter values as the initial point at every 5th iteration (i.e. 5th, 10th, 15th, etc.). In the case of the SAN (resp. MTM), the pre-selected hyperparameter values are shown in Table C.2 (resp. Table C.3). The hyperparameter $\beta$ corresponds to
Table C.2: Pre-selected hyperparameter values (SAN)

<table>
<thead>
<tr>
<th>BO Method</th>
<th>( \beta )</th>
<th>( \sigma_0^2 )</th>
<th>( \ell )</th>
<th>( \ell_{f^A} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla BO</td>
<td>1</td>
<td>5</td>
<td>0.5</td>
<td>–</td>
</tr>
<tr>
<td>Dynamic BO</td>
<td>1</td>
<td>5</td>
<td>0.5</td>
<td>10</td>
</tr>
<tr>
<td>Stationary BO</td>
<td>1</td>
<td>5</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table C.3: Pre-selected hyperparameter values (MTM)

<table>
<thead>
<tr>
<th>BO Method</th>
<th>( \beta )</th>
<th>( \sigma_0^2 )</th>
<th>( \ell )</th>
<th>( \ell_{f^A} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla BO</td>
<td>10</td>
<td>15</td>
<td>0.6</td>
<td>–</td>
</tr>
<tr>
<td>Dynamic BO</td>
<td>10</td>
<td>15</td>
<td>0.6</td>
<td>250</td>
</tr>
<tr>
<td>Stationary BO</td>
<td>10</td>
<td>15</td>
<td>0.5</td>
<td>100</td>
</tr>
</tbody>
</table>

the constant prior mean function (Eq. (3.14)). The rest of the hyperparameters correspond to the covariance function (Eq. (3.16)). For the other iterations, maximum likelihood estimation is performed with the optimized hyperparameter values from the previous iteration taken as the initial point. This procedure helps to prevent the hyperparameter values from diverging.

C.3 Demand Profiles

The external vehicle arrival rates (i.e. demand) to the network can change with time. We consider two different demand profiles – one in which the demand for the bidirectional side road is gradually increasing (Gradually Increasing) and another in which the demand for the main arterial and the bidirectional side road change drastically (Drastic Change). The two demand profiles are summarized in Table C.4. The first column of Table C.4 denotes the origin-destination (OD) pair. The second column shows the demand profile for the Gradually Increasing case. In this case, the demand is constant for all OD pairs except CD and DC, which increase by 150 veh/h every 15 minutes. This results in the demand for CD and DC going from less than the main arterial (600 veh/h compared to 700 veh/h) to more than the main arterial (1050 veh/h compared to 700 veh/h). The last column illustrates the demand profile for the Drastic Change case. In this case, the demand for AB and BA drops significantly after 30 minutes, while the demand for CD and DC increases dramatically at
Table C.4: Demand Profiles

<table>
<thead>
<tr>
<th>OD-Pair</th>
<th>Demand (Gradually Increasing; veh/h)</th>
<th>Demand (Drastic Change; veh/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB/BA</td>
<td>700</td>
<td>1050 → 350</td>
</tr>
<tr>
<td>CD/DC</td>
<td>600 → 750 → 900 → 1050</td>
<td>300 → 900</td>
</tr>
<tr>
<td>EF</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>HG</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>JK</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure C-1: Mean performance at a given iteration for different demand profiles.

...the 30-minute mark. This results in the major direction (i.e. direction with more demand) at the second intersection switching at the 30-minute mark. We evaluate the two demand profiles using a 2 time interval set-up (i.e. \( L = 2 \)).

Figure C-1 plots the optimization performance for both the Gradually Increasing and Drastic Change demand profiles. The x-axis in the plots represents the optimization iteration (i.e. does not count the 4 initial observations). The y-axis shows the objective function estimate of the best solution at a given iteration. Each BO method is run 3 times for 3 different initial sets (i.e. 9 runs in total). Each line in each plot of Figure C-1 denotes the mean of the best solution at a given iteration for 9 BO runs. The shaded regions denote the values ±1 standard error away from the mean. The black dash-dotted line corresponds to Vanilla BO, the blue dashed line to Dynamic BO, and the green solid line to Stationary BO. In these plots, the smaller the y-value reached by the curve, the better the solution is. Similarly, the sooner the curve reaches a small y-value, the more efficient the BO method is deemed to be.
From Figure C-1a, we observe that there is very little difference between the 3 BO methods for the Gradually Increasing demand profile. The 3 BO methods were able to find solutions with similar performance at almost the same time. This is also the case for the Drastic Change demand profile, as seen in Figure C-1b. This shows that Dynamic BO, with its use of the dynamic analytical traffic model, is competitive with Vanilla BO and Stationary BO. In general, the results shown in Figure C-1 suggest that the demand profile may not have much impact on the relative performance of the 3 BO methods, at least for the demand values used in Table C.4 for the SAN.
Bibliography


208


Little JD, 2011 *Or forum—little’s law as viewed on its 50th anniversary*. Operations research 59(3):536–549.


Schultz L, Sokolov V, 2018 Bayesian optimization for transportation simulators. Procedia Computer Science 130:973–978.


Wang JM, Fleet DJ, Hertzmann A, 2005 Gaussian process dynamical models. NIPS, volume 18, 3 (Citeseer).


Zhou T, Osorio C, Fields E, forthcoming A data-driven discrete simulation-based optimization algorithm for large-scale two-way car-sharing net-