Surrogate Model for Real Time Signal Control: Theories and Applications

Shiming XU

Centre for Transport Studies

Department of Civil and Environmental Engineering

Imperial College London

A thesis submitted for the degree of

Doctor of Philosophy of Imperial College London

2020

Declaration of Contribution

I hereby declare that all the work presented in this thesis is carried out by myself. The contributions from my supervisors: Dr Ke Han, Prof. Washington Y. Ochieng and Dr Arnab Majumdar are all in line with the roles and responsibilities for the supervisors defined by Imperial College London. All the work of others included in this thesis are fully cited and referenced.

Shiming XU

Declaration of Copyright

Acknowledgement

First and foremost, I would like to express my dearest thanks to Dr Ke Han for offering me this great opportunity to conduct my PhD study at Imperial College London, and for leading me into the world of academic research. This thesis would not have been possible without his constant support, inspiring advice and encouragement during my four-year study.

I would also like to thank Professor Washington Yotto Ochieng and Dr Arnab Majumdar for being my supervisors in the last year of my PhD, the most difficult time for me as for the most part I had to stay in China due to the COVID-19 pandemic. My most profound appreciation is due to them for reviewing my thesis and making valuable suggestions and critical comments. Without their professional advice, this thesis could not have reached its present form.

I am thankful to all the members of the Department of Civil and Environment Engineering for the pleasant working environment. Special thanks to all my friends in CTS (Centre for Transport Studies): Yang, Junwoo, Lin, Chenyang, Yuanying, Huiqiao, Nan, Hanyu, Top, Jun and Suwan for making my non-scientific life enjoyable. They were always there supporting me when I needed help. I had a wonderful time with them.

I wish to thank my friends in the UK. It was very lucky for us to meet in the UK, and I never felt lonely thanks to their company. Their support and encouragement greatly eased my anxiety and helped me go through this special period smoothly.

Last but not least, I would like to extend my sincere thanks to my beloved family for their unconditional love and understanding throughout the entire PhD research. It was their support that gave me the courage to face any difficulty. I dedicate this thesis to them.

Abstract

Traffic signal controls play a vital role in urban road traffic networks. Compared with fixed-time signal control, which is solely based on historical data, real time signal control is flexible and responsive to varying traffic conditions, and hence promises better performance and robustness in managing traffic congestion. Real time signal control can be divided into model-based and model-free approaches. The former requires a traffic model (analytical or simulation-based) in the generation, optimisation and evaluation of signal control plans, which means that its efficacy in real-world deployment depends on the validity and accuracy of the underlying traffic model. Model-free real time signal control, on the other hand, is constructed based on expert experience and empirical observations. Most of the existing model-free real time signal controls, however, focus on learning-based and rule-based approaches, and either lack interpretability or are non-optimised.

This thesis proposes a surrogate-based real time signal control and optimisation framework, that can determine signal decisions in a centralised manner without the use of any traffic model. Surrogate models offer analytical and efficient approximations of complex models or black-box processes by fitting their input-output structures with appropriate mathematical tools. Current research on surrogate-based optimisation is limited to strategic and off-line optimisation, which only approximates the relationship between decisions and outputs under highly specific conditions based on certain traffic simulation models and is still to be attempted for real time optimisation. This thesis proposes a framework for surrogate-based real time signal control, by constructing a response surface that encompasses, (1) traffic states, (2) control parameters, and (3) network performance indicators at the same time.

A series of comprehensive evaluations are conducted to assess the effectiveness, robustness and computational efficiency of the surrogate-based real time signal control. In the numerical test, the Kriging model is selected to approximate the traffic dynamics of the test network. The results show that this Kriging-based real time signal control can increase the total throughput by 5.3% and reduce the average delay by 8.1% compared with the fixed-time

baseline signal plan. In addition, the optimisation time can be reduced by more than 99% if the simulation model is replaced by a Kriging model. The proposed signal controller is further investigated via multi-scenario analyses involving different levels of information availability, network saturation and traffic uncertainty, which shows the robustness and reliability of the controller. Moreover, the influence of the baseline signal on the Kriging-based signal control can be eliminated by a series of off-line updates.

By virtue of the model-free nature and the adaptive learning capability of the surrogate model, the Kriging-based real time signal control can adapt to systematic network changes (such as seasonal variations in traffic demand). The adaptive Kriging-based real time signal control can update the response surface according to the feedback from the actual traffic environment. The test results show that the adaptive Kriging-based real time signal control maintains the signal control performance better in response to systematic network changes than either fixed-time signal control or non-adaptive Kriging-based signal control.

Table of contents

List of Figures

List of Tables

List of Abbreviations

ADP	Approximation Dynamic Programming
AE	Absolute Error
BITRE	Bureau of Infrastructure, Transport and Regional Economics
CDF	Cumulative Distribution Function
CFP	Cyclic Flow Profile
COP	Controlled Optimisation of Phases
CTM	Cell Transmission Model
CV	Coefficient of Variation
DACE	Design and Analysis of Computer Experiments
DIRECT	Dividing Rectangles
DP	Dynamic Programming
DRO	Distributionally Robust Optimisation
DTA	Dynamic Traffic Assignment
EI	Expected Improvement
FHWA	Federal Highway Administration
GA	Genetic algorithm
IN-TUC	Integrated-Traffic-responsive Urban Control
KPI	Key Performance Indicator
KSI	Fatal/Killed and Serious Injury
LDR	Linear Decision Rule
LKWM	Link-based Kinematic Wave Model
LQ	Linear-Quadratic
MAPE	Mean Absolute Percentage Error
MILP	Mixed Integer Linear Program
min	Minute
MINLP	Mixed Integer Nonlinear Linear Program
MLE	Maximum Likelihood Estimate

MSE	Mean Squared Error
MSRS	Metric Stochastic Response Surface
NDR	Nonlinear Decision Rule
OD	Origin-destination
OPAC	Optimised Policies for Adaptive Control
OPAC-VFC	Virtual Fixed-Cycle OPAC
OSCS	Optimal Sequential Constrained Search
PI	Probability of Improvement
PRODYN	Programmation Dynamique
PRODYN-D	Decentralised PRODYN
PRODYN-H	Hierarchical PRODYN
PSO	Particle Swarm Optimisation
RBF	Radial Basis Function
RE	Relative Error
RHODES	Real time Hierarchical Optimised Distributed and Effective System
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RO	Relative Occupancy
RSM	Response Surface Model
s	Second
SAE	Stacked Autoencoder
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split, Cycle and Offset Optimisation Technique
TABSCO	Telematics Applications in Bavaria, Scotland and Others
TOD	Time-of-day
TRANSYT	Traffic Network Study Tool
TUC	Traffic-responsive Urban Control
UK	United Kingdom
USA	United State of America

Veh	Vehicle
VMS	Variable Message Sign
WEI	Weighted Expected Improvement

Chapter 1

Introduction

This chapter presents a brief background of the research, highlighting its aim and objectives, and concluding with a description of the structure of this thesis.

1.1 Background

Greater urbanisation and rapid population growth worldwide has meant that an increasing number of cities are suffering from severe traffic congestion. BITRE (2015) showed that the annual 'avoidable' social costs of congestion for eight Australian state capitals had increased from $12.8 billion in 2010 to $16.5 billion in 2015. In Schrank, Eisele & Lomax's (2019) report, meanwhile, traffic congestion in urban areas of the USA resulted in a national congestion cost of $179 billion in 2017, growing to $237 billion by 2025. Traffic congestion can also lead to other negative externalities. Wang, Quddus & Ison (2003) found that congestion and frequency of fatal/killed and serious injury (KSI) accidents were positively related, based on data collected on the M25 and surrounding major roads in the UK. Congestion can also result in an increase in vehicle emissions due to factors such as frequent stop-and-go (De Vlieger, De Keukeleere & Kretzschmar, 2000; Choudhary & Gokhale, 2016; Kellner, 2016).

Conventionally, there are three main measures to alleviate traffic congestion: traffic network design, implementation of transport policy, and traffic control management. Since traffic congestion is mainly caused by an imbalance between traffic demand and supply it can be mitigated either by increasing supply or reducing demand (Bagloee & Sarvi, 2016). Inthis context, traffic network design addresses the congestion problem by increasing network capacity through the construction of new, and renovation of legacy, infrastructure. Intuitively, the construction of new infrastructure should be able to solve the congestion problem. In practice, however, it is just a temporary solution, since the balance between demand and

supply in the traffic system will again be disrupted as extra demand is induced by the increase in supply (Goodwin & Noland, 2003). Additionally, the construction of new infrastructure is often restricted by limited land resources in urban areas, high capital costs and disruption to traffic networks during construction.

The implementation of transport policies seek to solve the problem of traffic congestion at its source by regulating traffic demand. Transport policies alleviate congestion by reducing the number of vehicles in the network and redistributing the traffic demand spatially and temporally. Transport policies include Road Space Rationing (Lu, 2015), congestion charges (Bagloee & Sarvi, 2016) and the promotion of public transport (Thøgersen, 2009). The main obstacle to the implementation of transport policy is the lack of public support. This is because of the potential of policy to have a negative impact on the interests of certain sections of society. It is therefore crucial to account for acceptability and equity during the design of a new policy (Wu et al., 2012). Moreover, a traffic network is a delicate system, with the implementation of a new policy causing a chain effect that makes it very challenging to quantify the influence of a new policy and predict whether a desirable outcome can be achieved.

Traffic control management aims to facilitate maximal usage of the existing infrastructure. In contrast to traffic network design, traffic control management is reversible. In addition, it does not cause apparent damage to the interests of anyone and thus faces less resistance than the implementation of a new policy. These characteristics make traffic control management easy to implement and revise regularly. For these reasons, considerable research effort is dedicated to traffic control management as a way of addressing the congestion problem.

Among the variety of traffic control tools available, signal control is most heavily researched. Traffic signals are essential components of traffic systems. The first traffic signal was implemented in 1914, in the USA, initially designed to regulate the safe movement of vehicles in different directions at intersections. After a century of development, current signal control strategies can be classified into:

- Fixed-time signal control, which selects the appropriate signal plan from a set of pre-stored plans according to a given schedule. Since these plans are all generated off-line based on the traffic pattern obtained from historical data, they cannot adapt to short-term fluctuations of traffic flow and can suffer from an 'ageing' problem due to systematic changes in the traffic network. On the other hand, fixed-time signal control does not need any detectors and consequently has a low construction cost, which is the main reason that most of the cities still use it.

- Actuated signal control, which decides whether to extend or terminate the current green stage depending on local vehicle actuation to achieve the efficient use of green time. The current green phase extends when vehicles are detected until a maximum green time is reached or until no vehicle is detected on that approach. The signal plan is only adjusted by traffic demands passively and locally, however, rather than according to a given objective. Actuated signal control performs best under light traffic.

- Real time signal control, which adjusts its signal plan according to the real time traffic conditions. Real time traffic data collected by detectors, not limited to vehicle actuation, are input to the embedded signal optimisation algorithm for decision making, such that the new signal plan can achieve its pre-set objective under the current traffic conditions.

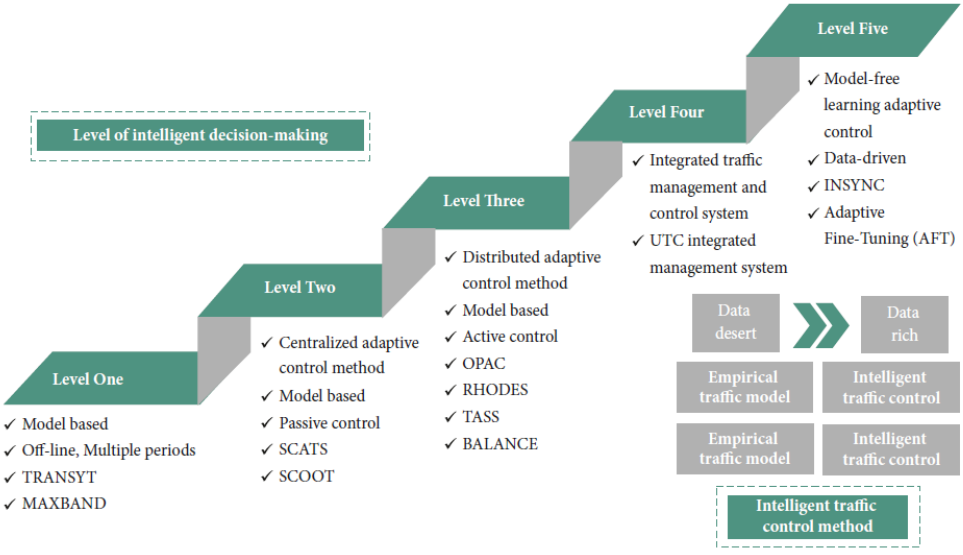


Figure 1.1 Development of real time signal control (Wang et al., 2018)

Of these three types of signal control, real time signal control is more flexible and effective in accommodating unpredictable traffic conditions, and it is for this reason that real time signal controls have been heavily researched in recent years. Real time signal control can be used to optimise a variety of objectives, including travel time reduction (Li et al., 2004; Sun et al., 2006; Christofa & Skabardonic, 2011), throughput maximisation (Liu et al., 2015) and emission minimisation (Rakha et al., 2000; Li et al., 2004; Han et al., 2016a).

Wang et al. (2018) summarised the development of real time signal control and divided it into five generations, as shown in Figure 1.1, based on the work of Gartner, Stamatiadis & Tarnoff (1995). The first generation is fixed-time signals that can only alter the signal plan according to a fixed schedule, and where no signal plan can be generated on-line. The second generation can be considered as the beginning of real time signal control. From this generation, signal control systems could change the signal plan on-line according to the prevailing real time traffic conditions. The third generation has a similar mechanism to that of the second generation, with the only difference being that this (third) generation allows the signal plan to change continuously in an acyclic manner. Besides the three generations summarised by Gartner, Stamatiadis & Tarnoff (1995), Wang et al. (2018) added another two generations. In the fourth generation, real time signal control is integrated with other traffic control management strategies such that they can cooperate efficiently. The fifth generation involves a significant innovation, being the start of model-free real time signal control.

According to the evolution of signal control discussed above, real time signal control can be classified into model-based approaches (from generation 2 to generation 4) and model-free approaches (generation 5) depending on whether a traffic model is required to interpret the traffic dynamics during the optimisation.

The model-based approach requires a traffic model to interpret the traffic dynamics of the network such that network performance under different signal plans can be estimated. Therefore, the performance of model-based signal control depends on the validity and accuracy of the underlying model. It is challenging, however, to construct an accurate traffic model that can well capture the complex interactions between the various components in the

traffic system with incomplete knowledge of the underlying problem. In addition, the embedded traffic model in model-based real time signal control limits the flexibility in the selection of inputs and objectives. Moreover, there is no systematic guideline for the calibration of model-based signal control, meaning that calibration needs to be carried out manually through a trial-and-error process, which acts as a barrier to frequent updates to the underlying model.

On the other hand, model-free real time signal control generates signal control decisions wholly based on expert experience and empirical observations, without being informed by any traffic model. At present, the research on model-free real time signal control is mainly rule-based and learning-based. In the case of the former, the signal plan is altered when a specific pre-determined rule is activated. It is often difficult to determine the appropriate rules for the traffic network, however, which limits the applicability of rule-based approaches. This drawback can be addressed by the learning-based approach. Learning-based signal control learns the interaction between traffic states and signal control directly. Moreover, it can update itself based on feedback from the traffic environment. The learning-based approach, however, lacks interpretability, and its reliability and transferability remain questionable due to the absence of analytical insights.

Another model-free method, still to be applied to real time signal control, is surrogate modelling. A surrogate model is built with sample data and yields insight into the functional relationship between the inputs and outputs (Barton & Meckesheimer, 2006; Jakobsson et al., 2010; Chen et al., 2014a) through an analytical and tractable mathematical representation, such that the value of any point can be estimated without the need for a traffic model. A surrogate model can replace the actual model or the physical process (Simpson et al., 2001; Gartiselov, 2012) while providing computational efficiency and mathematical tractability.

Wang et al. (2018) highlighted that the future trend in real time signal control would be model-free instead of model-based, using real time monitoring data instead of forecast data and automatically controlled instead of manually controlled. This shows that model-free real time signal control has great potential and is worthy of further investigation.

1.2 Aim and objectives

Given the limitations of existing real time signal control approaches discussed above, this research aims to propose a new real time signal control system based on surrogate modelling, which has the potential to improve network-level performance in a real time computational environment. The following objectives have been formulated to deliver the aim of this thesis.

1. Undertake a critical review of the literature on existing real time signal control systems and methods to identify their limitations.

2. Undertake a critical review of existing surrogate modelling techniques and surrogate-based traffic optimisation to illustrate the potential and feasibility of combining surrogate modelling with real time signal control and optimisation.

3. Develop a framework to interpret the state-control-objective relationship analytically using Kriging as the surrogate model, and develop an infill and training strategy to construct the Kriging-based response surface. The resulting surrogate model will be used for real time signal optimisation, in such a way that it can quickly generate near-optimal signal control parameters with given traffic states as the input.

4. Based on the surrogate model developed in Objective 3, perform real time optimisation via a modified Particle Swarm Optimisation (PSO) algorithm, which offers an efficient trade-off between computational time and solution optimality, and account for the estimation error of the surrogate model within the optimisation procedure.

5. Test, validate and assess the surrogate-based real time signal control developed in Objectives 3 and 4, using traffic simulations of a real-world road network, and compare the results with the benchmark. Assess the robustness of the proposed real time signal control by applying it under different scenarios with variable levels of information availability, demand and demand variability.

6. Identify the influence of the initial baseline signal used for surrogate modelling in

Objective 3 and real time signal control in Objective 4. Develop an adaptive baseline signal update strategy to mitigate the influence of the initial baseline signal.

7. Develop a surrogate-based real time signal control with an adaptive response surface that can self-update according to the feedback from the traffic network, such that it can accommodate systematic changes in demand.

1.3 Thesis structure

The thesis consists of six chapters, organised as follows:

Chapter 2 reviews six developed real time signal control systems and introduces their operational mechanisms. The current research on real time signal controls is classified according to their optimisation methods. The advantages and drawbacks of each method are reviewed. Then it reviews the state-of-the-art in surrogate modelling techniques and presents a detailed description of the three most widely-used models: the Response Surface Model, Radial Basis Function and Kriging. Three main approaches to the infill strategies are presented, forming the basis of the selection of the surrogate model and infill strategy in ***Chapter 3***. ***Chapter 2*** also reviews the literature related to surrogate-based optimisation, highlighting that the surrogate model can approximate the complicated control-objective relationship accurately and reduce the optimisation time. This leads to the conclusion that surrogate modelling has great potential for application to a real time signal control system.

Chapter 3 compares the surrogate modelling techniques for the fixed-time and real time problems, and identifies the challenges of constructing a response surface to approximate the state-control-objective relationship for the real time control problem. Based on the surrogate modelling techniques reviewed in ***Chapter 2***, the conventional surrogate modelling techniques are modified to construct a surrogate model for real time signal control. The whole process includes a sampling of traffic state variables and signal control variables, surrogate model construction and infill strategy. The approximation accuracy and estimation efficiency of the proposed surrogate modelling technique is assessed by carrying out a test on a small

network.

Chapter 4 proposes a surrogate-based real time signal control system that applies the surrogate model prepared in ***Chapter 3***. It starts with the introduction to the standard process of Particle Swarm Optimisation (PSO). A modified Particle Swarm Optimisation (PSO) is specially designed for the surrogate-based real time signal control system to account for the estimation error and reduce the convergence time. The performance of the proposed real time signal control system on network management is assessed in terms of total throughput, average delay and weighted throughput. Multi-scenario analysis is conducted to explore the performance of the proposed surrogate-based real time signal control more fully under different scenarios, including the number of traffic variables, demand level and demand variability, as well as the influence of the sensitivity to estimation error and the baseline signal.

Chapter 5 proposes an adaptive baseline signal update strategy to solve a problem identified in ***Chapter 4***, namely a deterioration in the performance of the proposed real time signal control due to the improper selection of a baseline signal. Moreover, the real time signal control developed in ***Chapter 4*** has a fixed response surface, which limits its ability to adapt to systematic changes in demand. By identifying the possible influences of the systematic changes of demand on real time signal control, a surrogate-based real time signal control with an adaptive response surface is developed, which can update the surrogate model and baseline signal according to the information collected during the optimisation. Sensitivity analysis is conducted to assess the influence of two major factors in the response surface update strategy: update frequency, and the number of infill points.

Chapter 6 revisits the objectives listed in ***Chapter 1*** and concludes the research undertaken in this thesis. Furthermore, recommendations for future research are presented.

Chapter 2

State-of-the-art in real time signal control and surrogate modelling

This chapter first reviews the state-of-the-art of real time signal controls in Section 2.1 to provide an overview of the current status of their development. The underlying operational mechanisms are discussed to show the advantages and disadvantages of each method. Then, in Section 2.2, a review of surrogate models is presented to compare the different surrogate modelling techniques and infill strategies, thus showing the potential to integrate surrogate models with real time signal control. Following this, Section 2.3 introduces the surrogate-based optimisation applications in the context of traffic control so as to confirm that surrogate-based real time traffic control is reasonable.

2.1 Real time signal control

This section reviews the state-of-the-art of real time signal control. It starts with an introduction to two fundamental features of real time signal control. Section 2.1.1 reviews the six most widely-used real time signal control systems, while section 2.1.2 reviews theoretical studies on real time signal controls.

Figure 2.1 shows an overview of the structure of real time signal control, which defines how multiple intersections are controlled by the same real time signal control system. The structure of real time signal control can be broadly divided into isolated and coordinated controls, which are distinguished by whether the interaction between intersections is considered when making decisions for each intersection. Since all intersections are controlled independently, isolated signal control is easier to implement than coordinated control, but is incapable of managing traffic flow at a network level.

Depending on whether all the decisions are made by a central computer or intersection controllers, coordinated control can be further classified into

- Centralised control
- Distributed control
- Hierarchical control.

In centralised control, a central computer is responsible for the decision making across the entire network through a centralised control algorithm that seeks to maximise network-level performance. All the data collected by local controllers are transmitted to the central computer for processing, and then once decisions are made, they are sent back to each controller. This operational mechanism makes centralised real time signal control sensitive to the working state of the central computer and transmission system. Any failure of the central computer or transmission system results in the whole or part of the traffic network being forced to adopt a backup signal plan, which is usually non-adaptive.

In contrast, the distributed system is more robust since intersection controllers make decisions, based on the local traffic conditions and information provided by adjacent intersections. A distributed system can be easily expanded but cannot be applied to accommodate a network-level objective. Hierarchical systems have the advantages of both centralised and distributed systems. Its controllers are multi-level and have their own control algorithms, and the parameters can be determined at various levels.

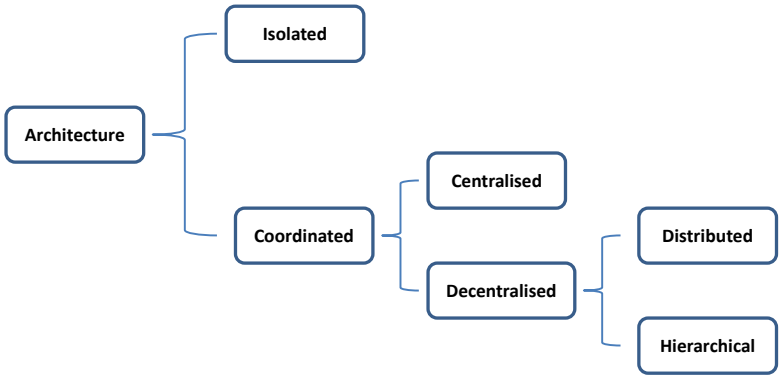


Figure 2.1 An overview of real time signal control structures

2.1.1 Developed real time signal control systems

Real time signal control systems have been extensively developed and implemented, given their potential for regulating the traffic flow and hence improving network performance. Most of these developed real time signal control systems use traffic models to estimate the arrival of vehicles and to choose between alternative decisions. In order to calibrate a real time signal control system, therefore, it is necessary to understand traffic dynamics.

These real time signal control systems differ in system type, objectives and optimisation mechanisms. In the following part, a review is undertaken of the most widely-used real time signal control systems: the Sydney Coordinated Adaptive Traffic System (SCATS) (Sims & Dobinson, 1980; Lowrie, 1982); the Split, Cycle and Offset Optimisation Technique (SCOOT) (Hunt et al., 1981; Hunt et al., 1982); Traffic-responsive Urban Control (TUC) (Diakaki, Papageorgiou & McLean, 1999); Optimised Policies for Adaptive Control (OPAC) (Gartner, 1983); Programmation Dynamique (PRODYN) (Henry, Farges & Tuffal, 1983; Farges, Khoudour & Lesort, 1990) and the Real time Hierarchical Optimised Distributed and Effective System (RHODES) (Mirchandani & Head, 2001). Their fundamental characteristics are summarised in Table 2.1.

	System Type	Objective	Arrival Prediction	Cyclic	Decision Type	Optimisation Methodology
SCATS	Distributed	Minimise delay/stop or Maximise throughput	×	✓	Predetermined signal plan	Plan selection
SCOOT	Distributed	Minimise queue length and stop	✓	✓	Incremental change of cycle time, split and offset	Hill Climbing
TUC	Centralised	Minimise the risk of saturation	×	✓	Cycle time, split and offset	Linear-Quadratic regulator/

		and spillback				Feedback-based algorithm/ Decentralised feedback control
OPAC	Hierarchical	Minimise delay and stop	✓	×	Phase switching plan	Optimal Sequential Constrained Search/Dynamic Programming
PRODYN	Hierarchical/ Distributed	Minimise delay	✓	×	Phase switching plan	Forward dynamic programming
RHODES	Hierarchical	Minimise delay, stop and queue	✓	×	Phase switching plan	Dynamic programming

Table 2.1 A review of developed real time signal control systems

Sydney Coordinated Adaptive Traffic System

The Sydney Coordinated Adaptive Traffic System (SCATS) was developed by the Roads and Traffic Authority of New South Wales, Australia, for signal control in Sydney (Sims & Dobinson, 1980; Lowrie, 1982). It can adjust the cycle length, phase split and offsets according to real time traffic data, and adapt to time-varying traffic conditions. It selects the signal plan that optimises the objective from a suite of pre-determined plans stored in its database. SCATS adopts three different objectives depending upon the traffic conditions, respectively: minimisation of stops under light traffic, minimisation of delay under medium traffic, and maximisation of throughput under heavy traffic.

SCATS uses a central computer, regional computers and intersection computers to control a large-scale network at three levels. Figure 2.2 shows the communication between the three control levels. Typically, the central computer does not participate in the normal operation of signal control, but only stores all the traffic data in the database and monitors the status of all components in the system. It can, however, manually adjust the control parameters, dynamic

functions and mode of the lower level computers, and even realise remote control if necessary. This hierarchical structure provides the central computer with the capability to check all the information stored in the lower level controllers so as to monitor the operational status of the entire system.

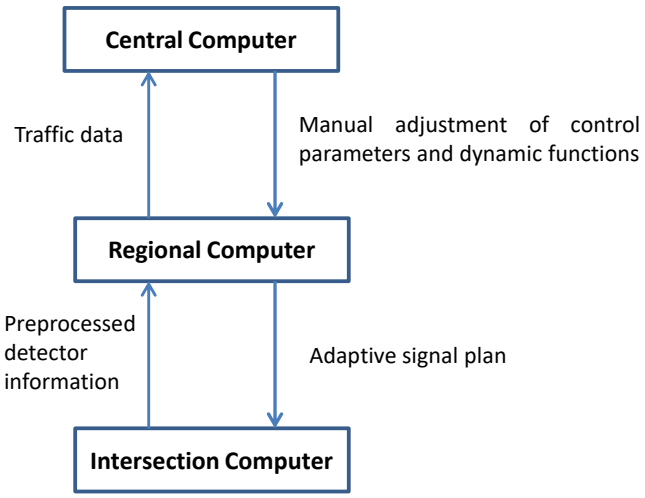


Figure 2.2 Hierarchical system of SCATS

Only the regional computers can make real time decisions according to the pre-processed information provided by the intersection computers. A region controlled by the same regional computer is called a system. As each regional computer only decides the signal setting for the intersections in the system it controls, SCATS is hence a distributed system, even though it has a hierarchical structure. It can be divided into several subsystems, including 1 to 10 intersections. Each subsystem works as the smallest control unit, assuming that the traffic condition within it is uniform. The allocation of subsystems is decided and stored in advance, according to the level of interdependence between signalised intersections. Although the signal plan is determined for each subsystem rather than for each intersection, SCATS does allow the intersection computer to make some small adjustments, known as tactical control. The minimum cycle length, maximum cycle length and geometrically optimal cycle length are defined for each subsystem. The subsystems can be linked with other subsystems that have similar traffic conditions, and the linked subsystems can be split if necessary. Four linking plans are provided to define the conditions for subsystem linkage.

Each regional computer selects the most appropriate cycle length, green split plan, offset plan and linking plan for its corresponding subsystem. The cycle length can control the degree of saturation within the subsystem, whereas the green split balances the degree of saturation at the critical intersection. Offsets can enhance the objectives indirectly by forming a platoon of vehicles.

SCATS, however, does not contain any optimisation algorithm and instead compares the performance of the stored plans based on the real time traffic conditions to make a decision.

Split, Cycle and Offset Optimisation Technique

The Split, Cycle and Offset Optimisation Technique (SCOOT) was developed by the Transport and Road Research Laboratory (TRRL) in the UK in 1980 (Hunt et al., 1981; Hunt et al., 1982). It is a distributed real time adaptive signal control system with the same underlying principles as the Traffic Network Study Tool (TRANSYT), which is a traffic simulation and signal optimisation program developed by Robertson (1969). TRANSYT can only achieve signal optimisation off-line, while SCOOT, its extension, can update the signal plan on-line to adapt to time-varying traffic conditions (Robertson, 1986; Robertson & Bretherton, 1991). Unlike other signal control systems, SCOOT changes the split, cycle time and offset incrementally based on short-term flow variations. The fact that these changes are small and incremental, however, limits SCOOT's adaptability.

SCOOT makes its decisions according to two performance indicators: total queue length and number of stops. The delay is closely related to queue length, and is reduced at the same time as the queue length decreases. The number of stops affects energy consumption, safety and driving experience. The weighted sum of these two indicators (queue length and number of stops) is expressed as the overall performance indicator (i.e. objective) of SCOOT, and the weighting parameter is user-defined.

SCOOT uses sensors positioned at the upstream end of each link to collect traffic data. The data collected is then converted into the Cyclic Flow Profile (CFP), which is the average traffic flow passing through the detectors in each time step of the cycle. For SCOOT, cycle

time is divided into several four-second time steps, and the CFP can be periodically updated. The CFP is used to simulate vehicles approaching the intersections downstream. The impact of any signal plan change can be estimated by the queue size, and the queue discharge can be simulated easily given the estimated arrival pattern. SCOOT optimises the split, cycle time and offset separately, following different rules in each of these three optimisation processes.

The split optimiser aims to find the optimal way to allocate the green time to each approaching link to an intersection. It compares the degree of saturation for each link, that is, the ratio of the demand flow to the maximum possible discharge flow. The signal plan is revised to minimise the degree of saturation at each intersection. Another aspect that needs to be taken into account is congestion, with congested links being given a longer green time. All the decisions are made a few seconds before the phase transition, allowing up to four seconds' alteration to optimise the overall performance indicator.

The offset optimiser estimates the link delay and stops for all the upstream and downstream links connected to the intersection, selecting the offset that can minimise the total delay and stops. The selection of offset also accounts for the congestion by giving congested links higher priority. The offset is revised once a cycle.

The cycle time can be extended when the intersection has a high degree of saturation and reduced otherwise. This decision is made by the cycle optimiser. The cycle time must not be greater than the maximum permitted cycle time, nor shorter than the minimum practical cycle time of the critical intersection in the region. The minimum practical cycle time of each intersection is also updated periodically based on the degree of saturation. If any link to the intersection has a higher degree of saturation than the ideal saturation level, the minimum practical cycle time increases by a small amount. Meanwhile, if all the links have a degree of saturation below the ideal saturation level, the minimum practical cycle time is reduced slightly. The cycle time can be adjusted by 4, 8 or 16 seconds within the boundary. The cycle optimiser is the only one of SCOOT's three optimisers that can be used to deal with large changes in traffic conditions.

Traffic-responsive Urban Control

Traffic-responsive Urban Control (TUC) was developed as part of the Integrated-Traffic-responsive Urban Control (IN-TUC) for the M8 motorway corridor in Glasgow, and belongs to the European Telematics Applications in Transport project TABASCO (Telematics Applications in Bavaria, Scotland and Others) (Diakaki, Papageorgiou & McLean, 1999). IN-TUC combines ramp metering, variable message signs and signal control. In this system, only split optimisation is considered in signal control (Diakaki, 1999) but TUC has subsequently been extended to include separate cycle and offset optimisations (Diakaki et al., 2003).

In order to describe the traffic flow process, TUC adopts the store-and-forward modelling method, which simulates traffic dynamics in a continuous manner without using discrete variables (Aboudolas, Papageorigiou & Kosmatopoulos, 2009). For simplicity, the outflow of each link is limited to binary variables equal to the saturation flow, or zero. In other words, when the link has the right to move, and downstream has space to accommodate the released traffic, the outflow is equal to saturation flow; otherwise, it is zero. This means that the average outflow in each cycle can be easily calculated in this model. The store-and-forward approach is similar to the cell transmission model (CTM) but more straightforward. The time step required must not be shorter than the cycle time, and each link is regarded as a cell. Even though it ignores the within-link traffic flow dynamics and short-term queue length fluctuations due to signal changes, it reduces computational complexity.

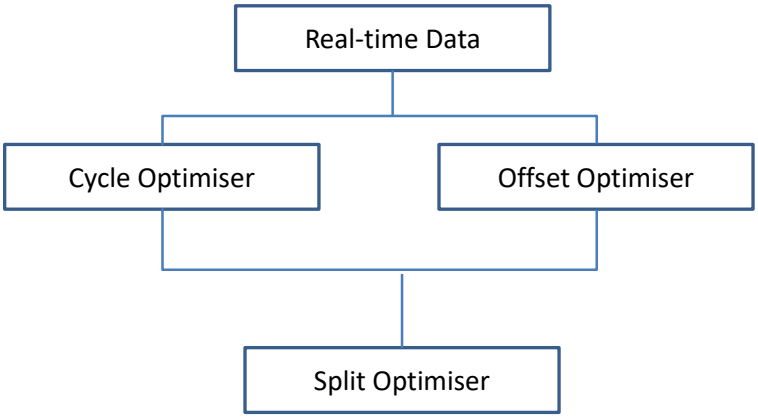


Figure 2.3 Optimisation system of TUC

As shown in Figure 2.3, the data collected are first input into a cycle optimiser and offset optimiser. With the obtained cycle time and offset, the optimal split can be determined. TUC assumes that the entire network shares the same cycle time. The cycle optimiser adopts a feedback-based algorithm. This is based on the concept that a longer cycle time will increase the capacity of the network since the proportion of lost time is reduced, but when the network is undersaturated, it leads to a longer waiting time, which can cause an extra delay. The actual cycle time is therefore half of the calculated cycle time when all the links to the intersection have a low saturation flow.

The offset optimiser aims to form green waves along the arterials. The offset optimisation is realised in a decentralised manner by calculating the offset of each couple of adjacent intersections on the same arterial individually. For each pair of intersections, two waves—the flow wave due to the upstream signal change, and the kinematic wave due to the downstream signal change are considered, together with the current queue length. The offset is the weighted sum of the time that the two waves arrive at the end of the queue. In order to realise the coordination between intersections based on the offset, a temporal transient cycle time is adopted for the intersections along the arterial.

Given the cycle time and transient cycle time, the Linear-Quadratic (LQ) approach (Diakaki, Papageogiou & Aboudolas, 2002; Dinopoulou, Diakaki & Papageogiou, 2006) is used to determine the green split based on the current traffic state and a pre-defined fixed signal plan. LQ optimisation also needs a constant feedback gain matrix. Although the construction of this gain matrix is time consuming, especially for a large-scale traffic network, it is calculated off-line and hence does not affect the efficiency of on-line optimisation. The ultimate goal of the split optimiser is to minimise the risk of oversaturation and queue spillback, which can be achieved by minimising and balancing the relative occupancies of the links connected to each intersection.

Optimised Policies for Adaptive Control

The Optimised Policies for Adaptive Control (OPAC) was developed by the University of

Massachusetts, Lowell and sponsored by the US Federal Highway Administration (FHWA) (Gartner, 1983). It is a truly demand-responsive real time signal control that can adapt to the time-varying traffic conditions. In addition, this system is under acyclic control, meaning that it is unconstrained by the fixed control period, a rigid structure and extrapolation of existing concepts. Furthermore, dynamic programming (DP) (Bellman, 1957) converts the signal optimisation problem into a direct minimisation of a performance indicator, that is, the weighted sum of delay and stops. The development of OPAC has experienced four main generations, from fixed time to real time control, and from an isolated intersection to a coordinated network (Gartner, Pooran & Andrews, 2001; Gartner, Pooran & Andrews, 2002).

OPAC-1 is the first generation that uses DP with an infinite horizon and is designed for isolated control. The inputs to OPAC-1 are the initial queues of each link. The entire horizon is divided into infinite time steps lasting five seconds. OPAC-1 assumes that the arrivals are known, such that backward DP can be applied. In practice, however, it is impossible to predict the arrivals over the entire control period. Another reason that limits OPAC-1 to a fixed-time control is the extensive computational resources required by the need to evaluate all the possible phase switching combinations. While OPAC-1 is therefore not applicable for real time signal control, it is a good benchmark for future generations.

In order to reduce the computational burden, OPAC-2 accommodates certain simplifications to reduce the size of the state space. Specifically, the control period is divided into several stages with n intervals, which are typically two to five seconds. Furthermore, instead of using the phase duration of decision making as in OPAC-1, OPAC-2 determines the switching time, in terms of a time step, so as to minimise the delay over the whole stage. The optimisation problem is solved by an optimal sequential constrained search, which chooses the optimal solution from all possible combinations. In order to reduce the number of possible combinations, phase switching can occur at least once and up to three times for each stage. Even though OPAC-2 is more time-efficient than OPAC-1, the requirement for a complete knowledge of the arrivals in the entire stage limits its applicability to real time control.

OPAC-3 is the first version capable of real time signal control. It adopts the rolling horizon

method to relax the requirement relating to the full knowledge of arrivals. OPAC-3 was first designed for a two-phase operation and was extended to an eight-phase operation. This method allows signal plans to be revised regularly.

The first three generations of OPAC are designed for isolated intersections only, with the fourth generation, however, OPAC was extended to coordinated networks. OPAC-4 is a hierarchical system, also known as the virtual fixed-cycle OPAC (OPAC-VFC). It has three layers: synchronisation, coordination and local control layers. The synchronisation layer is at the highest level and generates the virtual cycle length for the entire network or a group of intersections. On the premise of ensuring progression capabilities between adjacent intersections, the virtual cycle length is determined to ensure that the critical intersection has sufficient capacity. The coordination layer determines the offset of each intersection by taking into account the primary signals of adjacent intersections. With the virtual cycle length and offsets, the local control layer can generate the optimal phase switching plan for each intersection in a manner similar to OPAC-3.

Programmation Dynamique

The Programmation Dynamique (PRODYN) is a real time signal control system developed in France (Henry, Farges & Tuffal, 1983; Farges, Khoudour & Lesort, 1990). It solves the real time signal optimisation problem at an intersection level with a modified forward dynamic programming method. There are two versions: the hierarchical PRODYN (PRODYN-H) and the decentralised PRODYN (PRODYN-D). While both versions adopt forward dynamic programming, they differ in how the intersections in the network are coordinated.

PRODYN-H is a two-level hierarchical system. The control optimisation of each intersection is realised by the lower level, while the coordination of the entire network is achieved by the upper level. It uses iteration processes to solve the signal optimisation problem, and although this approach can find the optimal signal plan, it is computational demanding, meaning that the network size has to be limited to ten intersections.

PRODYN-D is a distributed system. It is more time-efficient than PRODYN-H since it does

not have a centralised layer, but allows the adjacent intersections to exchange information. The level of communication can be controlled by the information shared at the network level. For the lowest level of communication, the simulation output of the upstream intersection is transmitted to the downstream intersection controller in order to estimate arrivals. Barriere, Farges & Henry (1986) detail all the levels.

Real time Hierarchical Optimised Distributed and Effective System

The Real time Hierarchical Optimised Distributed and Effective System (RHODES) (Mirchandani & Head, 2001) is a hierarchical real time adaptive signal control system. The signal optimisation problem is decomposed into several subproblems and solved at each level, as shown in Figure 2.4. This hierarchical structure allows RHODES to respond to different sources of stochasticity in the network, such as trip generation, route selection, probabilistic events (e.g. closure, accidents), and the behaviours of pedestrians and drivers. Since these sources of stochasticity can have an impact on the traffic network lasting from a few seconds to a few weeks, RHODES has an established hierarchical structure where each level deals with the uncertainties at a different time scale.

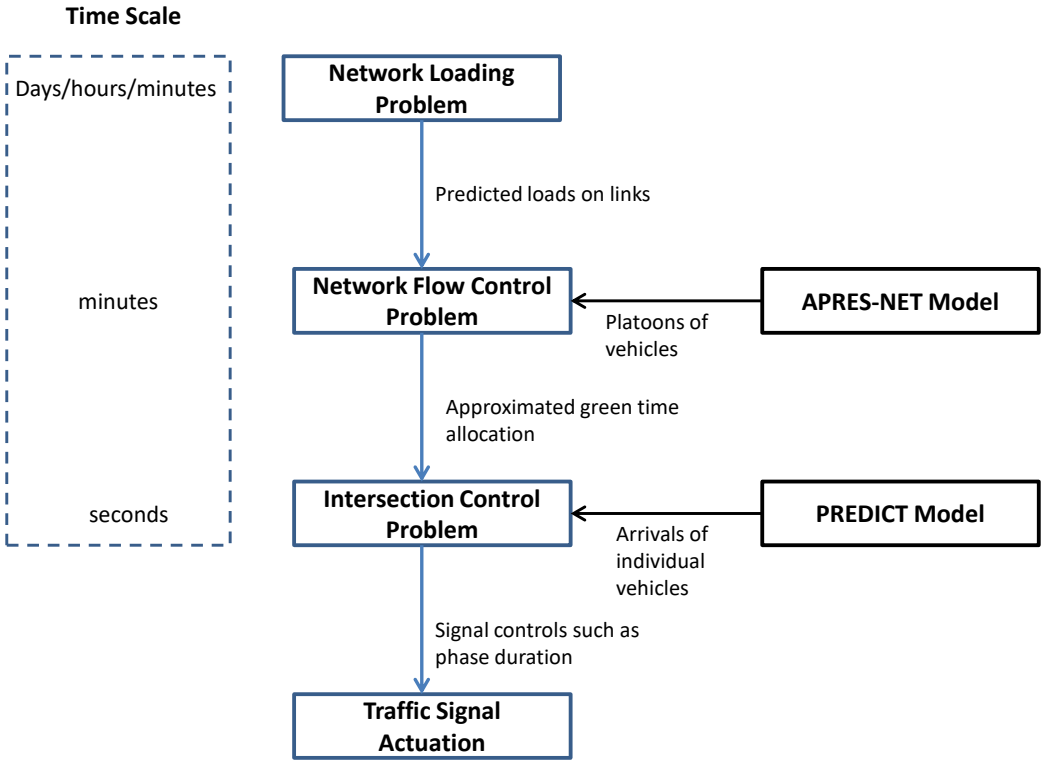


Figure 2.4 System structure of RHODES

The planning level is the highest level and solves the network loading problem. It simulates the traffic flow assignment under stochastic equilibrium, i.e. how drivers choose their routes based on the knowledge of travel time, delays and network characteristics. The predicted link load, i.e. the output of the planning level, is passed on to the lower levels. The planning level is the level with the longest time scale, lasting from hours to weeks. It can show the long-term influence on network loading due to changes in network design or traffic flow variation.

The second level solves the network flow control problem. It uses the APRES-NET model (Dell'Olmo & Mirchandani, 1996) to estimate the platoon flow and speed, and the REALBAND model (Dell'Olmo & Mirchandani, 1995) to deal with the conflicting signal requirements of different platoons approaching the same intersection. The second level can control the platoon movements by deciding which platoon is given the right to move, thus minimising the delay and stops. It manages the traffic flow of the whole network, and the decisions of approximated green time allocations made at the second level are the constraints input to the lowest level.

The lowest level addresses the intersection control problem. Each intersection makes its own decision independently based on the observed and predicted arrivals of individual vehicles generated by the PREDICT model (Head, 1995). Intersections do not communicate with each other, and it is the constraints from the second level that realise network-wide coordination. The phase duration is obtained by the Dynamic Programming-based COP (Controlled Optimisation of Phases). For each time interval, a decision on whether to switch to the next phase is made, and it is these decisions that eventually determine the final phase duration that is to be implemented. All these decisions are input into Traffic Signal Actuation, which is the signal controller of each intersection.

2.1.2 Theoretical research on real time signal control

In addition to the real time signal control systems that have been developed and implemented, researchers continue to develop new methods for real time signal control for a variety of

purposes. These real time signal controls can be classified into two categories: model-based and model-free approaches. Model-based approaches utilise a traffic model to evaluate and compare the performance of feasible solutions to support decision making under those specific conditions, while model-free approaches describe or learn the rules of the system directly without recalling a traffic model during the optimisation. In this thesis, optimisation is defined as the approach to find a quite good (i.e. near-optimal solution) or optimum solution to the proposed problem (Schneider & Kirkpatric, 2006).

Model-based approaches

In model-based approaches, the traffic model can be formulated as a mathematical formulation and solved by an exact optimisation approach. The centralised real time signal control problems under a macroscopic simulation environment (i.e. Cell Transmission Model (CTM)) can be formulated as a Mixed Integer Linear Program (MILP) with binary variables and easily solved by commercial solver CPLEX easily (Lin & Wang, 2004). Han et al. (2016a) proposed a MILP as the formulation of a link-based simulation model for a centralised real time signal optimisation problem with traffic-driven emission as extra constraints. Under the consideration of the error due to the reformulation of emission to a linear constraint, MILP was solved by robust optimisation. The proposed system was tested on a hypothetical network with only ten links. Christofa, Papamichail & Skabardonis (2013) formulated the signal control problem as a Mixed Integer Nonlinear Program (MINLP), which could be solved by branch-and-bound. Their research was limited to an isolated intersection under undersaturated conditions, however, with the traffic dynamics of the intersection simulated based on queuing theory.

The traffic model in model-based approaches can also be a simulation model. To ensure computational efficiency, such a model is always solved by a heuristic optimisation method which can only lead to a near-optimum rather than a true optimum solution. This heuristic optimisation approach is suitable for solving large and complex problems, and one such method that has been extensively used for real time signal control is Genetic Algorithm (GA).

GA can realise centralised real time signal control by using a link-based simulation model to

evaluate the performance of the whole network under each feasible solution. Girianna & Benekohal (2004) used a simple GA to solve the centralised real time signal control optimisation problem of an arterial network with 20 signalised intersections according to the net effect of released vehicles and the disutility function. The constraints of signal variables were transformed into penalty functions. In order to reduce the convergence time, only a small number of populations were generated with the optimisation process being rerun if convergence could not be achieved. GA can also solve the acyclic signal control problem. Lee et al. (2005) developed a GA-based acyclic adaptive signal control system. Efficiency was ensured by calculating the vehicle delay with a simplified link-based simulation model that assumed that all vehicles move with a cruise speed and that followed the First-In-First-Out principle.

Sun, Benekohal & Waller (2006) constructed a bi-level problem considering dynamic traffic assignment. The upper level represented the decision-making process, and was solved by GA, while the lower level captured the user travel behaviour based on the reactive dynamic stochastic user optimal principle. Route choice was modelled by the use of Incremental Logit Assignment. This bi-level problem is time-consuming since the lower level problem needs to be solved a large number of times during the optimisation of the upper-level problem. This proposed real time signal control was operated in a centralised manner to minimise the overall travel time of the network.

In summary, the heuristic optimisation approach can deal with the real time optimisation problem of a large-scale traffic network in a centralised manner, and more efficiently than the exact optimisation approach. Nonetheless, the fact that it needs to calculate the fitness value (i.e. the objective function) a considerable number of times with the embedded traffic model means that it can only be used with macroscopic traffic models such as CTM, or link-based models.

Model-free approaches

Model-free approaches are totally independent from the interpretation of traffic dynamics in advance. They either describe the rules of controls directly (rule-based approaches) or learn

the mapping between traffic state and controls (learning-based approaches).

Rule-based approaches typically do not have an explicit objective to optimise. Dion & Hellinga (2002) pointed out that the traffic network actually operates under certain rules, and signal changes usually occur after some specific events; an observation that inspired the development of rule-based optimisation. Rule-based optimisation techniques are simple and straightforward to understand, such that they can solve the problem of high computational demand, especially when there are many variables. Nonetheless, it is difficult to express the activation conditions for a signal change in a clear and exact numerical way, and thus activation rules and conditions are expressed more vaguely using fuzzy logic.

By using prespecified rules, fuzzy logic converts traffic states to the degree of membership, which contributes to the decision making at the intersection level in terms of phase selection (Wei et al., 2001), cycle time, phase split and offset (Chiu & Chand, 1993) and phase sequence and phase time (Murat & Gedizlioglu, 2005). Fuzzy logic can be integrated with a multi-agent control scheme where a specific set of rules is assigned to each agent (i.e. to traffic flows following the same signal) and then the final decision is made by allowing all the agents to negotiate with each other (Kosonen, 2003). Although the test is carried out at the level of an individual intersection, coordination of the signals can be achieved by increasing the neighbourhood area.

Bingham (2001) pointed out that the conventional fuzzy logic system was not adjustable, and its parameters could not change according to the time-varying traffic condition. Accordingly, a neurofuzzy approach was developed that combined neural networks and fuzzy logic. In this approach, the four parameters of the trapezoidal membership function were updated by the neural network in order to adapt to different traffic volumes.

These rule-based optimisation methods do not need a traffic model but instead rely on expert knowledge to identify the appropriate rules. Moreover, those rules can only show how the traffic should operate, but cannot achieve a maximisation/minimisation of a given objective. These weaknesses have led scholars to focus increasingly on learning-based approaches that

directly study the relationship between the traffic state and the corresponding optimal solution. This learning procedure can be conducted off-line, and once the relationship is established, on-line optimisation becomes time-efficient.

The most widely-used methodology is reinforcement learning (RL), which aims to select the optimal action under the current state according to the policy (i.e. mapping of state to action) that can maximise the long-term reward (i.e. value function). RL can be classified into three categories: critic-only, actor-only and actor-critic. The critic-only, or value-based, approach, updates the value functions according to feedback from the external environment. This approach has no explicit function of policy and hence an optimisation procedure needs to be involved. The actor-only approach is policy-based, with the policy continually being updated according to the policy gradient. The actor-critic approach combines the previous two approaches such that both the value function and policy are learnt and updated. Most real time signal control systems use the critic-only and actor-critic approaches, while the actor-only method is seldom used due to its low convergence speed.

Wiering (2000) and Wiering et al. (2004) proposed a multi-agent critic-only reinforcement learning approach to realise real time signal control that can minimise waiting time. Two types of agents were considered: vehicles in the system and traffic signals in the network. State variables were provided by the vehicles, including their location, final destination and the traffic signal they were moving to or waiting at. Since it was impossible to control all the traffic lights in a centralised way due to the large state space and control space, the signal control system was a distributed one with binary signal control variables (green or red). The objective was to minimise the total waiting time of all the vehicles around the junction. The look-up table of transition probability was updated continuously according to the actual counts from the traffic network.

Khamis & Gomaa (2014) extended Wiering's work by simulating the vehicle location in continuous-time and continuous-space, and by taking acceleration and deceleration into account. Their study adopted a Bayesian probability interpretation for the estimation of

transition probability. This research was also designed for distributed control. As noted by the authors, however, since the method depended on junction controller being able to access vehicle location and destination, it would be necessary to have sensors in vehicles that could achieve vehicle-to-infrastructure communication, which is impossible at this stage.

The research above adopts a discrete state space, yet, for the traffic problem, traffic states and control decisions are usually continuous, which makes the representation of the value function challenging. Hence, a new RL approach has been developed that uses approximation functions to represent the state-and-action value function. In this approach, the approximation function can be parametric, such as in a linear function (Cai, Wong & Heydecker, 2009; Yin, Dridi & EI Moundni, 2015) and tile coding (Abdoos, Mozayani & Bazzan, 2014), or nonparametric, such as with Neural Network (Li, Lv & Wang, 2016; Shabestary & Abdulhai, 2018).

Since the critic-only approach needs an optimisation procedure, such as the enumeration method, computation becomes demanding if the action space is large or continuous. The actor-critic method has been proposed as a way of solving this problem. The actor-critic approach has two parts: actor and critic. The former is responsible for selecting the action based on the current state according to a specific policy. The policy learns from the action-value provided by the critic part and is updated to ensure the action selected can bring a higher reward. The critic part calculates the action-value, which is used to evaluate the performance of the action. This evaluation feedback guides the next update of the actor part, such that it can be revised using the gradient that leads to lower variance.

Chu et al. (2020) used the actor-critic approach A2C to realise real time signal control. The traffic signals were operated in a distributed manner due to the large state space and action space of the centralised control. The communication between the agents was achieved by two approaches. The first approach was to design the local policy accounting for the neighbourhood's policy. The second approach was to calculate the local Q-function by adding the weighted sum of the neighbourhoods' Q-functions. Both the actor and critic parts were trained using the deep neural network with long-short term memory (Hochreiter &

Schmiduhber, 1997).

Aslani, Mesgari & Wiering (2017) reviewed the discrete state actor-critic and continuous state actor-critic methods. The authors highlighted that the discretisation of the continuous state variables and action variable reduced the ability to approximate an accurate value function. They therefore proposed that the value function can be approximated by tile coding and radial basis functions (RBF) leading to a distributed signal control system which they tested in Tehran's downtown area under different types of disruption.

The action space increases exponentially with the number of intersections controlled, therefore each RL algorithm only manages a single intersection (Wiering, 2000) or a small size network (Prashanth & Bhatnagar, 2011). The application of RL-based real time signal control can be achieved by using a hierarchical structure (Abdoos, Mozayani & Bazzan, 2013) or a distributed structure with information sharing (Salkham et al., 2008). The summary of coordination methods of RL-based real time signal control refers to El-Tantawy & Abdulhai (2010).

Besides learning the mapping between the traffic state and signal control via the feedback from the traffic network, some research learns the state-control relationship off-line. Liu et al. (2015) developed a two-stage real time signal control system using a linear decision rule (LDR) which determined the signal plans of the entire traffic network in a centralised manner. In LDR, the optimal solution was expressed as a linear function of traffic state. In order to account for the influence of uncertainty in traffic flow, the parameters of LDR were determined by distributionally robust optimisation (DRO). For a small network, LDR can be easily solved by MILP, while for a large network, the parameters can be determined by PSO with a link-based macroscopic simulation model for fitness estimation for a large network.

Song et al. (2019) extended the work of Liu et al. (2015) by formulating the relationship between the traffic state and signal control with a nonlinear function, known as the nonlinear decision rule (NDR). In their paper, the feedforward neural network and recurrent neural network were selected. Off-line training was achieved by PSO, with the fitness function being

based on a weighted objective accounting for delay reduction, CO_2 and black carbon emissions reduction, as estimated by a microscopic simulation model (i.e. S-Paramics). They showed that their system was able simultaneously to determine the signal plans of all the intersections in the network in a centralised manner.

2.2 Surrogate modelling

This section reviews existing real time signal controls, identifying surrogate modelling as a possible solution to the real time signal control and optimisation problem. Surrogate models can approximate a complicated input-output relationship solely based on observations from models or physical processes. An increasing amount of research is being dedicated to the surrogate modelling technique and its applications in a variety of domains. This section reviews the state-of-the-art of surrogate models to explore their underlying operational mechanism. This review of the applications of surrogate models is helpful to identify their potential in real time signal control optimisation.

2.2.1 Surrogate modelling techniques and classifications

Surrogate models can be broadly classified into physical and functional models (Conn, Scheinberg & Vincete, 2009; Osorio & Bierlaire, 2013). Since surrogate models are applied in a wide variety of different fields, however, the definitions of 'physical' and 'functional' models are equivocal. Søndergarrd (2003) reviewed existing terminologies for surrogate modelling to give clear definitions, and these are adopted in this thesis.

Physical models are defined as '*models based on knowledge about the particular physical system in question*' (Søndergarrd, 2003). Physical surrogate models are based on the knowledge of the underlying problem. They simplify the physical process or the model in a physical or numerical way. As a prerequisite, physical surrogate models need a structurally tractable model, referred to as the 'cheap' model due to its time efficiency. Although the cheap model is in theory a physical surrogate model, it cannot guarantee an accurate interpretation of the underlying physics of the system due to its low fidelity. A number of approaches have

therefore been developed to form a more accurate approximation based on the cheap model by means such as Response Correction, Multipoint Method and Space Mapping (Søndergarrd, 2003). As stated in the definition, each physical surrogate model is constructed for a particular system. This means that when a proposed physical surrogate model is applied to a new system it needs to be reconstructed, even to the extent that the cheap model needs to be replaced in some cases. Moreover, Zheng et al. (2019) pointed out that the complexity of the traffic system makes it difficult to obtain an analytical and differentiable cheap traffic model for physical surrogate modelling.

Compared with physical surrogate models, functional surrogate models are defined as '*models constructed without any particular knowledge of the physical system or governing equations*' (Søndergarrd, 2003). Functional surrogate models are constructed based on the empirical data from the physical system or governing equations so as to form a response surface that can approximate the input-output relationship. Functional surrogate models have a mathematical structure, where the parameters represent the features of the underlying system. This means that functional surrogate models are transferable to different problems and systems (Conn, Scheinberg & Vincete, 2009). Since the functional model only embodies the underlying physics of the system at the sample points, however, it is necessary to distribute multiple sample points in the entire design space to form an accurate response surface, with the number of sample points depending on the dimension and complexity of the problem. Figure 2.5 shows a classification of some popular surrogate modelling techniques.

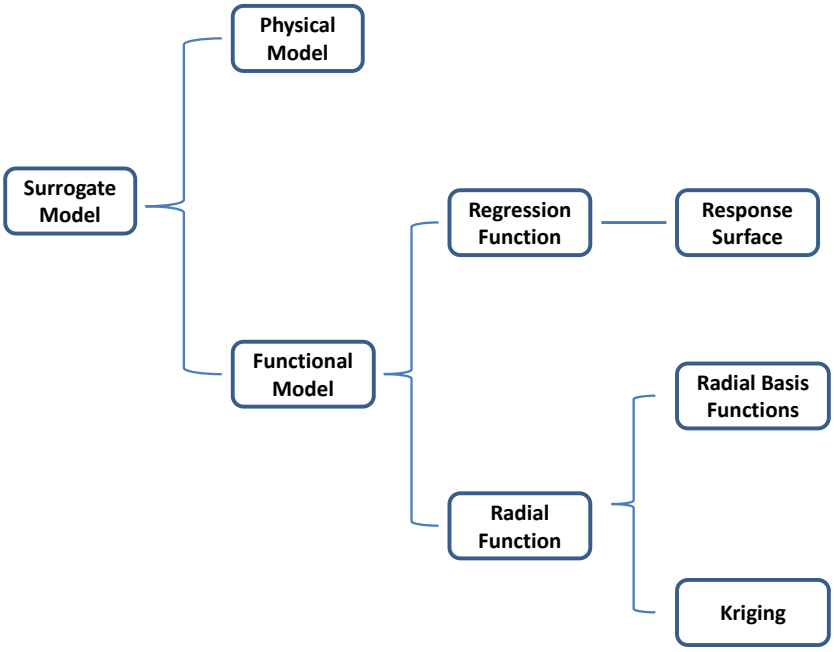


Figure 2.5 Classification of surrogate models

There are a variety of functional surrogate models, which are constructed based on algebraic expressions called basis functions. The two most common basis functions are regression functions and radial functions. The regression model establishes a basis function for the whole design space, which is fitted to the sample data. The radial function approach, meanwhile, constructs a basis function around each sample point and combines them together based on the assumption that adjacent points have high similarity.

This thesis addresses the functional surrogate class of models since the aim is to propose a model-free real time signal control, which conflicts with the physical model-based nature of the physical surrogate model. The following sections therefore introduce the most widely-used functional surrogate models: response surface model, radial basis function and Kriging.

General Notations

For surrogate modelling, given a set of training points, $\boldsymbol{X} = \left[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}\right]^T$, and the corresponding observations, $\boldsymbol{y} = \left[y^{(1)}, y^{(2)}, \dots, y^{(N)}\right]^T$, the estimated output of any k-dimensional testing point, $\boldsymbol{x} = [x_1, x_2, \dots, x_k]^T$ from a surrogate model can be represented

as $\hat{f}(\boldsymbol{x})$.

To show the formulation of surrogate modelling, a notation list is given:

General	
$\boldsymbol{X} = \left[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(N)}\right]^T$	An initial set of training points
$\boldsymbol{y} = \left[y^{(1)}, y^{(2)}, \dots, y^{(N)}\right]^T$	Corresponding observations of the initial set of training points
$\boldsymbol{x} = [x_1, x_2, \dots, x_k]^T$	A *k*-dimensional testing point
$\hat{f}(\boldsymbol{x})$	Estimated output at $\boldsymbol{x}$
$\hat{s}^2(\boldsymbol{x})$	Mean Squared Error (MSE) at $\boldsymbol{x}$
$\hat{s}(\boldsymbol{x})$	Standard error at $\boldsymbol{x}$
Response Surface Model	
β_0	Interception coefficient
β_i	Linear term coefficient of the *i*-th dimension
β_{ii}	Quadratic term coefficient of the *i*-th dimension
β_{ij}	Interaction term coefficient between the *i*-the and *j*-th dimensions
Radial Basis Function	
$\psi(\cdot)$	Radial basis function

$\boldsymbol{w} = [w_1, w_2, \dots, w_N]^T$	The vector of weighting parameter
$\boldsymbol{\Psi}$	Gram matrix, where $\boldsymbol{\Psi}_{i,j}$=$\psi(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$
$\boldsymbol{\psi}$	Matrix of radial basis function between testing point and training points, where $\boldsymbol{\psi}_i$=$\psi(\boldsymbol{x}^{(i)}, \boldsymbol{x})$
Kriging	
$f(\boldsymbol{x})$	Output at $\boldsymbol{x}$
$\boldsymbol{g}(\boldsymbol{x})$ $= [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \dots, g_m(\boldsymbol{x})]^T$	A set of regression functions
$\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_m]^T$	Weighting parameter
$Z(\boldsymbol{x})$	Gaussian random function at $\boldsymbol{x}$
σ^2	Variance
$r(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$	Correlation function between $\boldsymbol{x}^{(i)}$ *and* $\boldsymbol{x}^{(j)}$
$\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_k]^T$	Parameters in the exponential correlation function
$\boldsymbol{p} = [p_1, p_2, \dots, p_k]^T$	
μ	Mean of the objective function
$\boldsymbol{R}$	Correlation matrix, where $\boldsymbol{R}_{i,j}$=$r(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$
$\boldsymbol{r}$	Correlation matrix, where $\boldsymbol{r}_i$=$r(\boldsymbol{x}^{(i)}, \boldsymbol{x})$
Infill Strategy	

$\boldsymbol{x}_{infill}$	Infill point
$\hat{e}(\boldsymbol{x})$	Estimation error at $\boldsymbol{x}$
$LB(\boldsymbol{x})$	Statistical lower bound at $\boldsymbol{x}$
α	Weighting parameter in the statistical lower bound
$Y(\boldsymbol{x})$	One of the random values at **x**, when $\hat{f}(\boldsymbol{x})$ is considered as a realisation of a random variable
$PI(\boldsymbol{x})$	Probability of Improvement at $\boldsymbol{x}$
y_{min}	Current minimum observation among all the training points
$\Phi(\cdot)$	Cumulative distribution function.
$EI(\boldsymbol{x})$	Expected Improvement at $\boldsymbol{x}$
$\phi(\cdot)$	Probability density function
$WEI(\boldsymbol{x})$	Weighted Expected Improvement at $\boldsymbol{x}$
ω	Weighting factor in Weighted Expected Improvement

Response Surface Model

The Response Surface Model (RSM) was first proposed by Box & Wilson (1951). For many years, it was the most widely used surrogate model, being replaced only recently by other methods such as the radial basis function (Simpson et al., 2008). With the lower order

polynomial model as an example, the following equations indicate how to use the first-order polynomial model and the quadratic polynomial model to make the approximation. In Equations (2.1) and (2.2), β_0 is the interception coefficient, β_i is the linear term coefficient, β_{ii} is the quadratic term coefficient and β_{ij} is the interaction term coefficient. x_i is the *i*-th dimension of input variable *x*, and the value of each coefficient also indicates the contribution of each term to the overall observation.

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i \tag{2.1}$$

$$\hat{f}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{j=1}^{k} \sum_{\substack{i=1 \\ i \neq j}}^{k} \beta_{ij} x_i x_j \tag{2.2}$$

All the parameters $\boldsymbol{\beta}$ can be estimated by the least squares method. The parameters that can minimise the sum of the squared errors between estimated performance and actual observations of the sample data are chosen. Hence, the parameters $\boldsymbol{\beta}$ can be derived from Equation (2.3).

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(\boldsymbol{y}^{(i)} - \hat{f}\left(\boldsymbol{x}^{(i)}\right) \right)^2 \tag{2.3}$$

Both the first-order polynomial model and the quadratic polynomial model are relatively simple in structure and thus cannot fully reflect the complicated input-output relationship, making them unsuitable for highly nonlinear problems. Although the flexibility of polynomial models can be enhanced by adding additional high order terms, extending low-order polynomials to high-order polynomials is not always practical due to the limitation of the computation budget. Moreover, while it is true that a more sophisticated polynomial model can fit the sample data more accurately, this may lead to a deterioration in the estimation

ability in respect to an unknown input due to the overfitting of noise. This means that, although very challenging, it is crucial to choose an appropriate RSM. The selection of an RSM depends on the expected behaviour of the underlying problem, which is usually unknown. RSM is always combined with the optimisation method called Response Surface Method (Conn et al., 2000).

Radial Basis Function

The radial basis function (RBF) model is composed of a set of radial basis functions that are only related to the distance between points. The optimisation problem related to RBF is always solved by a derivative-free optimisation technique, since RBF does not have a closed-form solution (Wild & Shoemaker, 2011; Le Thi, Vaz & Vincinte, 2012). Compared with the lower order polynomial method, the RBF is more suitable for dealing with high-dimensional and complex problems.

The estimation in the RBF can be expressed as the weighted sum of radial basis functions, the values of which depend only on the radial distance of $\boldsymbol{x}$, the point to estimate, and the centre of each basis function. In RBF, the centre of the basis function coincides with the training point, $\boldsymbol{x}^{(i)}$, which is the *i*-th sample point as well as the centre of the *i*-th basis function. The estimation of any untried point can be rewritten using Equation (2.4), where $\boldsymbol{x}$ is the test point, and w_i is the weighting parameter of the *i*-th radial basis function $\psi(\cdot)$.

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \, \psi\left(\left\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\right\|\right) \tag{2.4}$$

There are many forms of basis function, including simple fixed bases and more complex parametric basis functions that have additional parameters. The most common basis function is the Gaussian function given in Equation (2.5), where σ is the width parameter of the Gaussian function that needs to be pre-defined:

$$\psi\left(\left\|\boldsymbol{x}-\boldsymbol{x}^{(i)}\right\|\right)=exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}^{(i)}\right\|^2}{2\sigma^2}\right) \tag{2.5}$$

The radial basis function is an interpolation model that can reproduce the observations of training points. The weighting parameter can therefore be estimated by inputting the training points into the function, such that $\hat{f}\left(\boldsymbol{x}^{(i)}\right)=y^{(i)}$. The derivation process is shown in Equations (2.6) and (2.7), where $\boldsymbol{w}=[w_1,w_2,\ldots,w_N]^T$ is the set of weighting parameters, $\boldsymbol{\Psi}$ is the Gram matrix, which calculates the basis function values between each pair of training points using Equation (2.8).

$$\boldsymbol{\Psi w}=\boldsymbol{y} \tag{2.6}$$

$$\boldsymbol{w}=\boldsymbol{\Psi}^{-1}\boldsymbol{y} \tag{2.7}$$

$$\boldsymbol{\Psi}=\begin{bmatrix}\psi\left(\left\|\boldsymbol{x}^{(1)},\boldsymbol{x}^{(1)}\right\|\right) & \psi\left(\left\|\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)}\right\|\right) & \cdots & \psi\left(\left\|\boldsymbol{x}^{(1)},\boldsymbol{x}^{(N)}\right\|\right)\\ \psi\left(\left\|\boldsymbol{x}^{(2)},\boldsymbol{x}^{(1)}\right\|\right) & \psi\left(\left\|\boldsymbol{x}^{(2)},\boldsymbol{x}^{(2)}\right\|\right) & \cdots & \psi\left(\left\|\boldsymbol{x}^{(2)},\boldsymbol{x}^{(N)}\right\|\right)\\ \vdots & \vdots & \ddots & \vdots\\ \psi\left(\left\|\boldsymbol{x}^{(N)},\boldsymbol{x}^{(1)}\right\|\right) & \psi\left(\left\|\boldsymbol{x}^{(N)},\boldsymbol{x}^{(2)}\right\|\right) & \cdots & \psi\left(\left\|\boldsymbol{x}^{(N)},\boldsymbol{x}^{(N)}\right\|\right)\end{bmatrix} \tag{2.8}$$

Once the parameters are determined, the approximation of any untried point can be reformulated as Equation (2.9), while $\boldsymbol{\psi}$ in Equation (2.10) calculates the basis function values between the test point and each training point. The closer the distance is, the greater the Gaussian basis function value.

$$\hat{f}(\boldsymbol{x})=(\boldsymbol{\Psi}^{-1}\boldsymbol{y})^T\,\boldsymbol{\psi} \tag{2.9}$$

$$\boldsymbol{\psi} = \begin{bmatrix} \psi\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}\right) \\ \psi\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}\right) \\ \vdots \\ \psi\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}\right) \end{bmatrix} \tag{2.10}$$

With RBF, the mean squared error (MSE) of the test point can be calculated by Equation (2.11):

$$\hat{s}^2(\boldsymbol{x}) = \mathbf{1} - \boldsymbol{\psi}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\psi} \tag{2.11}$$

Kriging

Kriging is a popular surrogate model proposed by Krige (1951). It approximates the unknown function by minimising the MSE, and is able to be adapted to highly nonlinear and high-dimensional functions. The following section will introduce the Kriging model construction based on the method proposed by Sacks et al. (1989).

The output response is a realisation of a random function (or a stochastic process) and can be represented by the combination of the global trend function $\boldsymbol{g}^T(\boldsymbol{x})\boldsymbol{\beta}$ and Gaussian random function $Z(\boldsymbol{x})$, as shown in Equation (2.12).

$$f(\boldsymbol{x}) = \boldsymbol{g}^T(\boldsymbol{x})\boldsymbol{\beta} + Z(\boldsymbol{x}) \tag{2.12}$$

where $\boldsymbol{g}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \dots, g_m(\boldsymbol{x})]^T$ is a set of known regression basis functions, and $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_m]^T$ is a set of corresponding weighting parameters. In practice, a constant trend function is sufficient for most problems hence, in what follows, a constant value, μ, is used instead of $\boldsymbol{g}^T(\boldsymbol{x})\boldsymbol{\beta}$. $Z(\boldsymbol{x})$ is the realisation of a stochastic process, the mean of which is equal to zero, while the variance equals σ^2 and covariance is non-zero. Equation (2.13) shows the covariance:

$$Cov\left[Z\left(\boldsymbol{x}^{(i)}\right), Z\left(\boldsymbol{x}^{(j)}\right)\right] = \sigma^2 r\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) \tag{2.13}$$

In Equation (2.14), $r\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right)$ is the correlation function between two sample points, $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$, which only depends on the Euclidean distance between them. Although there are many correlation functions, this thesis adopts the exponential correlation function.

$$r\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = exp\left[-\sum_{l=1}^{k} \boldsymbol{\theta}_l \left|\boldsymbol{x}_l^{(i)} - \boldsymbol{x}_l^{(j)}\right|^{p_l}\right] \quad 1 < p_l \leq 2 \tag{2.14}$$

In the correlation function given in Equation (2.14), $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_k]^T$ and $\boldsymbol{p} = [p_1, p_2, \ldots, p_k]^T$ are the parameters to be determined. This is similar to the Gaussian basis function of RBF (i.e. Equation (2.5)), except that Equation (2.14) replaces the constant weighting parameter with a set of parameters, such that each dimension has its own parameter. It can be seen from the equation that the correlation is highly sensitive to the distance between two points. When the distance between two points $\left|\boldsymbol{x}_l^{(i)} - \boldsymbol{x}_l^{(j)}\right|$ tends to zero, the correlation tends to one, which is the maximum, and when the distance tends to infinity, then the correlation tends to zero.

The two parameters are tuned to minimise MSE. Derived from Sacks et al. (1989), the maximum likelihood estimates (MLE) for μ and $\hat{\sigma}^2$ can be written as Equation (2.15) and (2.16), which are functions of $\boldsymbol{\theta}$ and $\boldsymbol{p}$. $\boldsymbol{R}$ is a $\mathbf{n} \times \mathbf{n}$ correlation matrix, as shown in Equation (2.17), and $\mathbf{1}$ is a $n \times 1$ matrix, where all values are 1.

$$\mu = \frac{\mathbf{1}^T \boldsymbol{\Psi}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{\Psi}^{-1} \mathbf{1}} \tag{2.15}$$

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\mu)^T \boldsymbol{\Psi}^{-1} (\boldsymbol{y} - 1\mu)}{n} \tag{2.16}$$

$$\boldsymbol{R} = \begin{bmatrix} r\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}\right) & r\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) & \cdots & r\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}\right) \\ r\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(1)}\right) & r\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(2)}\right) & \cdots & r\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}^{(N)}\right) \\ \vdots & \vdots & \ddots & \vdots \\ r\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}\right) & r\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(2)}\right) & \cdots & r\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}\right) \end{bmatrix} \tag{2.17}$$

The parameter $\boldsymbol{\theta}$ and $\boldsymbol{p}$ can be obtained by maximising the concentrated ln-likelihood function, which is expressed as:

$$ln(L) \approx -\frac{1}{2} ln(\hat{\sigma}^2) - \frac{1}{2} ln\ |\boldsymbol{R}| \tag{2.18}$$

Once the values of $\boldsymbol{\theta}$ and $\boldsymbol{p}$ are derived, the Kriging model can be constructed, and the value at any sample site can be estimated. The approximation of the output is of the form:

$$\hat{f}(\boldsymbol{x}) = \mu + \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{R}^{-1}(\boldsymbol{y} - \mu\boldsymbol{1}) \tag{2.19}$$

where $\boldsymbol{r}$ is the $n \times 1$ correlation matrix between the test point and all the sample points:

$$\boldsymbol{r} = \begin{bmatrix} r\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}\right) \\ r\left(\boldsymbol{x}^{(2)}, \boldsymbol{x}\right) \\ \vdots \\ r\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}\right) \end{bmatrix} \tag{2.20}$$

The Kriging model can also estimate the approximation error (i.e. MSE) with Equation (2.21).

$$\hat{s}^2(\boldsymbol{x}) = \sigma^2 \left[1 - \boldsymbol{r}^T\boldsymbol{R}^{-1}\boldsymbol{r} + \frac{1 - \boldsymbol{1}^T\boldsymbol{R}^{-1}\boldsymbol{r}}{\boldsymbol{1}^T\boldsymbol{R}^{-1}\boldsymbol{1}}\right] \tag{2.21}$$

The construction of the Kriging model and prediction can be realised with Design and Analysis of Computer Experiments (DACE) (Lophaven, Nielsen, & Søndergaard, 2002),

which is a Kriging model toolbox of MATLAB.

2.2.2 State-of-the-art in infill strategies

In addition to the initial training points, the accuracy of a surrogate model can be increased by adding more points to the training points set. An infill strategy is used to decide which point to select, where the points selected are called infill points.

Forrester & Keane (2009) mentioned that these infill strategies could be divided into three categories, as shown in Figure 2.6. The first one is the pure exploitation approach, which only selects the points that are close to the current optimal point. The second one is the pure exploration approach, which selects the points that can improve global accuracy most. These two approaches are driven by different aims. Pure exploitation focuses on the search for the optimal point, and theoretically, when it eventually converges, it can find the optimal point. This may not be the *'true'* optimal point, however, as the *'true'* optimum may lie in an area with few sample points (Jakobsson et al., 2010). Although pure exploration can enhance global accuracy by inserting more points in areas with greater uncertainty, if the same emphasis is placed on every corner of the design space, pure exploration takes a long time to meet the stopping criteria due to the waste of effort on regions far away from the optimal point. The third approach, therefore, combines the exploration approach and exploitation approach together in an effort to improve the global accuracy while ensuring that more attention is paid to the potential optimal region. The following part will discuss some infill strategies in detail. Here, for generality, the case of minimisation is considered.

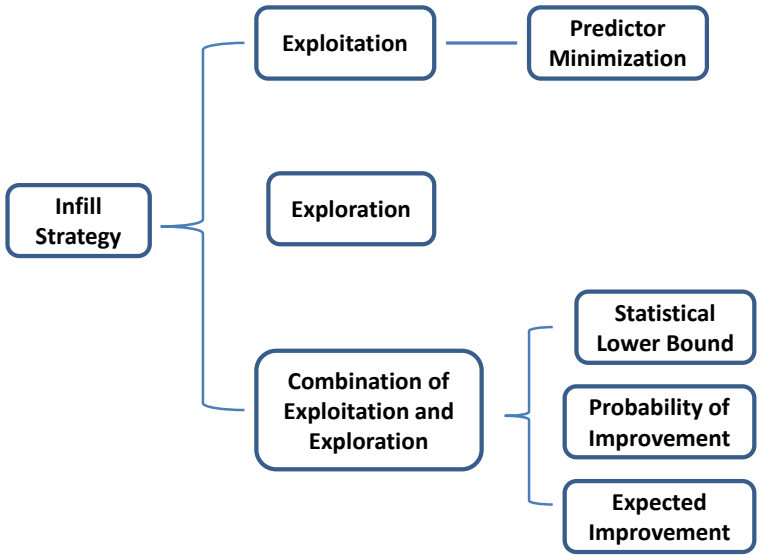


Figure 2.6 An overview of infill strategies

Exploration approach

The surrogate model aims to obtain a continuous function based on a limited number of training points. Since no background knowledge is required, the performance of the model depends entirely on the number and location of training points. The region with fewer training points may have lower approximation accuracy. The exploration approach selects infill points from the entire design space to ensure every corner of the design space can be approximated accurately. It can be seen as an attempt to fill the gaps between the initial training points. The easiest way is to find the points with the largest estimation error, as shown in Equation (2.22), where $\hat{e}(\boldsymbol{x})$ is the estimation error. For a Gaussian-based surrogate model, $\hat{e}(\boldsymbol{x})$ can be easily calculated with the equation provided. Forrester & Keane (2009) pointed out that pure exploration search can contribute to the construction of a globally accurate model that will be further used for real time optimisation.

$$\boldsymbol{x}_{infill} = argmax\ \hat{e}(\boldsymbol{x}) \quad x_{min} \leq \boldsymbol{x} \leq x_{max} \tag{2.22}$$

Exploitation approach

A predictor minimisation strategy is a local infill strategy that selects the optimal point to refine the surrogate model, as shown in Equation (2.23). The output $\hat{f}(\boldsymbol{x})$ is predicted by the

surrogate model.

$$\boldsymbol{x}_{infill} = argmin\, \hat{f}(\boldsymbol{x}) \quad x_{min} \leq \boldsymbol{x} \leq x_{max} \tag{2.23}$$

The addition of new infill points can improve the approximation accuracy of the promising region. Compared with the exploration approach, the exploitation approach can converge to the optimum more quickly but it can only find the *'true'* optimum when the surrogate model is accurate enough. If the surrogate model is not accurate enough, the estimation error may render the optimal solution predicted by the surrogate model suboptimal in the actual model. This method is suitable for surrogate-based optimisation when the surrogate model has already been calibrated.

Combination of exploration approach and exploitation approach

The preceding paragraphs have introduced the exploitation and exploration approaches. The exploitation approach is sensitive to the approximation accuracy of the surrogate model, and may fail to find the global optimum if the initial samples are improperly distributed, or the global optimum is located in an uncovered area. The exploration approach needs a long time to converge and hence wastes computational resources if the ultimate aim is to find the global optimum rather than construct a globally accurate surrogate model over the entire design space. It is therefore a good idea to combine these two approaches. The following part introduces three main methods accounting for both exploration and exploitation.

Statistical lower bound

Minimising the statistical lower bound is the simplest way to balance the exploitation (optimisation of the predictor) and exploration (seeking areas of maximum uncertainty). The aim is to find the point $\boldsymbol{x}$ with a minimum statistical lower bound $LB(\boldsymbol{x})$, which is the weighted sum of the predicted value $\hat{f}(\boldsymbol{x})$ and standard error $\hat{s}(\boldsymbol{x})$. The minimisation of the lower bound is shown in Equation (2.24), while the lower bound is expressed by Equation (2.25).

$$\boldsymbol{x}_{infill} = argmin\, LB(\boldsymbol{x}) \quad x_{min} \leq x \leq x_{max} \tag{2.24}$$

$$LB(x) = \hat{f}(\boldsymbol{x}) - \alpha\, \hat{s}(\boldsymbol{x}) \tag{2.25}$$

where α is the weighting parameter controlling the balance between exploration and exploitation. When α =0, $LB(\boldsymbol{x})$ only depends on $\hat{f}(\boldsymbol{x})$, which is a pure exploitation approach, and when $\alpha \to \infty$, $LB(\boldsymbol{x})$ is dominated by exploration and the problem becomes one of finding the $\boldsymbol{x}$ that yields the maximum $\hat{s}(\boldsymbol{x})$. It is hard to determine the value of α, however, and an improper estimation of α may lead to over-exploitation or over-exploration.

Probability of Improvement

Another method to select the infill points, considering both exploitation and exploration, is by calculating the probability of improvement (PI). Considering $\hat{f}(\boldsymbol{x})$ as a realisation of a random variable, where $Y(\boldsymbol{x})$ is one of the random values of that variable that follows the normal distribution $Y(\boldsymbol{x}) \sim N(\hat{f}(\boldsymbol{x}), \hat{s}^2(\boldsymbol{x}))$. The probability of improvement $\boldsymbol{x}$ is:

$$PI(\boldsymbol{x}) = \Phi\left(\frac{y_{min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) \tag{2.26}$$

where y_{min} is the current optimum among all the training points, and $\Phi(\cdot)$ is the cumulative distribution function. The probability improvement is equivalent to the probability that $Y(\boldsymbol{x})$ is smaller than the current minimum y_{min}, such that:

$$PI(\boldsymbol{x}) = Probability(Y(\boldsymbol{x}) \leq y_{min}) \tag{2.27}$$

The final solution selected is the one that has the maximum value of PI, as shown in Equation (2.28). PI is an infill strategy that combines exploration and exploitation since it tends to select the point with small $\hat{f}(\boldsymbol{x})$ and large $\hat{s}(\boldsymbol{x})$.

$$\boldsymbol{x}_{infill} = argmax\ PI(\boldsymbol{x}) \quad x_{min} \leq x \leq x_{max} \tag{2.28}$$

Expected Improvement

PI can only show the probability of improvement, but not how much it can improve. A new method, called expected improvement (EI), has been proposed to select new points which can lead to the maximum improvement to the model (Jones, Schonlau & Welch, 1998; Sansena, Papalambros & Goovaerts, 2000). The EI of point $\boldsymbol{x}$ can be expressed by Equation (2.29).

$$EI(\boldsymbol{x}) = \begin{cases} (y_{min} - \hat{f}(\boldsymbol{x}))\Phi\left(\frac{y_{min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + \hat{s}(x)\phi\left(\frac{y_{min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) & if\ \hat{s} > 0 \\ 0 & if\ \hat{s} = 0 \end{cases} \tag{2.29}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cumulative distribution function and probability density function, respectively. y_{min} is the minimum observation among all the training samples, and $\hat{s}(\boldsymbol{x})$ is the standard error. The first term of the equation represents the improvement over the current minimum, which is the difference between the current minimum value and the predicted value of $\boldsymbol{x}$ multiplied by PI. The second term is the product of standard error and the probability that $\hat{f}(\boldsymbol{x})$ is equal to f_{min}. When the estimation error is zero (i.e. $\boldsymbol{x}$ is the training point), EI is zero, thus avoiding the repeated selection of the same point. The expected improvement will be large where $\hat{f}(\boldsymbol{x})$ has a high possibility of being smaller than y_{min} and/or there is high uncertainty in the value of prediction (Sasena, Papalambros & Goovaerts, 2003).

The basic idea is to search for the point with maximum expected improvement, which is usually likely to be in under-sampled regions near to the global optimum.

$$\boldsymbol{x}_{infill} = argmax\ E[I(\boldsymbol{x})] \quad x_{min} \leq x \leq x_{max} \tag{2.30}$$

Sobester, Leary & Keane (2005) modified the original expected improvement method by

introducing a weighting factor to control the balance between exploitation (optimisation of the predictor) and exploration (seeking areas of maximum uncertainty) more precisely. The new method is called weighted expected improvement, and is expressed in Equation (2.31):

$$WEI(\boldsymbol{x}) = \begin{cases} \omega(y_{min} - \hat{f}(\boldsymbol{x}))\Phi\left(\frac{y_{min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) + (1-\omega)\hat{s}(\boldsymbol{x})\phi\left(\frac{y_{min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}\right) & if\ \hat{s} > 0 \\ 0 & if\ \hat{s} = 0 \end{cases} \quad (2.31)$$

where ω is the weighting factor and $\omega \in [0, 1]$. When ω=1, WEI will search the region around the current optimum for the point with the greatest expected improvement, and when ω=*0*, WEI will search the whole design space with the primary goal of uncertainty reduction.

2.3 Application of surrogate modelling

Currently, the importance of traffic network management has been the focus of increasing attention among transportation agencies, who hope to understand how to make planning or control decisions supported by quantitative evidence. Traffic models are therefore used to quantify and compare the performance of the network under various decisions. Typically, interactions within the traffic network are complex and difficult to interpret analytically. Thus, a simulation model is usually adopted to represent the network dynamics. Simulation models are usually time-consuming, with expensive-to-evaluate objective functions. In order to address this problem, an optimisation problem can be integrated with the surrogate model, so as to approximate the relationship between the input variables and the performance of the network in a time-efficient way. Currently, however, surrogate-based optimisation is limited to strategic design for a specific network.

In a traffic management context, surrogate models have most often been used to solve the problem of toll charge plans. Chen et al. (2014a) proposed a surrogate-based optimisation system for the design of a highway toll charges plan. The problem was described as a bi-level

problem that takes dynamic traffic assignment (DTA) under fixed demand into account and their paper compared the performance of several surrogate models. The results showed that a model with a simple structure (i.e. quadratic polynomial) was only suitable for simple problems. Moreover, the performance of the interpolation Kriging model and the regression Kriging model that accounted for noise were compared. This revealed that when the original model contained noise, the regression Kriging model outperformed the interpolation Kriging model. GA was adopted to identify the suitable toll plan leading to the minimum average travel time.

Chen, Zhu & Zhang (2015) extended the work of Chen et al. (2014a) to solve the multi-objective optimisation problem. They aimed to minimise the average travel time and maximise the total toll revenue by taking the ratio of the average travel time and toll revenue as the objective. The input-output relationship was approximated by Kriging, and the optimisation was solved by GA. It was proved that a toll plan generated by the surrogate-based optimisation can reduce the average travel and increase the revenue.

Chen et al. (2016) formulated the trade-off between multiple objectives using desirability functions. The Kriging model was used to replace the stochastic mesoscopic simulator under DTA. The parameter tuning of the surrogate model, and thus toll charge optimisation, were solved by a deterministic global search algorithm called Dividing Rectangles (DIRECT) (Jones et al., 1993). The test results on a large-scale network showed that surrogate-based optimisation can reduce the computation time dramatically compared to simulation-based optimisation. Furthermore, the optimal toll plan obtained from the surrogate-based optimisation performed better than the benchmark.

He et al. (2016) formulated the multi-objective toll charge problem with no weight or expectation given to any objective as a Pareto Set problem, and the problem with an expectation given to the objective as a constrained optimisation problem. Regressing Kriging was shown to be able to handle the noise in the stochastic simulation model, and modified expected improvement infill methods were developed for the Pareto and constrained optimisation problems.

Zhang et al. (2014) proposed a Bayesian Stochastic Kriging model that can deal with parameter uncertainty and heteroscedasticity simulation noise (i.e. different variance at different locations). The developed model was used to solve the optimisation problem of the joint application of the toll plan and variable message signs. The results of the experiment showed that when the actual dynamics were heteroscedastic, the Bayesian Stochastic Kriging model had a better performance than regressing Kriging, Kriging and quadratic polynomial function models.

Surrogate models can also be used in traffic signal optimisation problems, which have more variables. Osorio & Bierlaire (2009) and Osorio (2010) compared the microscopic and macroscopic simulation models in terms of realism and efficiency. The importance of the interaction between the upstream flow and downstream flow was underlined. In order to substitute the expensive-to-evaluate simulation model, they developed a physical surrogate model based on finite capacity queueing theory. This was the first analytical model to take spillback into account. In addition, their proposed physical surrogate model could identify the occurrence of congestion and its impact on performance. The fixed-time signal optimisation problem based on this physical surrogate model was solved by the trust region approach. The empirical experiment verified the efficiency of the surrogate-based optimisation and the performance of the proposed fixed-time signal control.

Osorio & Bierlaire (2013) extended their previous work by combining a quadratic polynomial function, a general purpose functional surrogate model, with the analytical queueing model, a problem-specific physical surrogate model, to handle the local approximation and global information jointly. Chong & Osorio (2017) extended the static analytical model in Osorio & Bierlaire (2013) to a transient analytical model to consider the temporal variation of congestion. The whole control period was divided into several time horizons, and the signal plan of each time horizon could be solved simultaneously.

Osorio & Selvam (2015, 2017) used a global model covering the whole area, and a local model covering only the city centre, to approximate traffic dynamics. Even though both models had the same resolution, the local model was less accurate, since a fixed boundary

condition was assumed, but more efficient, since a smaller area was simulated. The two models were not combined into one output estimation model but instead a surrogate model was used to estimate the error between the two models to provide guidance as to which model to invoke during the optimisation.

In addition to toll charge optimisation and fixed-time signal control, surrogate models can be used to solve some other optimisation problems. Li, Lin & Chen (2017) pointed out that surrogate models could be applied to all kinds of network design problems, regardless of whether the design variables are continuous, discrete or mixed. Chen et al. (2015) solved the average delay minimisation problem by expanding existing links and adding new ones. This network design problem was formulated as a bi-level Mixed Integer Network Design Problem, which is integrated with DTA, and solved by surrogate-based optimisation. In addition, it was recommended that multiple surrogate models can be combined to provide a robust approximation. Other optimisation problems can also be addressed through surrogate-based optimisation, including the selection and design of projects that can improve safety and travel time (Rodriguez-Roman, 2018), maximisation of social welfare (Chen et al., 2014b) and identification of key determinants for sustainable transportation planning (Sayyadi & Awasthi, 2016).

According to the literature reviewed above, it can be seen that surrogate models are flexible enough to solve a large variety of transportation optimisation problem. With surrogate models, some time-consuming problems such as bi-level problems and multi-objective problems can be solved easily. As a result, surrogate models have been widely used in optimisation problems with sophisticated dynamics and no closed-form objective functions.

2.4 Conclusions

This chapter first reviewed the numerous real time signal control systems that have been developed or that are under research. The review of the developed systems showed that such systems cannot currently optimise an explicit network-level objective. Rather, they either optimise a localised objective for each intersection (e.g. SCATS, SCOOT, OPAC, PRODYN

and RHODES) or an inexplicit objective (i.e. minimise and balance link relative occupancy) like TUC. In addition, challenges exist in balancing their performance and efficiency. Although SCATS and SCOOT can be applied to large-scale networks, they lack responsive behaviour due to their limited decision alternatives (Dion & Yager, 1996). In contrast, OPAC, PRODYN and RHODES can only operate in a distributed or hierarchical way, due to their exponential complexity. Moreover, these developed systems lack the flexibility to accommodate user-defined input variables and objectives.

Section 2.1.2 reviewed the state-of-the-art in the research of real time signal control, especially their optimisation methods and whether a traffic model is necessary during optimisation. The detailed information, including optimisation method, operation structure, traffic model and objective are summarised in Appendix A. The limitations of the current research are summarised below.

- **Model-based approaches**: The performance of model-based real time signal control approaches is highly dependent on the quality of the traffic model used. It is difficult, however, to calibrate a traffic model due to the complexity of the traffic system. Moreover, for the sake of computational efficiency, only a macroscopic simulation model or mathematical formulation can be used, which makes the calibration even more challenging.

- **Rule-based approaches**: The rule-based approaches require additional expert knowledge to identify activation rules in advance, which is difficult to obtain. Moreover, each set of rules only focuses on one intersection, which might neglect the interaction between adjacent intersections.

- **Learning-based approaches**: Learning-based approaches are model-free but, because of this, are not interpretable, especially when integrated with an approximation function. They cannot explain how and why a decision is made, or identify which feature plays a dominant role. Therefore, it is impossible to extend the system for further analysis.

The review of the limitations of existing real time signal control research and practice in this chapter reveals that there is currently no model-free real time signal control that can optimise a network-level objective in a centralised manner.

After examining surrogate modelling techniques in detail, the following properties were found that suggest that surrogate modelling represents an attractive approach to real time signal control:

- **Model-free**: Surrogate modelling can approximate input-output relationships based on empirical observations without any human expertise, meaning that the output of given inputs can be estimated through a surrogate model rather than an analytical or simulation model.

- **Flexibility**: Few assumptions and restrictions are imposed on the form of inputs and outputs (i.e. traffic states, control variables and objectives).

- **Efficiency**: An output can be estimated via a surrogate model efficiently on the basis of its mathematical closed-form function.

- **Interpretability**: In contrast to the lack of interpretability associated with learning-based approaches, surrogate modelling can provide an explanation of the logic involved through its parameters.

The application of surrogate modelling to solve problems such as the toll charge problem, demonstrates its capability to solve complex optimisation problems, such as fixed-time signal control, efficiently. These problems are all off-line strategic design problems, however, and therefore this thesis focuses on developing a surrogate-based *real time* signal control system.

The following chapter now reviews surrogate modelling techniques and their applications in order to strengthen the contention that surrogate modelling has potential in real time signal control.

Chapter 3

Surrogate modelling for real time signal control

This chapter reviews surrogate modelling techniques and their applications in order to strengthen the contention that surrogate modelling has potential in real time signal control. It develops a conceptual and general framework for the construction of a surrogate model to interpret the relationship between traffic state, signal control and objective. Section 3.1 identifies the process of surrogate model construction. The sampling of initial training points is illustrated in Section 3.2. Section 3.3 describes the construction and training of the surrogate model. Section 3.4 presents how to carry out the infill procedure to a surrogate model for real time signal control. In Section 3.5, a numerical analysis is conducted on a test network constructed on the basis of an actual test site in West Glasgow so as to evaluate the approximation accuracy and computational efficiency of the proposed surrogate model.

3.1 Introduction

This section offers a conceptual discussion regarding the construction of the surrogate model for real time optimisation. As mentioned in Chapter 2, existing surrogate-based traffic control and optimisation models are all concerned with strategic or off-line decisions (i.e. decisions that are unresponsive to either network changes or to traffic conditions observed in real time). Examples include network design (Chen et al., 2015), congestion pricing (Chen, Zhu & Zhang, 2015), and fixed-time signal controls (Osorio & Bielaire, 2013). Surrogate modelling has yet to be applied to real time traffic control and management, and the key challenge lies in an effective construction and approximation of a response surface that encompasses:

(a) real time traffic states;

(b) control variables or parameters; and

(c) network performance dependent on (a) and (b).

In contrast, conventional surrogate-based traffic optimisation with off-line decisions involves only (b) and (c). Inevitably, the inclusion of (a) results in a greater dimensionality of the design space, and thereby complicates the analytical interpretation of the relationship between (b) and (c). For example, the same set of traffic signal control parameters can lead to very different network performance under varying real time traffic states. This section therefore investigates response surface construction for real time signal control systems.

As an approximation technique that interprets and predicts the input-output relationship of a complex system, surrogate modelling consists of: (1) sampling of the training points; (2) surrogate model construction; and (3) model validation and infill. Figure 3.1 outlines this framework.

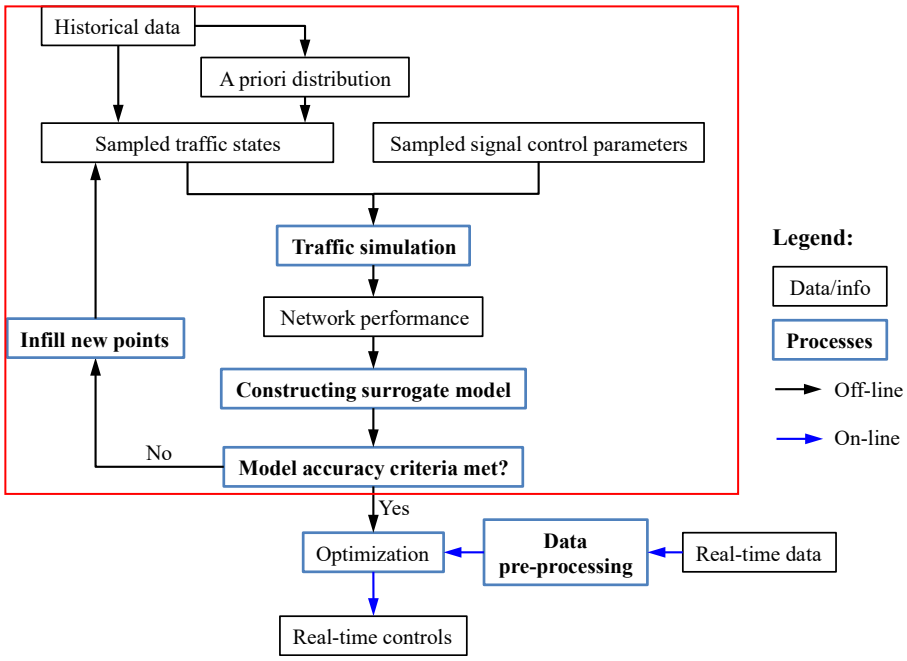


Figure 3.1 Framework of surrogate model construction for real time optimisation

In an off-line optimisation environment where the input space is known *a priori* (such as with deterministic demand or a known demand distribution), the problem of finding the optimal design feature or control parameters can be approached by analysis of a response surface, as illustrated in Figure 3.2 (left). This is approximated by training a surrogate model (such as

Kriging, radial basis functions, lower-order polynomials) with a set of sample points (shown as $p_1 \dots p_6$) obtained via a coordinated use of random initialisation and on-the-fly selection of infill points (Osorio & Bierlaire, 2013; Chen et al., 2014a). The problem of finding the optimal solution to the original problem (shown as the red curve) amounts to solving an optimisation problem on the surrogate model (shown as the blue curve). It is noted that the control-objective (u, z) relationship, described by either the red or blue curves, is defined for a fixed state, s_0, of the system. This state, s_0, can be one of a given network configuration, fixed demand, or known demand distribution. When the system state changes, so does the control-objective relationship; when such changes occur frequently or in real time, however, existing surrogate modelling approaches that only consider the control-objective relationship are unsuitable.

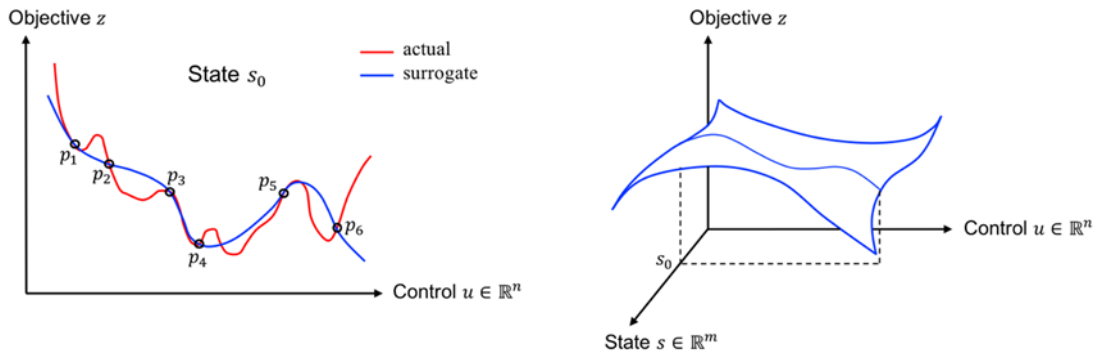


Figure 3.2 Illustration of off-line (left) and real time (right) optimisation with surrogate modelling

As shown in Figure 3.2 (right), the application of surrogate modelling to real time control requires the construction of a response surface that approximates the joint state-control-objective relationship, which enables the optimal control point to be found for any given state, s_0, by looking up the state-control-objective (s_0, u, z) relationship. The key challenges lie in determining how to:

a) define the state variables so as to capture the essential information needed by the controller without the kind of redundancy that would hinder its performance

b) sample the (s, u, z) relationship and construct the response surface in such a way as

to yield sound performance of real time optimisation

c) enable adaptive learning of the surrogate model so that its performance can be improved.

3.2 Generation and sampling of training points

To generate initial training points for the surrogate model, the overall strategy is to generate a battery of combinations of outputs with randomly-generated traffic states, s, and signal parameters, u. When there are sufficient historical data available, training points can be easily selected from that data. When the historical data are incomplete or insufficient for surrogate model construction, however, then synthetic data are generated. These synthetic data need to learn the feature distribution of the actual data via the use of indirectly-related data and information (Shi, Steenkiste & Veloso, 2018). The distribution of synthetic data needs to be highly similar to that of the actual data to ensure statistical equivalence between them (Templ, 2017). Once the feature distribution is obtained, synthetic data can then be generated from that. If the input variables are synthetic data rather than data selected from the historical database directly, the simulation model needs to be called to estimate the corresponding output performance.

3.2.1 Data processing

Since surrogate models are data-driven models their quality largely depends on the quality of the data used to construct the response surface, since it is this that is used to interpret the real dynamics of the network. To ensure a high level of accuracy of the surrogate model, if training data are selected from historical data, three types of data need to be removed or processed before use.

1. **Erroneous and missing data**

   The collected data may contain erroneous and missing data due to failures and faults of the detectors and transmission systems. Erroneous data can be identified and screened

via the following tests.

- **Threshold test**: Traffic data (e.g. volume, speed occupancy) have physical meaning; therefore they have their own upper and lower bounds. Hence, data that are beyond valid thresholds are regarded as erroneous data.

- **Basic Traffic Theory Test:** The traffic data should satisfy basic traffic theory, such as flow conservation and traffic flow theory between flow and density of the network. Data that violate these theories are regarded as erroneous data.

- **Temporal and Spatial Correlation Test**: Traffic flow is periodic over time, and therefore shows high similarity at the same period of the day and even day of the week. Moreover, traffic data at different locations of the traffic network, such as between downstream and upstream, have spatial correlation. Hence, data that do not follow the expected temporal and spatial correlation are thought to lack consistency and contain errors.

Since erroneous and incomplete data will lead to poor interpretation of traffic dynamics, they need to be screened out by these three tests, and the erroneous data removed. The removed data and missing data can, however, be imputed if necessary.

2. **Data collected under abnormal condition.**

The normal condition, also known as the typical condition, refers to the condition in which no special events have occurred that can affect the traffic pattern (Castro-Neto, 2009). Such special events can be classified as either expected events, e.g. road maintenance works, planned activities or unexpected events, e.g. incidents, accidents and extreme weather. These special events alter the network dynamics. Moreover, these special events usually occur infrequently. When background information is extracted from historical data, the data collected under abnormal conditions will be regarded as noise. It is therefore necessary to detect and delete the data collected under such atypical conditions. Only the recurrent fluctuations in the historical data that did not occur due to

special events are acceptable.

3. **Data affected by temporal variation of the traffic network**

   The traffic network changes dynamically with time and long-term variations of traffic demand, travel behaviour and network topology will lead to permanent changes in the network dynamics. Hence it is important to ensure that the traffic network has not experienced significant long-time variations during the data collection period. Otherwise, the data need to be removed or pre-processed before use, such as by scaling to a reference day.

Given the historical data, pre-processed as above, training points for the construction of the surrogate model can be selected. When synthetic data are used, however, an extra source of uncertainty is introduced by the so-called synthetic gap; that is, the difference in feature distribution between the synthetic data and the actual data (Zhang et al., 2018). This uncertainty can be reduced but never eliminated as it is caused by the process of generating synthetic data (Zhang et al., 2018). In this research, however, methods to evaluate the synthetic data are not discussed, as they are beyond the scope of this thesis. For details on such methods see Templ (2017).

3.2.2 Traffic state variables

In contrast to other engineering problems, traffic states are not uniformly distributed in the design space, and follow a certain feature distribution. It is not sensible, therefore, to select samples from the design space with equal probability. For traffic state variables, feature distribution is equivalent to its probability distribution. Sample points are selected from this distribution, and thus more points can be sampled around the region with a higher probability of occurrence. Compared with uniform sampling, this sampling approach can more truly reflect the traffic states of the actual traffic network.

In order to illustrate whether the selection of sample points following a certain distribution leads to a more accurate surrogate model than a uniform distribution, a small test was

conducted. Given that x is the input variable, y is output with the expression of $y = (6x-2)^2 \sin(12x-4)$. The input variable, x, has a probability distribution following a normal distribution, *X~N(0.5, 0.1)*. Two Kriging models were constructed, each with five training points. The training points were selected from two distributions: (1) normal distribution *X~N(0.5, 0.1)* and (2) uniform distribution *X~U[0, 1]*. Figure 3.3 and Figure 3.4 show the true function and these two constructed Kriging models. It can be seen that these two Kriging models have different shapes due to the different sampling distribution of training points. In the model in Figure 3.3, all five points were sampled near the central part, and it can be seen that Kriging model behaves nearly the same as the true model around the middle part, but does not perform well when x is between 0.8 and 1. In the model in Figure 3.4, meanwhile, training points were sampled from the uniform distribution, and hence the entire curve has a similar level of accuracy, except for the part when x is larger than 0.8, because of the sharp jump in the true function. These two figures show how the spatial arrangement of sample points affects the surrogate model construction intuitively.

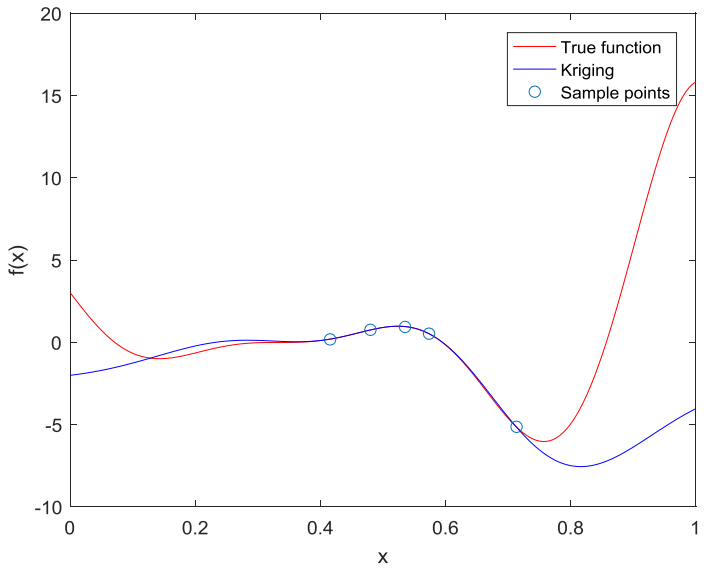


Figure 3.3 True function and Kriging model with sampling distribution 1 (normal distribution)

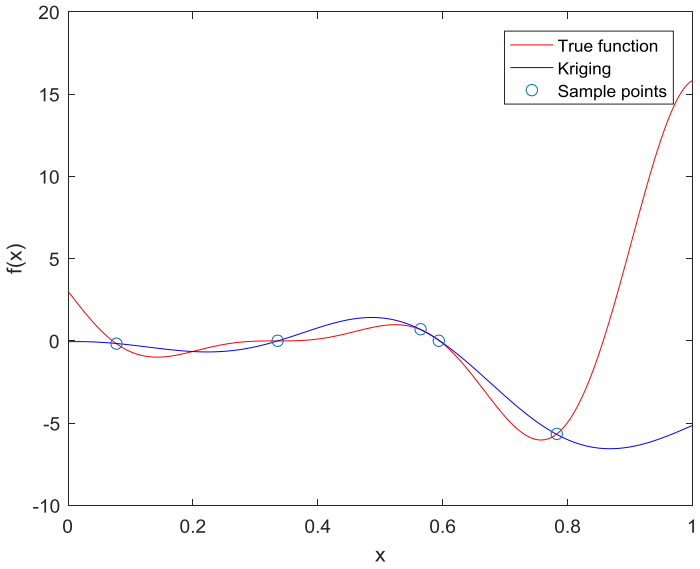


Figure 3.4 True Function and Kriging model with sampling distribution 2 (uniform distribution)

In order to evaluate the global accuracy of these two Kriging models, 1000 testing points were generated for validation. Assuming that the testing points also followed the normal distribution *N (0.5, 0.1)*, the histogram of all the testing points is shown in Figure 3.5. This mimics the fact that traffic states or conditions always follow a certain distribution.

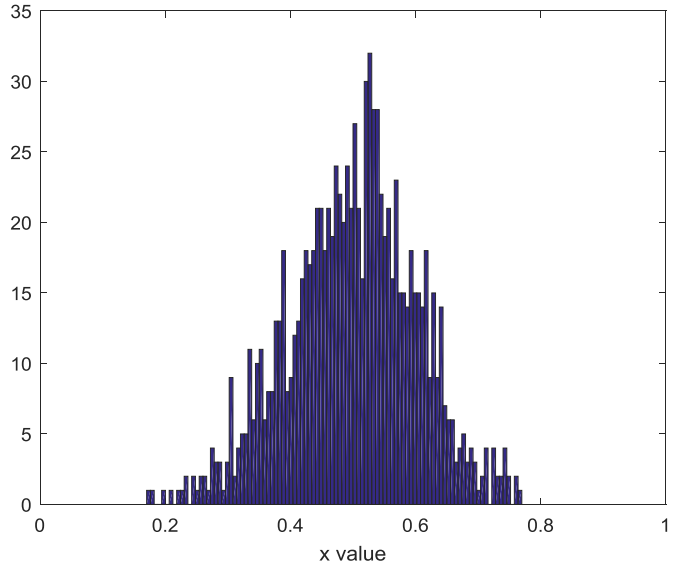


Figure 3.5 Histogram of 1000 testing points

As the sample points are randomly selected, in order to eliminate the influence of randomness, 100 Kriging models were constructed for each sampling distribution. This can reduce the uncertainty caused by the arrangements of sample points. Figure 3.6 shows the cumulative distribution function of the root mean squared error (RMSE) between the true function and the Kriging model. It can be seen that the one based on the normal distribution (top) has a much better distribution than the one based on the uniform distribution (bottom). In addition, the maximum RMSE of Kriging 1 (top) is less than 1.4, which is much lower than that of Kriging 2 (bottom). Table 3.1 displays the mean and variance of the RMSE. Kriging 1 has a much smaller mean and variance than Kriging 2. Both the figure and the table show that the Kriging model trained with points generated by the normal distribution has better performance and higher reliability. This experiment shows that if the distribution of the given traffic state is known, the training points should be sampled according to the same distribution since this can enhance the global accuracy and reduce the number of training points. In other words, where a limited budget is available, the primary target is to ensure the area with higher probability is accurate enough.

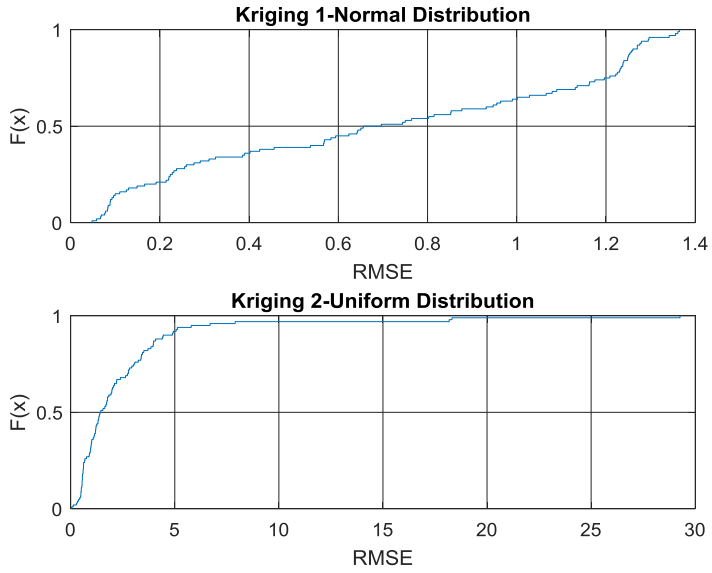


Figure 3.6 Empirical CDF plot of the RMSE of Kriging models generated with training points with (1) normal distribution and (2) uniform distribution

	Kriging 1-Normal Distribution	Kriging 2-Uniform Distribution
Mean	0.6992	2.5559
Variance	0.2145	14.9466

Table 3.1 Mean and variance of RMSE of two Kriging models trained by uniformly distributed samples and normally distributed samples

3.2.3 Signal control variables

Another component of input variables is the signal control variable, which will be optimised later. If the traffic system has already been controlled by real time signal controls, then various signal control variables are available. However, in most cases, the network is originally controlled by a non-adaptive signal control. In order to solve this problem, it is necessary to generate synthetic signal control variables to train the surrogate model. Signal control variables are sampled around a reasonable baseline signal plan. Compared with the uniform sampling in the entire traffic control space, this method can avoid the sampling of unreasonable control decisions to the specific problem and thus reduce the number of samples.

Cai et al. (2017) mentioned that the search process could be sped up if the candidate points were to be generated using a normal distribution design around the current optimum, so as to increase the probability of finding an accurate optimum point. Even though the final target of this work is not finding the global optimum, it can be used as a reference. Thus, if the current signal control or fixed-time signal control is available, this can be used as a basis for the generation of the training points. The normal distribution sampling method can be applied around the benchmark, which avoids wasting effort on sampling unreasonable signal controls.

The random sampling of signal control variables from a normal distribution cannot guarantee that all the samples are within the feasible region, however. Hence samples out of the feasible region need to be projected back to the feasible region. The projection of signal variables is introduced in detail in Section 3.4.2.2.

3.2.4 Network performance

When insufficient historical data with different combinations of traffic states and control variables are available, synthetic input data are generated, and a simulation model is used to estimate corresponding outputs.

As a supplement to the historical data, the simulation model should be able to reflect the actual dynamics of the network. Due to the simplifications and assumptions made in the simulation model, however, it can never replicate the actual network completely. Since the final target is to manage the traffic of the real network, the uncertainty representing the difference between the simulation model and the real network needs to be considered. This uncertainty arises from two main sources:

- **Structural uncertainty**: uncertainty due to the simplification of the model. The actual traffic network has complex interactions. For the sake of simplicity, however, simulation models can only extract the main behaviour rules and characteristics from the real network. The neglect of some less important behaviours and characteristics leads to structural uncertainty. As a result, simulation models can never be exactly the same as the real network.

- **Parameter uncertainty**: uncertainty related to the model parameters. The simulation model contains some parameters that are either calibrated with the data or customised by users. Parameter uncertainty includes the misestimation of these parameters, resulting in the simulation model failing to reflect the behaviour of the real network precisely. In addition, the traffic network is stochastic, and thus some of the parameters possess stochastic characteristics, which need to be expressed by probability distributions. The estimation of these probability distributions is challenging.

These uncertainties related to the simulation model will propagate through the following processes and affect the final performance. The inherent structural uncertainty cannot be eliminated and can only be reduced by adopting a new simulation model with higher

resolution. Parameter uncertainty, meanwhile, can be reduced by model calibration.

It should be noted that the selection of the simulation model is flexible. The framework itself does not have any constraint on it. Regardless of whether the model is macroscopic or microscopic, deterministic or stochastic, it can always be embedded in the framework as long as it can reflect the traffic dynamics accurately. When choosing a simulation model, therefore, the only thing to consider is the trade-off between realism and efficiency.

3.3 Model training and infill

3.3.1 Initial model training

The initial model training includes the selection and construction of the surrogate model. There is no surrogate modelling technique that always outperforms another, however: the choice depends on the nature of the problem to be solved, such as the number of dimensions and level of linearity. The following part introduces the construction of the response surface for real time signal control and optimisation.

Based on the initial samples selected, the response surface shown in Figure 3.2 (right) can be expressed as a mapping:

$$\boldsymbol{\Psi}\colon\ x = (s,u) \mapsto z, \qquad s \in \Omega \subset \mathbb{R}^m, \qquad u \in \Lambda \subset \mathbb{R}^n \tag{3.1}$$

where the 2-tuple (s,u) is the input and z denotes the output (objective value), which is obtained from off-line evaluation of the traffic system, either through analytical computation or simulation approaches. The vector, u, denotes the signal control parameters; while Λ is the feasible region for signal control parameters. The state variable, s, can be defined based on (partial) observation of the entire system at any given point in time, where Ω is the feasible region constrained by traffic theory. In the case of traffic control problems, s can be the inflow, queue length, or occupancy of certain links at the present time, t. In addition, s

is allowed to include the same quantities in the near past, $t-1, t-2, \ldots t-k$, to account for the latency effect inherent in a controlled traffic network; for example, a surge in traffic flow somewhere in the network may take several minutes to reach and interact with a signal controller, and hence the latter's decision should be informed by a state variable that contained such information a few moments earlier.

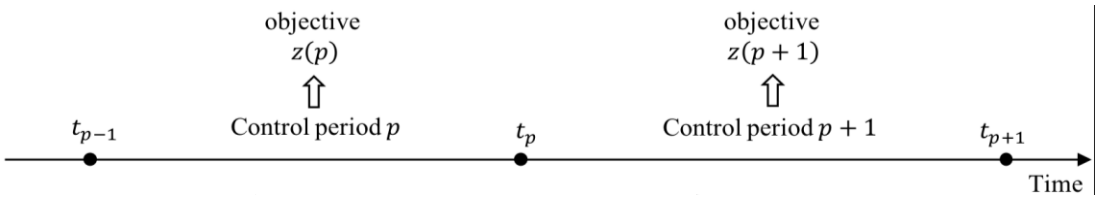


Figure 3.7 Real time signal control approach based on surrogate modelling

The proposed real time signal optimisation approach is illustrated in Figure 3.7. The whole analysis period is divided into several control periods with the same length of time (e.g. 5 min), and the points on the time axis represent the end of each control period. The signal control parameters remain fixed within each control period.

Traffic states are collected at the end of each control period, so traffic state variable $s(p)$ denotes the traffic state at the end of p-th control period t_p. In the surrogate-based approximation, it is not necessary to assume that the traffic state of the entire control period is known. The estimation of the objective within control period t_p only needs knowledge of the traffic state at the end of the previous control period $s(p-1)$, and the signal parameters $u(p)$ of control period t_p. Hence, the implicit and complex relationship can be approximated by the surrogate model as:

$$\widehat{\boldsymbol{\Psi}}\colon\ (s(p-1), u(p)) \mapsto z(p) \tag{3.2}$$

Remark 1. From Equation (3.2), the state, control and objective, whose relationship we are aiming to approximate with surrogate models, are not from the same control period. Indeed, the control decision to be made at the end of each control period must be based on past

observations unless some traffic prediction capabilities are involved, which is not considered in this thesis as they may bring large uncertainties in themselves. Moreover, the control decision, $u^*(p)$, can only take effect in the next control period, p. It is important to capture the relationship $(s(p-1), u(p)) \mapsto z(p)$ using surrogate models to guarantee the performance of the on-line signal control.

3.3.2 Infill strategy

The surrogate model needs to be proved to be accurate enough before it can be used for real time optimisation. In practice, it is difficult to use surrogate models with few sample points to describe the relationship between inputs and outputs due to the lack of knowledge of the objective functions. External validation is therefore carried out to provide a quantitative measure of global accuracy. Equation (3.3) and Equation (3.4) show how to quantify the error between true observations and estimated outputs of the surrogate model. RMSE and Mean Absolute Percentage Error (MAPE) are both global error measures that therefore offer insight into the model accuracy of the entire design space.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_{Test}} \left(\boldsymbol{\Psi}(x_i) - \widehat{\boldsymbol{\Psi}}(x_i)\right)^2}{N_{Test}}} \tag{3.3}$$

$$MAPE = \frac{1}{N_{Test}} \sum_{i=1}^{N_{Test}} \frac{\left|\boldsymbol{\Psi}(x_i) - \widehat{\boldsymbol{\Psi}}(x_i)\right|}{\boldsymbol{\Psi}(x_i)} \times 100\% \tag{3.4}$$

Where N_{Test} is the total number of testing points, $\boldsymbol{\Psi}(x_i)$ and $\widehat{\boldsymbol{\Psi}}(x_i)$ are the true observation and the estimated output from the surrogate model of the i-th testing point, respectively.

The global accuracy of the surrogate model can be improved if a larger number of points are used to construct the model, but this will increase the computational time needed for model

construction and output estimation. It is therefore worth exploring how to locate the training points. An infill strategy is a procedure to add more points to the surrogate model to enhance its accuracy. Several infill strategies have been discussed in Section 2.2.2. There, it was shown that, compared with the exploitation-based infill approach, aiming to find the optimal point, the exploration approach, which improves the global accuracy over the entire design space, is more suitable when real time optimisation is the final goal.

Generally, the infill strategy searches the entire search space, selecting one infill point at a time. Each infill needs to solve an optimisation problem in continuous space. This conventional infill strategy involves lengthy searching time, especially when the search space is large. To address this, Regis & Shoemaker (2007) proposed an infill strategy called the Metric Stochastic Response Surface (MSRS) method. This selects the next point to evaluate from a number of randomly-generated candidate points according to the weighted score of response surface prediction and distance metric of all the candidate points. The results show that this candidate sampling approach is easier to implement compared with the typical infill criteria, such as EI and PI, that need to solve the optimisation of sub-problems with non-convex objective function over the entire design space. This candidate sampling approach is flexible as it allows various selection criteria to be embedded within it to guide the generation of infill points.

In the case of real time optimisation, error-based approaches are suitable criteria to guide the selection of infill points. The error measures of the points can be derived from equations like Absolute Error (AE) and Relative Error (RE), in Equations (3.5) and (3.6), respectively, or with the closed-form analytical expression of estimation error as in Equation (2.21) of the Kriging model.

$$AE = \left|\boldsymbol{\Psi}(x_i) - \widehat{\boldsymbol{\Psi}}(x_i)\right| \tag{3.5}$$

$$RE = \frac{\left|\boldsymbol{\Psi}(x_i) - \hat{\boldsymbol{\Psi}}(x_i)\right|}{\boldsymbol{\Psi}(x_i)} \times 100\% \tag{3.6}$$

Compared with the single point infill strategy, the multi-point infill strategy can reduce the number of infill iterations. It selects k infill points from K candidate points in each iteration. One of the drawbacks of a multi-point infill strategy is that some, or even all, of the k infill points cluster within a small region that has low estimation accuracy. To avoid the k infill points being too close to each other, and ensure sample diversity, it is necessary to define appropriate criteria to force the infill points to maintain the minimum distance between any two infill points. Ma et al. (2017) defined that if the Euclidian distance between two points $d = \sqrt{\sum_{i=1}^{k}\left(x_i^{(1)} - x_i^{(2)}\right)^2}$ is smaller than the specified critical distance, ε_{d}, then these two points are correlated and cannot be selected as infill points at the same time. Dong et al. (2018) defined the minimum allowable distance between two points, $Dis = w \times \|max(\boldsymbol{Range}) - min(\boldsymbol{Range})\|$, to ensure sample diversity, where $\boldsymbol{Range}$ is the vector representing the range of the design space. Both of these methods limit the distance between the points with a fixed critical value, but it is hard to determine that critical value without trial and error. In addition, depending on the regularity of the design space, some regions with higher nonlinearity may need smaller critical values. Furthermore, as more infill points are added, the average distance between the points becomes small due to increasing sample density. This makes the determination of the critical distance even more complicated.

In view of these issues, an infill strategy with adaptive distance thresholds is designed as follows. Firstly, sort all the candidate infill points in descending order by their mean square errors. Secondly, working from the top of the list, compute the minimum distance of the candidate infill point from the existing sample points, δ_E, and the minimum distance of the candidate infill point from the newly added infill points, δ_B. If $\delta_E \leq \delta_B$, the candidate is added as an infill point; otherwise move on to the next candidate in the list until all k infill points have been selected. This procedure is described in Algorithm 3.1.

Algorithm 3.1: Multi-point infill with adaptive distance threshold (infill k points)

Input	Surrogate model trained with all points in E.
Step 1	Generate the set of candidate infill points, C, with cardinality K.
Step 2	Calculate the mean square errors of all the candidate points, and rank them in descending order $c_1, c_2, \dots, c_K \in C$. Initialise infill point set $B = \{c_1\}$. $i = 1$.
Step 3	$i = i + 1$.
Step 4	Calculate the minimum distances $\delta_E \doteq min_{e \in E} \|c_i - e\|$, $\delta_B = min_{b \in B} \|c_i - b\|$. *If* $\delta_E \leq \delta_B$, $B = B \cup \{c_i\}$.
Step 5	If $\|B\| < k$, go to Step 3; otherwise, terminate the algorithm and output B

The k infill points generated by Algorithm 3.1 will be added to the set of training points, and then the surrogate model is re-trained. The involvement of additional sets of samples allows the surrogate model to be improved or tailored. After each re-training of the surrogate model, the global accuracy is evaluated with Equations (3.3) and (3.4). The whole infill and evaluation process is repeated until a certain criterion has been, met or the available computational budget has been used.

3.4 Simulation test

The following section introduces an empirical test conducted in a small test network. The purpose of this test is to investigate whether the surrogate model can accurately simulate the complex state-control-objective relationship of the real network. By applying the infill strategy introduced in Section 3.3.2, the test results can show whether this infill strategy is effective in improving approximation accuracy.

Due to the limitation in the computational budget, the number of links and signalised intersections in the test network should not exceed 50 and 10, respectively. Since the proposed surrogate-based real time signal is a centralised framework, however, there should be multiple traffic signals in the network. In addition, since DTA is not considered in the simulation, the geometry of the test network is restricted by the assumption that there is only

one route between each origin-destination (OD) pair. Besides the restrictions on geometry, the ideal test network is one that suffers from a congestion problem, and where demand has apparent temporal variations at a mesoscopic level (e.g. 15 min). The traffic data of the network should be collected over a relatively short time scale to capture the temporal-spatial congestion propagation.

The general information on the test network is provided in Section 3.4.1. Section 3.4.2 discusses the data collection and generation, while the simulation model used to obtain the network performance is shown in Section 3.4.3. Section 3.4.4 explains the reasons why the Kriging model is used to construct the response surface for the simulation test. Then, the numerical results are presented and analysed in Section 3.4.5.

3.4.1 The test network and numerical settings

Based upon the ideal requirements above, a test network in West Glasgow, Scotland was chosen. The outline of the network is shown in Figure 3.8, and consists of five signalised intersections and 35 links. These 35 links consist of eleven entrance links, ten exit links, six minor turning links and eight main links, marked as red in the Figure. The test period of the signal control spanned one hour from 8-9 am on a typical working day: June 7, 2010.

Figure 3.8 The test network in West Glasgow (Google Map)

The whole test period was divided into several time steps lasting for 5 s, which means that all the data were collected every 5 s. A one-hour simulation was undertaken with a 400 s warm-up session to ensure that the simulation results were not affected by the network loading. There were twelve signal control periods, each spanning five minutes, during which the signal control parameters remained constant. For a single simulation run, therefore, eleven sets of samples can be collected.

All the data used are held within the CARBOTRAF project funded by 7th EU framework (www.carbotraf.eu). Historical traffic flow data were collected by Sky High CountOnUs between 2007 and 2009 (Mascia et al., 2016). Since the traffic counts of each intersection were not collected on the same day, and thus may be affected by the temporal variation of the traffic network, all traffic counts were converted to the same reference day by using a scaling factor to avoid the bias error. The scaling factors take both day-of-the-week effect and seasonal effect into account. For the sake of simplicity, the average traffic demands of the eleven entrance links were aggregated to 15 minutes, and then used for the generation of the

time-varying source flow for the eleven inflow links. The vehicle turning ratio was estimated based on the turn-by-turn counts of each intersection. For the detailed information related to source flow and turning ratio, refer to **Appendix B**.

3.4.2 Data generation

3.4.2.1 Random demands

Since average traffic demand cannot reflect daily variations, synthetic data were randomly generated for each day around the averaged demands. The temporal variability of traffic demands at each boundary link of the network was perceived on two different time scales: a 7.5 min time window indexed by τ, and a smaller time step (e.g. 5 s) indexed by t. Thus it is assumed that the random inflow of a given link i is given by:

$$q_i(t) = \bar{q}_i(\tau) + \epsilon_i^\tau + \varepsilon_i^t \quad t \in \tau \tag{3.7}$$

where $\bar{q}_i(\tau)$ denotes the average demand of link i during time window τ; ϵ_i^τ represents random noise on the scale of time windows τ; ε_i^t represents random noise on the scale of time steps t. In other words, the two random terms represent mesoscopic and microscopic deviations from the mean value, respectively. Equation (3.7) generates complex and realistic demand profiles for traffic simulation. In practice, ϵ_i^τ and ε_i^t may be estimated from empirical data.

In this numerical case study, these noises were generated by drawing from given distributions. In order to generate sufficient samples to represent the stochasticity in the demand profiles, Equation (3.7) was followed by making ϵ_i^τ follow $\mathcal{N}(\mu_i^\tau, \sigma^2)$, where μ_i^τ is the average demand on link i during time window τ; and $\sigma = r \times \mu$ where $r = 0.2$ was used to parameterise the stochasticity of the demands (a larger r indicates higher variability). Let ε_i^t follow a uniform distribution: $U[-0.01, 0.01]$, and this short-term variation between each time step is far smaller than the variation between each time window. The short-term variation,

ε_i^t, is insignificant compared with the traffic flow of each link. The proposed distributions of the two random noises, ϵ_i^τ and ε_i^t, should be obtained from historical data theoretically but, due to the limitation of the data available in this empirical test, it was impossible to know the exact feature distributions of actual data. The fact that the proposed demand profile was not verified by actual data may create a large synthetic gap between actual and synthetic data. Furthermore, this gap will propagate if the actual data were to be used for final optimisation. To control this uncertainty, it was assumed that all demand data for testing and optimisation were generated from the same profile proposed above. This assumption helps to remove the influence of the synthetic gap on the final performance.

3.4.2.2 Signal control parameters

In this empirical test, the signal parameters were limited to the green stage times of every signalised intersection. For the sake of simplicity, all the other signal parameters, such as cycle time, phasing sequence and offset, were fixed. The design space of control parameters, u, can be expressed as:

$$\Lambda = \prod_{j=1}^{J} \Lambda_j \tag{3.8}$$

where j represents a signalised intersection, and the feasible region Λ_j for intersection j is expressed as:

$$\Lambda_j = \left\{ g_1, g_2, \dots, g_l \mid g_i \in [g_{min}, g_{max}], i = 1, \dots, l, \sum_{i=1}^{l} g_i = \Delta_j \right\} \tag{3.9}$$

where $g_1, \dots, g_l$ are the green times of all relevant stages, which are bounded by g_{min} and g_{max}, and sum up to be a constant Δ_j (which may be the cycle time minus amber and all-red times).

The feasible region expressed in Equation (3.9) is entirely based on signal timing constraints,

without any reference to *a priori* consideration of the suitability of a feasible control parameter. To narrow down the design space further, and focus on regions where the control parameters are likely to be close to the optimal under the prevailing demand profile, a baseline signal control was computed through an off-line optimisation. Once the baseline control parameters were known, the control parameters were then generated by randomly sampling its neighbourhood, as this can speed up the searching process (Cai et al., 2017). Here, the signal control samples were sampled by using the normal distribution around the benchmark, as shown in Equation (3.10), where $\hat{g}_{i,j}$ is the *j*-th sample of stage *i*, $\bar{g}_i$ is the green time of the *i*-th stage of the baseline signal control. For a normal distribution, 99.73% of the area is within the range of $[\mu - 3\sigma, \mu + 3\sigma]$. The baseline signal was a fixed-time signal control that the Glasgow City Council used to approximate the SCOOT system during the morning peak (Liu et al., 2015). Given the lower bound, g_{min}, and the upper bound, g_{max}, then σ can be estimated as $\frac{\bar{g}_i - g_{min}}{3}$ and $\frac{g_{max} - \bar{g}_i}{3}$, respectively. To ensure that most of the samples were within the boundary, σ is the minimum one of $\frac{\bar{g}_i - g_{min}}{3}$ and $\frac{g_{max} - \bar{g}_i}{3}$.

$$\hat{g}_{i,j} \sim N\left(\bar{g}_{i,} \frac{min\big((\bar{g}_i - g_{min}), (g_{max} - \bar{g}_i)\big)}{3}\right) \quad \forall i \in l, \forall j \tag{3.10}$$

Since the samples of each stage were generated separately, however, it was not possible to guarantee that the constraints in Equation (3.9) are satisfied, and thus the operator of projection needs to be applied. For each intersection, the projection can be formulated as the quadratic programming in Equation (3.11), where $\hat{\boldsymbol{g}} = (\hat{g}_1, \hat{g}_2, \ldots, \hat{g}_l)^T$ is a set of green times of all the stages sampled with Equation (3.10), and $\boldsymbol{g}$ is the projection of $\hat{\boldsymbol{g}}$ onto the feasible region. The projection follows the rule that the feasible solution, $\boldsymbol{g}$, is the one with minimum 2-norm distance with $\hat{\boldsymbol{g}}$ among all the feasible solutions.

$$\min_{\boldsymbol{g}} \frac{1}{2} \|\boldsymbol{g} - \hat{\boldsymbol{g}}\|^2 = \frac{1}{2} (\boldsymbol{g} - \hat{\boldsymbol{g}})^T (\boldsymbol{g} - \hat{\boldsymbol{g}}) \tag{3.11}$$

The solution $\boldsymbol{g}$ can be derived by applying the Karush-Kuhn-Tucker condition (Friesz, 2010), as shown in Equation (3.12)

$$\boldsymbol{g} = (g_1, g_2, \dots, g_l)^T, \quad g_i = \{\hat{g}_i - \lambda\}_{g_{min}}^{g_{max}} \tag{3.12}$$

where $\{\hat{g}_i - \lambda\}_{g_{min}}^{g_{max}}$ is derived with Equation (3.13)

$$\{\hat{g}_i - \lambda\}_{g_{min}}^{g_{max}} = \begin{cases} g_{min} & if\, \hat{g}_i - \lambda < g_{min} \\ \hat{g}_i - \lambda & if\, g_{min} \le \hat{g}_i - \lambda \le g_{max} \\ g_{max} & if\, \hat{g}_i - \lambda > g_{max} \end{cases} \tag{3.13}$$

The projection of the signal sample is therefore transformed into the problem of finding the value of λ satisfying the constraint in Equation (3.14). This algebraic equation can be easily solved numerically.

$$\sum_{i=1}^{l} \{\hat{g}_i - \lambda\}_{g_{min}}^{g_{max}} = \Delta_j \tag{3.14}$$

It should be noted that the minimum and maximum green times (g_{min} and g_{max}) of all the stages are the same in the equations above. In fact, each stage may be assigned with different minimum green and maximum green times. The assumption of equal g_{min} and g_{max} can easily be relaxed by replacing g_{min} and g_{max} with g_{min}^i and g_{max}^i which represent the specific boundary conditions of each stage. In this case, the equations above are still valid.

A simple signal plan was adopted in this empirical test, and the phase configuration is displayed in Figure 3.9. In practical application, the phase configuration should be carefully designed to ensure the service of movements and safety. In this test, the number of phases is assumed to be equal to the number of links terminated at the intersection, and each link is given green only once. In addition, only one link is given green in each stage. Once the link is

allowed to move, then all the movements leaving this link are given green. This simple phase configuration can ensure completely conflict-free driving without considering any early cut-off or late release. The intersection with three links has a similar configuration but with three phases only.

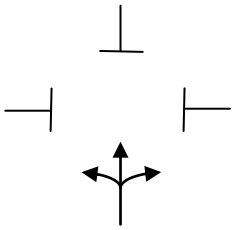
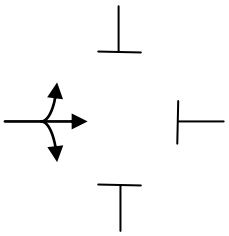
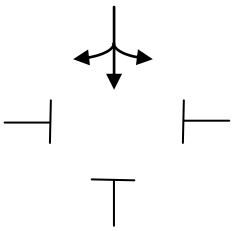
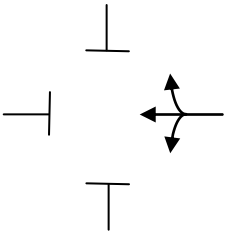
Phase 1	Phase 2	Phase 3	Phase 4
			

Figure 3.9 Phase configuration of the intersection with four links

Besides the phase configuration introduced above, the following specifications were also applied:

1. **No pedestrian phase was considered.**

   Typically, the phase configuration should consider the movements of pedestrians as well. In this test, however, the focus was on managing vehicle movements and therefore it was reasonable to neglect the pedestrian stage.

2. **No inter-stage time was considered, and the change of stage occurred instantaneously**.

   In practice, for safety reasons, inter-stage periods such as amber time and all-red time are designed to allow the drivers to react to signal changes and leave the intersection before the conflict streams are allowed to move. In this test, however, for the sake of simplicity, inter-stage time was not considered. This means that the total green time of all the stages is equal to the cycle time. Nevertheless, this assumption can be easily relaxed by including additional lost time.

Even though this empirical test was conducted under the above assumptions, it is important to know that the surrogate model does not set any restriction related to signal control. These assumptions can be relaxed easily by introducing a more complex but more realistic phase configuration.

According to the specifications described above, the sum of the green time of all the stages is equal to the cycle time for this particular problem. The cycle time of all the signalised intersections was 100 s. The lower and upper bounds of the signalised intersection with four links were 15 s and 55 s, while those of a three-link intersection were 14 s and 72 s, respectively.

$$\Lambda_j = \left\{ g_1, g_2, g_3, g_4 |\, g_i \in [15, 55], i = 1, 2, 3, 4 \sum\nolimits_i g_i = 100 \right\}$$

or (3.15)

$$\Lambda_j = \left\{ g_1, g_2, g_3 |\, g_i \in [14, 72], i = 1, 2, 3 \sum\nolimits_i g_i = 100 \right\}$$

3.4.3 Extraction of state variables and objectives from the simulation model

3.4.3.1 Link-based Kinematic Wave Model

Given the network demand and signal control parameter, u, generated in section 3.4.2, the corresponding outputs need to be obtained from a simulation model. In this test, the continuous-time Link-based Kinematic Wave Model (LKWM) proposed by Han, Piccoli & Szeto (2015) was adopted to simulate the network dynamics. LKWM is a reformation of the LWR equation (Lighthill & Whitham, 1955, Richards 1956) using its equivalent Hamilton-Jacobi equation, for which a variational approach known as the Lax-Hopf formula is applied to obtain semi-analytical solutions. The resulting system of differential algebraic equations is elaborated in Han, Eve & Friesz (2019). It is a continuous-time model, such that the definition and behaviour of the model are not affected by the time step size. In addition, unlike some macroscopic models that use density as the representation of network evolution, LKWM uses vehicle flow as input variables, which makes it suitable to be integrated with the traffic control problem, because traffic controls typically act on vehicle flow.

The conservation law of the LWR equation, which represents the evolution of density, both temporally and spatially, is presented in Equation (3.16):

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}f\big(\rho(x,t)\big) = 0 \tag{3.16}$$

where $\rho(x,t)$ represents the density at location x and time t. $f\big(\rho(x,t)\big)$ is the traffic flow expressed as a function of density. The function $f(\cdot)$: $[0,\ \rho^{jam}] \rightarrow [0,\ C]$ can be presented as the fundamental diagram, where ρ^{jam} and C represent the jam density and the flow capacity, respectively. LKWM employs the triangular fundamental diagram, as shown in Figure 3.10. The two axes represent the link flow and density. The critical density, ρ_{crit}, is the density of the link when the link flow reaches its capacity. Critical density, together with flow capacity, divides the triangular fundamental diagram into two parts: the uncongested condition (left) and the congested condition (right). Equation (3.17) describes the relationship between flow and density under both of the conditions, where *v* and *w* represent the forward and backward propagating wave speed, respectively.

$$f(\rho) = \begin{cases} v\rho, & \rho \in [0,\sigma] \\ -w\big(\rho - \rho^{jam}\big), & \rho \in (\sigma,\rho^{jam}] \end{cases} \tag{3.17}$$

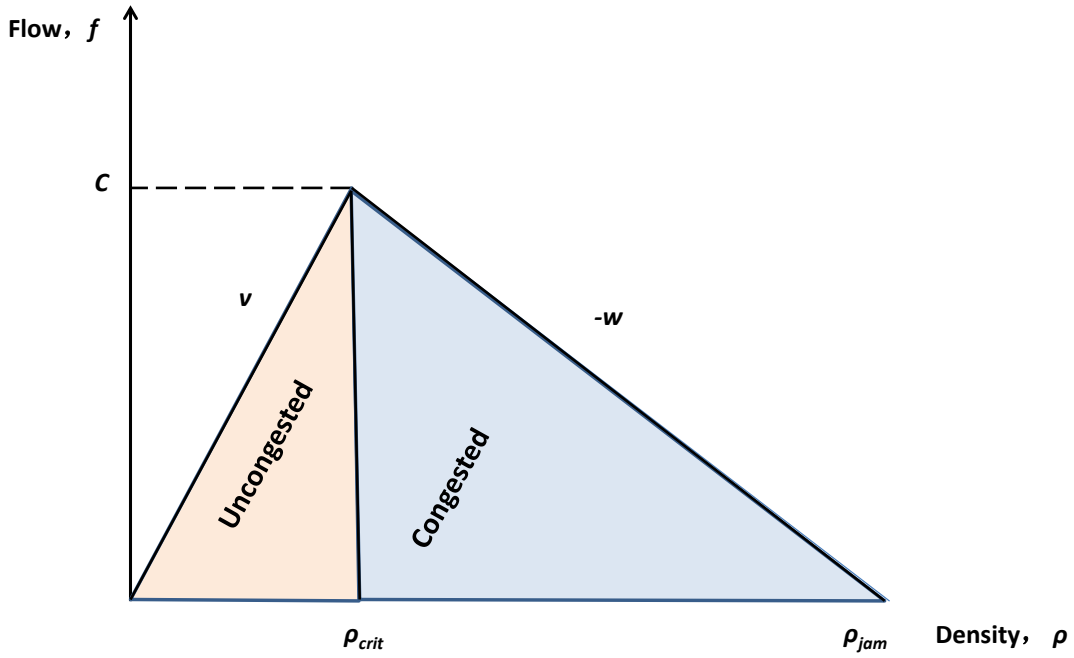


Figure 3.10 Triangular fundamental diagram

In LKWM, a separating shock can divide the link into the uncongested region and congested region. It is assumed that there is, at most, one separating shock in a link. This allows the congestion propagation in the network to be modelled. The Lax-Hopf formula is employed to detect the location of the separating shock. When the separating shock reaches the boundaries of the link, a latent separating shock is formed, and the traffic conditions at the two boundaries are related to each other. Equations (3.18) and (3.19) describe the interaction of two boundaries of a link with spatial boundary [a, b] when a separating shock is formed.

$$x^*(t) = a \quad \leftrightarrow \quad N_{up}(t) = N_{down}\left(t - \frac{L}{w}\right) + \rho^{jam}L \tag{3.18}$$

$$x^*(t) = b \quad \leftrightarrow \quad N_{up}\left(t - \frac{L}{v}\right) = N_{down}(t) \tag{3.19}$$

where $x^*(t)$ is the location of the separating shock at time t, $N_{up}(t)$ and $N_{down}(t)$ are the cumulative vehicles entering and leaving the link at time t, and L is the length of the link. Equation (3.18) shows that when the separating shock reaches the upstream boundary, the

whole link becomes congested, with the wave moving backwards at a speed of w. The cumulative vehicles entering the link is therefore the sum of $N_{down}\left(t-\frac{L}{w}\right)$ and total vehicles within the link. In Equation (3.19), meanwhile, the whole link is uncongested as the separating shock is at the downstream boundary. The cumulative vehicle counts at both of the boundaries are correlated with each other with the relationship shown in Equation (3.19).

The inflow and outflow are the derivatives of the cumulative counts of vehicles entering and leaving the link, respectively, as shown in Equation (3.20). For the junction v, the links connected to it fall into two categories: incoming links, denoted as I^v, and outgoing links denoted as O^v. The demand $D_i(t)$ of the incoming link, I, and the supply, $S_j(t)$, of outgoing link, j, can be formulated as Equations (3.21) and (3.22), respectively; where $D_i(t)$ is the maximum flow that can leave the link. In Equation (3.21), the first case shows that when the downstream boundary is uncongested, the number of vehicles that can leave the link depends on the inflow $\frac{L_i}{v_i}$ ago. When the downstream boundary is congested, however, demand is equal to link capacity. The supply in Equation (3.22) is defined as the maximum flow that can enter the link. When the entrance of the link is congested, supply depends on how many vehicles left the link $\frac{L_j}{w_j}$ ago, otherwise it is equal to the link capacity.

$$\frac{\partial}{\partial t}N_{up,i}(t)=q_{in,i}(t),\ \ \frac{\partial}{\partial t}N_{down,i}(t)=q_{out,i}(t) \tag{3.20}$$

$$D_i(t)=\begin{cases} q_{in,i}\left(t-\frac{L_i}{v_i}\right), & if\ N_{up,i}\left(t-\frac{L_i}{v_i}\right)=N_{down,i}(t) \\ C_i, & if\ \ N_{up,i}\left(t-\frac{L_i}{v_i}\right)>N_{down,i}(t) \end{cases} \quad \forall\, i\in I^v \tag{3.21}$$

$$S_j(t)=\begin{cases} q_{out,j}\left(t-\frac{L_j}{w_j}\right), & if\ N_{up,j}(t)=N_{down,j}\left(t-\frac{L_j}{w_j}\right)+\rho_j^{jam}L_j \\ C_j, & if\ N_{up,j}(t)<N_{down,j}\left(t-\frac{L_j}{w_j}\right)+\rho_j^{jam}L_j \end{cases} \quad \forall\, j\in O^v \tag{3.22}$$

The traffic dynamics at the intersection can be solved by Riemann Solvers. Under the assumption that only one link is allowed to move across the junction each time, the intersection is always a diverge intersection with only one incoming link. Figure 3.11 shows the layout of a diverge junction with two outgoing links.

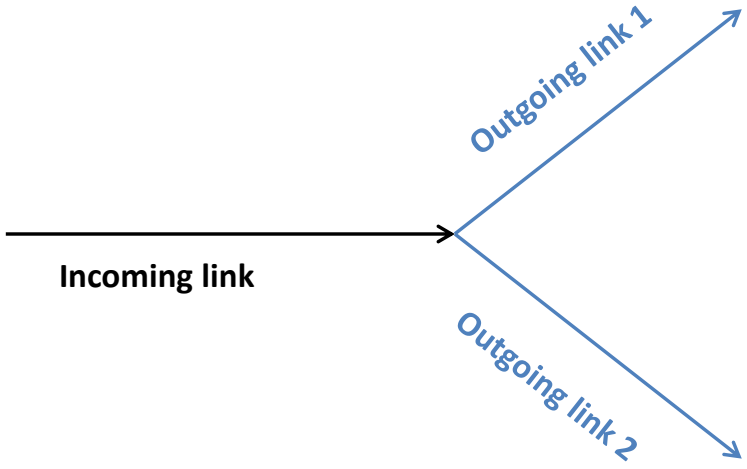


Figure 3.11 Layout of diverge junction

The traffic dynamics at the junction can then be expressed by Equation (3.23) and (3.24), where $q_{out,i}(t)$ is the outflow of incoming link *I*, which depends on how many vehicles outgoing links can receive and its own demand. $q_{in,j}(t)$ is the inflow of outflow link *j*, which is equal to the product of turning ratio $\alpha_{i,j}$ and the outflow of incoming link *i*. The turning ratio $\alpha_{i,j}$ describes the percentage of flow from link *i* entering link *j*. By solving the Equations (3.10) – (3.24) sequentially, the network dynamics within the whole control period can be derived.

$$q_{out,i}(t) = min\left(D_i(t), \min_{j \epsilon O^v}\left(\frac{S_j(t)}{\alpha_{i,j}}\right)\right) \tag{3.23}$$

$$q_{in,j}(t) = \alpha_{i,j} q_{out,i}(t) \qquad \forall\, j \in O^v \tag{3.24}$$

The LKWM can be converted to a time-discretisation version, as below, by dividing the

control period into time intervals lasting Δt.

Equation (3.25) – (3.30) are presented using the notations listed below.

$N_{up,i}^k$	Cumulative number of vehicles entering the link i at time step k
$N_{down,i}^k$	Cumulative number of vehicles leaving the link i at time step k
$q_{in,i}^l$	Inflow to link i at time step k
$q_{out,i}^l$	Outflow from link $i$at time step k
Δ_i^f	Number of steps for forward wave to travel to downstream boundary
Δ_i^b	Number of steps for backward wave to travel to upstream boundary
L_i	Length of link i
v_i	Forward wave speed of link i
w_i	Backward wave speed of link i
D_i^k	Demand of link i at time step k
S_j^k	Supply of link j at time step k
C_i	Flow capacity of link i
$\alpha_{i,j}$	Turning ratio from link i to link j
I^v	Set of incoming links of junction v
O^v	Set of outgoing links of junction v

$$N_{up,i}^k = \Delta t \sum_{l=1}^{k} q_{in,i}^l \,, \qquad N_{down,i}^k = \Delta t \sum_{l=1}^{k} q_{out,i}^l \tag{3.25}$$

$$\Delta_i^f = \left[\frac{L_i}{v_i \Delta t}\right] \qquad \Delta_i^b = \left[\frac{L_i}{w_i \Delta t}\right] \tag{3.26}$$

$$D_i^k = \begin{cases} q_{in,i}^{k-\Delta_i^f}, & if\ \Delta t \sum_{l=1}^{k-\Delta_i^f} q_{in,i}^l = \Delta t \sum_{l=1}^{k} q_{out,i}^l \\ C_i, & if\ \Delta t \sum_{l=1}^{k-\Delta_i^f} q_{in,i}^l > \Delta t \sum_{l=1}^{k} q_{out,i}^l \end{cases} \quad \forall\ i \in I^v \tag{3.27}$$

$$S_j^k = \begin{cases} q_{out,j}^{k-\Delta_j^b}, & if\ \Delta t \sum_{l=1}^{k} q_{in,j}^l = \Delta t \sum_{l=1}^{k-\Delta_j^b} q_{out,j}^l + \rho_j^{jam} L_j \\ C_j, & if\ \Delta t \sum_{l=1}^{k} q_{in,j}^l < \Delta t \sum_{l=1}^{k-\Delta_j^b} q_{out,j}^l + \rho_j^{jam} L_j \end{cases} \quad \forall\ j \in O^v \tag{3.28}$$

$$q_{out,i}^k = min\left(D_i^k, \min_{j \epsilon O^v}\left(\frac{S_j^k}{\alpha_{i,j}}\right)\right) \tag{3.29}$$

$$q_{in,j}^k = \sum_{i \in I^v} \alpha_{i,j}\, q_{out,i}^k \tag{3.30}$$

Signalised Junction

Signal control can be easily integrated with this simulation model by converting the signal plan into binary variables $p_i^k \in [0,1]$ representing the right of movement of each link *i* at time step *k*. If it is green, then p_i^k=*1*, otherwise p_i^k=*0*. Then q_{out} can be modified as follows:

$$q_{out,i}^k = min\left(D_i^k, p_i^k \times \min_{j \epsilon O^v}\left(\frac{S_j^k}{\alpha_{i,j}}\right)\right) \tag{3.31}$$

Entrance Link

The entrance links of the network need to be treated differently. By adopting the notation in the cell transmission model, the entrance links can be regarded as source units that generate the flow. The upstream links connecting entrance links are not modelled, and hence the actual inflow cannot be calculated from the outflows of upstream links. Thus Equation (3.32) is established, where d_i^l is the flow that is going to enter the source links *i* at time step *l*. The inflow of entrance link *i* is calculated by considering two conditions: (1) no point queue outside the link and (2) a point queue is formed outside the link. This is determined by whether the sum of inflow, i.e. vehicles actually entering the link, is less than the sum of d_i^l, vehicles going to enter the link. If it is, then a queue is considered to have formed outside the link.

$$\begin{cases} q_{in,i}^k = min\left(S_i^k, d_i^k\right) & if\ \Delta t \sum_{l=1}^{k} q_{in,i}^l = \Delta t \sum_{l=1}^{k} d_i^l \\ q_{in,i}^k = S_i^k & if\ \Delta t \sum_{l=1}^{k} q_{in,i}^l < \Delta t \sum_{l=1}^{k} d_i^l \end{cases} \quad \forall i \in I \tag{3.32}$$

In order to account for the queue accumulated outside the network, the cumulative number of vehicles entering the link is calculated with d_i instead of $q_{in,i}$. Equation (3.33) shows the expression for the cumulative inflow vehicles at the entrance link.

$$N_{up,i}^k = \Delta t \sum_{l=1}^{k} d_i^l \tag{3.33}$$

Exit Link

The exit links of the network can be seen as sink units and are assumed to be always uncongested. Without considering the actual length of exit links, inflow is equal to outflow, and vehicles can leave the network as soon as they enter the exit links. This is because the exit links are not considered as a part of the network and can be seen as virtual links to estimate

how many vehicles leave the network. The only constraint on the exit link is therefore the link capacity, and hence the supply is always equal to the link capacity. Equations (3.34) – (3.36) show the relationship between inflow and outflow for the exit link *j*.

$$S_j^k = C_j \tag{3.34}$$

$$q_{in,j}^k = \sum_{i \in I^v} \alpha_{i,j}\, q_{out,i}^k \tag{3.35}$$

$$q_{out,j}^k = q_{in,j}^k \tag{3.36}$$

General Assumptions

It is evident that if the simulation model itself is not well calibrated, the information obtained from it may be misleading and serve to deteriorate the estimation capability of the response surface. The validation and calibration of the simulation model are crucial but challenging because of the large amount of data required. In this numerical experiment, however, the validation of the simulation model against reality is not discussed. The following assumptions are made as the primary objective of this experiment is to gain insight into the approximation capability of a surrogate model.

1. The traffic flow is assumed to be homogeneous.

This is an assumption related to the macroscopic nature of LKWM. Macroscopic models do not consider the distinction between individual behaviours and such detailed information is not necessary if the objective is to optimise network-level performance (i.e. total throughput) (Nahi, 1973). In this test, the purpose was to evaluate the overall performance of the entire network under certain traffic conditions and traffic controls. Therefore, it is reasonable to use a macroscopic model with aggregated variables that reflect overall flow characteristics, which are assumed to be homogeneous.

In contrast, a microscopic model is time-consuming, since it assigns specific behavioural rules to each individual or group of vehicles. A microscopic model can therefore capture the complex interactions between vehicles, or between vehicles and infrastructure, in more detail, but this leads to a lack of tractability. The selection between macroscopic and microscopic approaches is a trade-off between efficiency and realism. In this test, the simulation model needs to be run multiple times to generate enough training points and candidate points for infill in this test. Given that microscopic simulation increases the computational burden, typically microscopic models are used to evaluate and compare the performance of a limited number of management alternatives. Such an approach would not be efficient in this numerical test, although it is worth noting that this conceptual framework imposes no requirement as to what type of simulation model can be used. In other words, if sufficient computational budget is available, a microscopic model can be used as readily as a macroscopic one.

2. The exogenous parameters are assumed to be deterministic.

This is an assumption related to the deterministic nature of LKWM. In reality, the behavioural parameters are not stationary due to the stochastic nature of the traffic system. The simulation model that accounts for this stochastic nature is a stochastic simulation model. In a stochastic simulation model, exogenous parameters are randomly selected from given distributions, and hence different outputs are generated from multiple runs, even with the same inputs. The result of a single run of a stochastic model is therefore only a realisation of a random number and usually cannot represent the true result. A surrogate model, on the other hand, constructs a deterministic relationship between traffic state, signal control and network performance, and deterministic models have fixed parameters such that given inputs always correspond to fixed outputs. Thus, the output of a stochastic model cannot be used directly to construct a surrogate model. The implication of this is that a deterministic model should be used in this test to estimate the expected network performance under a given traffic state and signal control.

A stochastic model can still be used to estimate network performance for the construction of a

surrogate model, however, by using the mean of the outputs obtained from multiple runs. Theoretically, with infinite runs, the average of outputs converges to their expectation. Increasing the number of simulation runs, however, increases the computational time, especially given that nearly all the stochastic models are microscopic models, which is even more time-consuming. For this reason, a deterministic model is preferred since its output is equivalent to the expectation of outputs from a stochastic model to a known extent.

3. The network is assumed to be under normal conditions.

The traffic dynamics may change abruptly because of the influence of unpredictable events. It is hard for a simulation model to emulate the traffic system under these abnormal conditions. In addition, the inclusion of network behaviour under abnormal conditions complicates the relationship between state, control and objective, which hinders the surrogate model from making an estimation accurately. These events occur with low probability, however, and usually recover in a short time. Hence, the network under abnormal conditions is not simulated since the aim is to relieve recurrent congestion.

4. The vehicles are assumed to accelerate and decelerate instantaneously, and reaction time is neglected.

In reality, vehicles accelerate from an idle state or decelerate to an idle state gradually when the signal changes. For the sake of simplicity, however, in this test the acceleration and deceleration of vehicles are not simulated. This assumption can be easily relaxed, however, by replacing the actual green time by the effective green time.

5. The signal is assumed to have the same number of stages as the incoming links. Within each stage, only one link is allowed to move.

As illustrated in Section 3.4.2.2, a simple phase configuration is adopted. This simple phase configuration makes sure that all junctions work as diverge junctions, such that only one link is allowed to move each time. This assumption guarantees the validity of Equation (3.23) and (3.24). The current phase configuration can be replaced with a more complex but realistic phase structure, however, although the equations representing the junction dynamics would

need to be modified by introducing a new concept of merge junctions. For the simulation of traffic dynamics at a merge junction refer to Han, Piccoli & Szeto (2015).

3.4.3.2 State variables and objectives

The performance of the surrogate model for real time control relies on the careful selection of traffic state variables, which depends on the characteristics of traffic networks. Traffic state variables should have the ability to reflect the current traffic condition accurately. The following sections illustrate the selection of state variables and objectives.

State variables

Since LKWM employs the flow and the integral of flow as the primary variables, a number of traffic states can be simulated. Here, the relative occupancy (RO) is considered as the state variable for link i:

$$RO_i(t) = \frac{N_{up,i}(t) - N_{down,i}(t)}{L_i \rho_i^{jam}} \tag{3.37}$$

where $N_{up,i}(t)$ and $N_{down,i}(t)$ are, respectively, the cumulative entering and exiting vehicle counts of link i. The difference between $N_{up,i}(t)$ and $N_{down,i}(t)$ denotes the number of vehicles remaining in the link at time t, and thus, by definition, the division between the number of vehicles remaining in the link and the length of the link $\frac{N_{up,i}(t)-N_{down,i}(t)}{L_i}$ is the current density of the link. L_i and ρ_i^{jam} denote link length and jam density, respectively. The relative occupancy is a measure of link-level congestion. The RO is chosen over link flow (e.g. entrance or exit flow) in this thesis because the latter does not correspond to a unique traffic state (Lighthill & Whitham, 1955), and does not correctly reflect traffic congestion in the case of vehicle spillback (Han et al., 2016b).

In this case study, the RO of all eleven entrance links of the network, as well as the eight major links, are compiled as the state vector of the traffic system (see Figure 3.8). The reason

that RO is used instead of inflow is because of the possibility of vehicle spillback on these links (Han et al., 2016b), in which case flow is not a reliable indicator of congestion. To account for the latency effect of traffic dynamics, and to eliminate the effect of on-and-off signal controls, the RO of each link is averaged over the past 20 time steps (100 s), which equals a full signal cycle time.

Objectives

The surrogate-based optimisation is performed at the end of every control period, and the following quantity was chosen as its objective to be maximised:

$$Network\ throughput = \sum_{t \in T_p} \sum_{i \in O} f_i^{out}(t) \times dt \tag{3.38}$$

where $f_i^{out}(t)$ denotes the outflow of link i during time step t, dt is the step size, T_p denotes the set of time steps in control period p. O denotes the set of exit links of the network, and hence network throughput represents the total number of vehicles leaving the network within the control period T. This quantity may be measured in a real-world network through flow detectors (e.g. inductive loops or microwave detectors).

3.4.4 Surrogate model selection and construction

In this test, there were 19 state variables and 19 signal variables. Therefore, lower order polynomials cannot approximate the complex relationship accurately. High order polynomials are not applicable either since too many parameters need to be estimated. Both RBF and Kriging have more complex structures and are more appropriate to deal with this problem. Compared with RBF, Kriging has extra parameters to represent the width of each dimension and allows the variables that are more crucial to the network performance to be identified. In addition, the Gaussian function in the Kriging model permits the prediction error of any point in the design space to be estimated without any true observations. The Kriging model was

therefore selected in this work.

Firstly, the surrogate model was constructed with eleven training points. Then, during each infill iteration, ten infill points were selected from 550 candidate points according to the infill strategy described in Section 3.3.2. For a Kriging model, the prediction error of any point in the design space can be easily estimated without needing to recall the simulation model. The infill stopped when the total number of training points reached 451, as no significant improvement in global accuracy could be achieved by adding additionally points. Figure 3.12 shows the detailed steps of the Kriging model construction, and the data communication between each step.

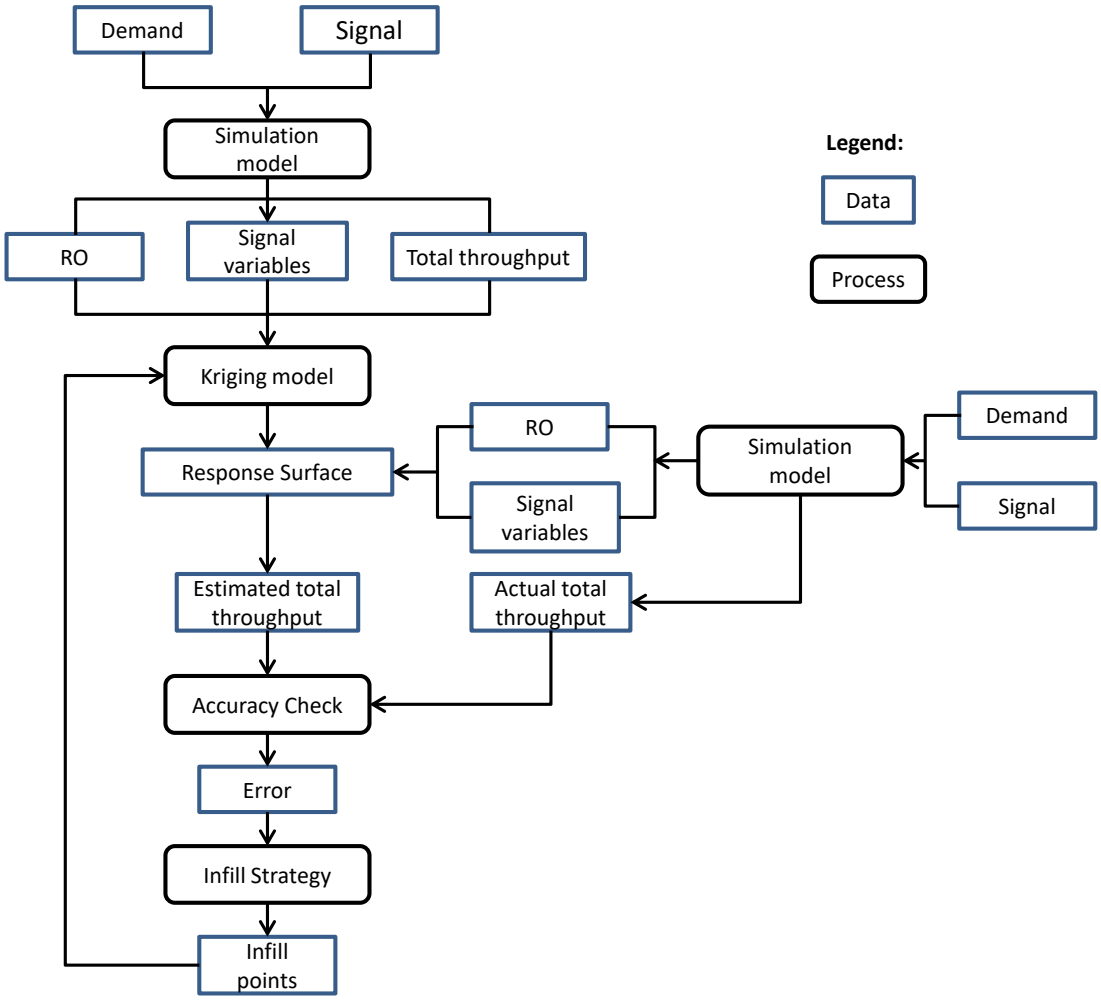


Figure 3.12 Data communication of Kriging model construction

Construction of a surrogate model contains the following sources of uncertainty:

- **Structural uncertainty**: uncertainty caused by the simplifications and assumptions made to the surrogate model. The more complex the surrogate model structure is, the more realistic it will be. On the contrary, the more complex the actual network, the more difficult it is for surrogate model to capture the dynamics accurately. The selection of the model is problem-dependent.

- **Parameter uncertainty**: uncertainty related to model parameters. The parameters of the surrogate model are completely determined by the samples selected to train the model. Parameter uncertainty refers to errors in the parameter estimation and boundary conditions.

- **Improper sampling**: uncertainty due to improper sampling. As a data-driven model, the quality of the surrogate model is affected by the quality of the samples used to construct the surrogate model.

Since these uncertainties are the major causes of estimation error, their quantification by external validation makes it possible to assess the estimation ability of surrogate models. The parameters in surrogate models are calibrated by training points, and parameter uncertainty mainly arises due to a lack of training points. It can therefore be reduced by adding more points to the training set. Because of the inherent structural uncertainty of surrogate models, however, the estimation error cannot be completely eliminated. The sampling arrangement also affects the estimation capability of surrogate models. Hence, the behaviours of surrogate models with different training points will be slightly different. If several surrogate models with the same number of training points but different sample arrangements are constructed, differences in their performance indicate the influence of sampling on approximation accuracy.

3.4.5 Result analysis

Approximation accuracy

To average the influence of initial samples, 50 Kriging models with randomly generated

initial samples were established. In order to avoid the influence of the sampling of candidate points, all the Kriging models selected infill points from the same set of candidate points. The approximation accuracy of Kriging models was evaluated with 11000 randomly-generated testing points. MAPE and RMSE were calculated to determine how accuracy of the model.

Figure 3.13 and Figure 3.14 below show the evolution of the average RMSE and MAPE of 50 Kriging models with the increase of training points. The 95% confidence interval, which is defined as $[\mu - 1.96\sigma, \mu + 1.96\sigma]$, is also drawn in the figure. The light orange curves represent how the RMSE and MAPE of the individual Kriging models varied. It can be seen that both average RMSE and MAPE decrease with the growth of training points. The results of the test show that the proposed infill strategy can enhance the global accuracy effectively, and that parameter uncertainty can be reduced if more points are added to the training set. When the number of training points was increased from 11 to 451, the RMSE and MAPE averages decreased by around 40% and 37%, respectively. Ultimately, the RMSE was less than 12, and the MAPE less than 3.5%, which indicates that the applied Kriging model had high estimation capability and low inherent structural uncertainty. It is also important to observe the flattening trends of the average RMSE/MAPE, seen in Figure 3.13 and Figure 3.14, which highlights the need to select the appropriate amount of infill points to balance the accuracy and complexity of the model. In other words, the contribution of the same number of infill points to the improvement of accuracy becomes less significant when there are already a large number of points in the training set. Thus, it is important to determine when to stop the infill strategy.

Moreover, the area enclosed by the 95% confidence interval becomes smaller as the number of training points is increased. In the beginning, the large enclosed region indicates that the 50 Kriging models performed quite differently due to the uncertainty of initial sampling, confirming that the Kriging model is sensitive to the arrangement of initial samples. The area bounded by the confidence interval decreased, however, as the number of points increased. This phenomenon shows that increasing the number of training points decreases the variance of estimation capability, and reduces the sensitivity of Kriging models to initial samples. This

decreasing trend of variance can be seen more clearly from Figure 3.15 and Figure 3.16, which show the coefficient of variation of RMSE and MAPE. The coefficient of variation (CV) is the division between standard deviation and mean, as expressed in Equation (3.39). It describes the degree of dispersion around the mean value. The curve of CV shows an overall decreasing trend with fluctuations around it, which are caused by the random components in the sampling process. The CV of RMSE and MAPE ultimately coalesce around 5%, which is low enough to conclude that Kriging models have similar estimation capability when they all have 451 training points.

$$CV = \frac{Standard\ Deviation}{Mean} \times 100\% \tag{3.39}$$

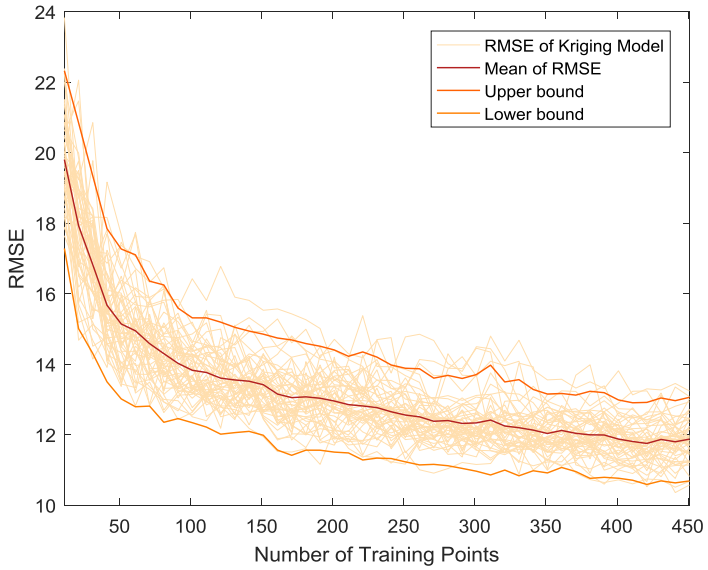


Figure 3.13 The RMSE of the 50 Kriging models with different numbers of infill points. The centreline represents the average, and the upper and lower bounds represent the 95% confidence interval

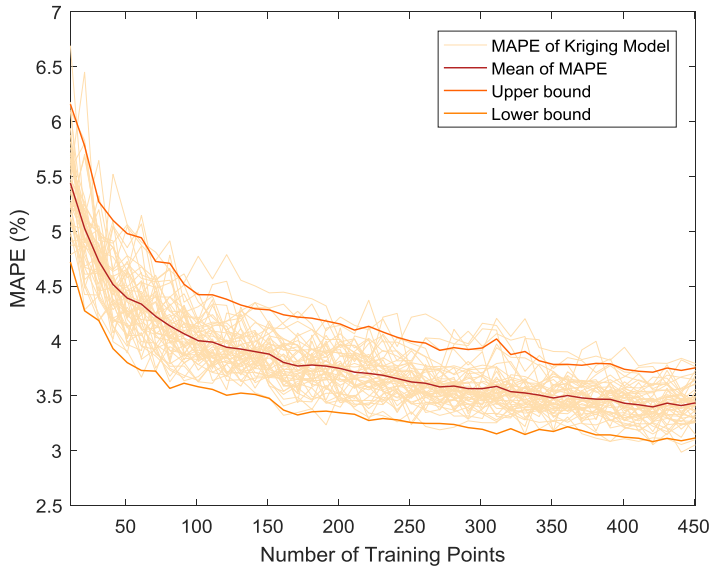


Figure 3.14 The MAPE of the 50 Kriging models with different numbers of infill points. The centreline represents the average, and the upper and lower bounds represent the 95% confidence interval

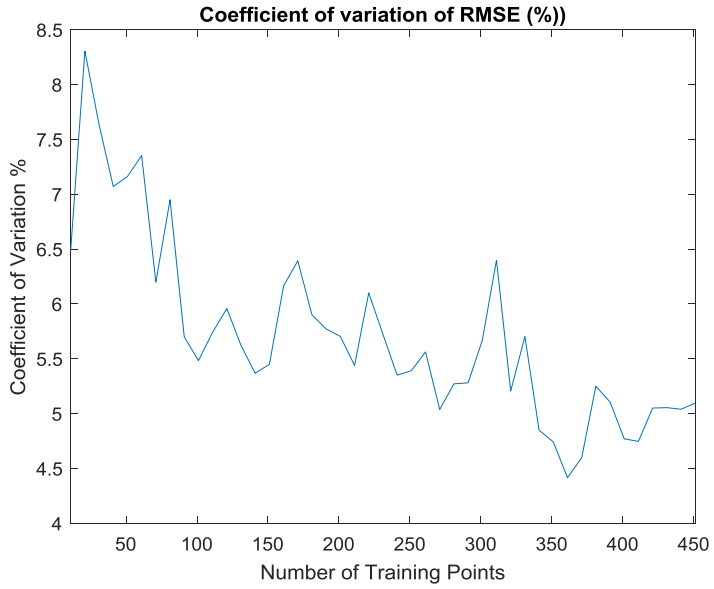


Figure 3.15 Average coefficient of variation of RMSE of Kriging models with different number of infill points

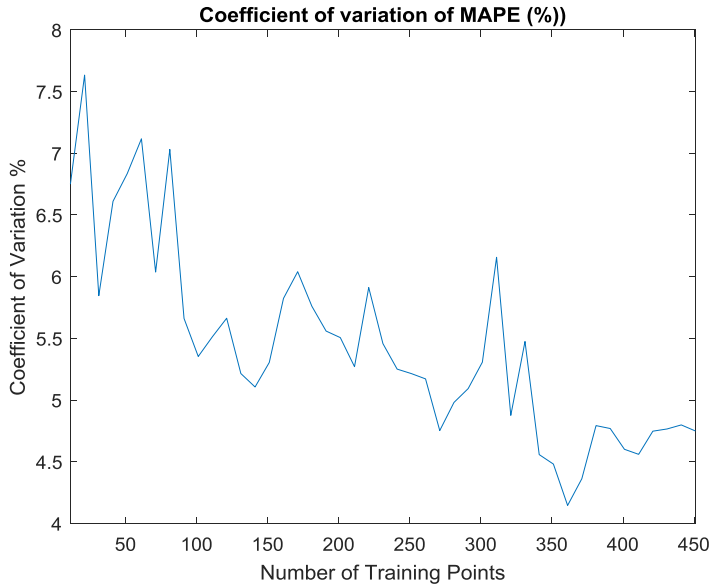


Figure 3.16 Average coefficient of variation of the MAPE of Kriging models with different numbers of infill points

Figure 3.17 and Figure 3.18 show the cumulative distribution function of the RMSE and MAPE of Kriging model with 201, 251, 351 and 451 training points over 50 replications. They show as the number of training points were increased, the average values of the MAPE and RMSE decreased and their distribution improved. From an overall perspective, Kriging models with more training points have better estimation capability than those with fewer points. The curve of Kriging-251 has similar distances to those of Kriging-201 and Kriging-351, which indicates that levels of improvements on global accuracy are similar. In addition, adding 100 points to Kriging-351 results in little improvement in estimation accuracy, since the curves of Kriging-351 and Kriging-451 are close to each other. This leads to the conclusion that the contribution that the same number of infill points can make to the improvement of global accuracy decreases as the number of training points increases.

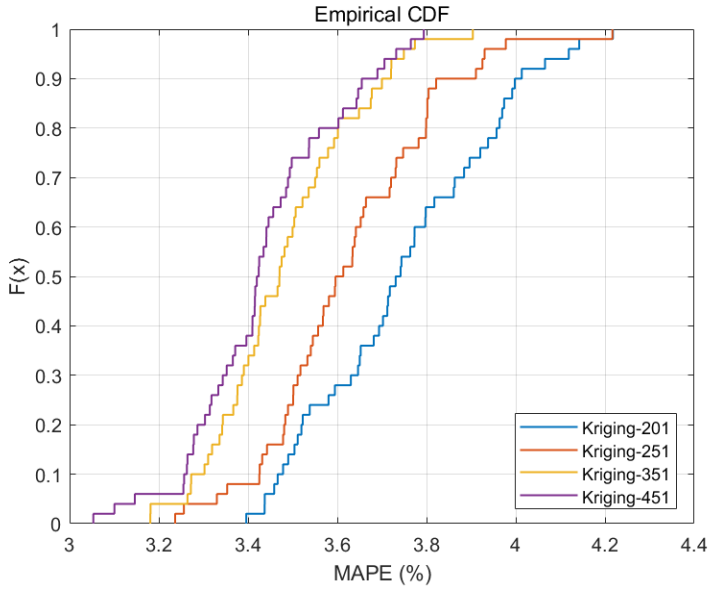


Figure 3.17 Empirical CDF plot of the MAPE of Kriging models with 201, 251, 351 and 451 training points

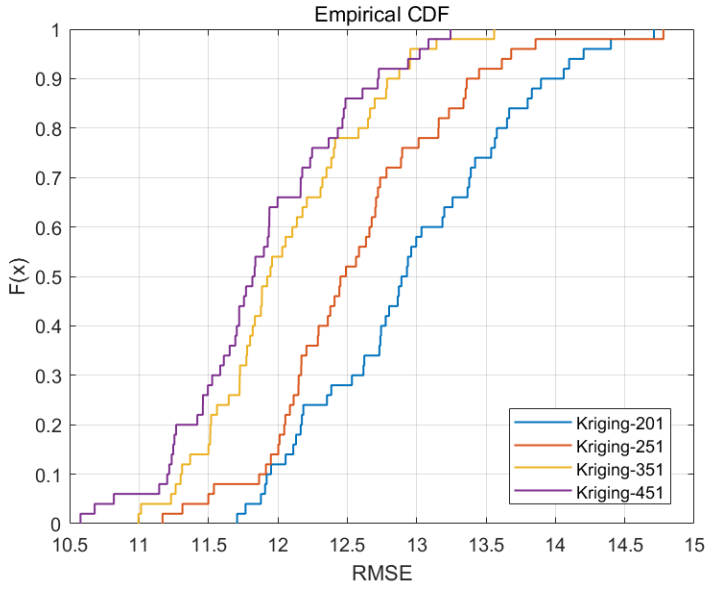


Figure 3.18 Empirical CDF plot of the RMSE of Kriging models with 201, 251, 351 and 451 training points

In order to see in detail how MAPE varies, three Kriging models were selected randomly and the evolution of MAPE recorded and plotted in Figure 3.19. The curves show a global downward trend as the number of training points grows. Each individual curve, however, displays small fluctuations in the process of adding infill points. This means that, sometimes,

MAPE increases when specific infill points are added. This phenomenon has been mentioned in the book written by Forrester, Sobester & Keane (2008). They found there is a good deal of scatter in the trend of lower MAPE when there are a larger number of training points in the Kriging prediction due to the random elements in the sampling plan generation and Kriging parameter tuning. This scattering effect within each individual Kriging model leads to the fluctuations in CV seen in Figure 3.15 and Figure 3.16.

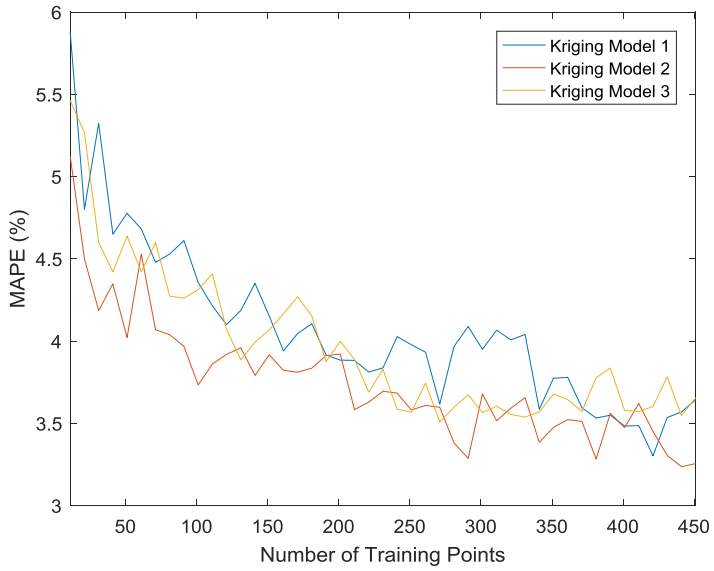


Figure 3.19 MAPE variations of three randomly-selected Kriging models

Computational Efficiency

One of the advantages of Kriging model is its high efficiency. The following results demonstrate how efficient it is in terms of model construction and estimation. In Figure 3.20, the computation time for model construction exhibits a nonlinear growth as more training points are involved. The most time-consuming part of Kriging model construction is the generation of the covariance matrix. The size of the covariance matrix depends on the total number of training points: therefore the construction time grows exponentially with the number of training points. Even for the Kriging model with 451 points, however, the construction time was only 20 s, which is short enough to prove the efficiency of Kriging model in respect to model construction.

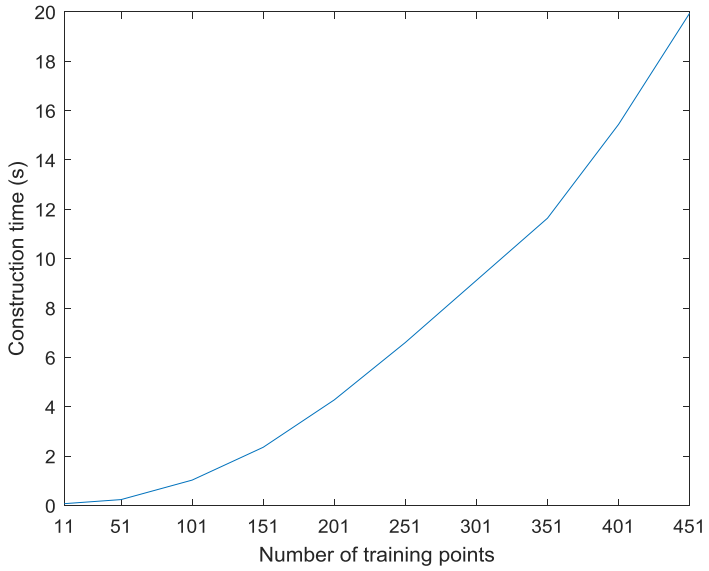


Figure 3.20 Construction time against the number of training points

The estimation time of LKWM, as well as Kriging models with 201, 251, 351 and 451 training points, are summarised in Table 3.2. The data in the last column is the time consumed by LKWM to simulate the traffic dynamics for five minutes. To average the randomness of estimation time, 100 replications were performed. The Kriging model with more training points was found to have a longer estimation time. Even though the correlation matrix between the training points was already stored before estimation, the correlation matrix between all the training points and the untried point still needed to be constructed, and this is also affected by the number of training points. This is the reason why the estimation time increases when the Kriging model is constructed with more training points. Nonetheless, even for the Kriging model with 451 points, the estimation time of 1.9432×10^{-4} s is much shorter than the average estimation time of LKWM. Furthermore, if LKWM were to be replaced with a microscopic simulation model, the estimation time would become even longer. The Kriging model thus has the advantage of high computational efficiency, which makes it suitable for heuristic optimisation that needs to calculate the fitness a large number of times.

	201	251	351	451	LKWM
Estimation Time $(\times \mathbf{10^{-4}s})$	1.1637	1.3875	1.5032	1.9432	430.17

Table 3.2 Average estimation time of Kriging models with 201, 251, 351 and 451 training points and LKWM

3.5 Conclusions

This chapter has proposed a conceptual framework of surrogate modelling for real time optimisation. Training points sampling, surrogate model construction, model evaluation and infill strategy are discussed in detail. The key features of the proposed framework are summarised as follows:

- In contrast to other problems that use uniform sampling across the entire design space, traffic demands are generated according to the feature distribution. This distribution can be obtained directly from historical data or other indirectly-related information. With this sampling principle, more points can be sampled around the region with a higher probability.

- The signal control variables are sampled around the benchmark, which is generated off-line. Since the ultimate goal of the surrogate model is to achieve real time signal control, the baseline signal helps to identify the area that is more likely to find the optimal solution, meaning that the number of sample points can be reduced.

- In contrast to surrogate models used for strategic design, models for real time optimisation need to construct the response surface between the traffic state, control variables and network performance. Surrogate models do not need any additional prediction model to estimate the traffic states; they merely use the traffic state at the end of the previous control period to approximate the relationship between the traffic states $u(t-1)$, signal control variables $s(t)$ and performance $z(t)$ directly.

- A new multi-point candidate-based infill strategy was proposed. In this strategy, for each

infill iteration, k infill points with the largest estimation error are selected from K candidate points. Additional constraints on the minimum distance between infill points are made to avoid clustering and ensure sample diversity. The surrogate model can then be re-trained with the new set of training points. In order to ensure the performance of the surrogate model in real time optimisation, global accuracy is also tested.

An empirical test was carried out on a test network. The experimental results of the test show that the proposed infill strategy is effective at improving the global accuracy of the Kriging model. The degree of improvement decreases as the number of training points increases, however. Due to the simplification of the surrogate model, the estimation error can never drop to zero.

The estimation errors of Kriging model eventually decline to a small value, which indicates that the proposed Kriging model can interpret the complex dynamics accurately. Due to sampling uncertainty, initial samples have a high influence on the Kriging model, but this influence decreases as more infill points are added. It can be reasonably concluded that the Kriging models can have a similar level of global accuracy after several iterations of infill, regardless of the arrangement of initial samples.

The proposed Kriging model shows its superiority in time-efficiency to the simulation model. It can be built in less than 20 s. All the parameters are tuned according to the training data automatically without additional manual calibration. It usually takes a much longer time to construct a simulation model and calibrate its parameters. In addition, the estimation time of Kriging model is much shorter than that of the simulation model, which makes Kriging model more suitable for real time optimisation as it needs to be recalled a large number of times. The following chapter introduces the real time signal control based on the surrogate model proposed in this chapter.

Chapter 4

Real time signal control with surrogate model

In order to demonstrate the feasibility and effectiveness of the surrogate modelling techniques proposed in Chapter 3 for real time signal control, this chapter develops a surrogate-based real time signal control framework. The only part of this real time signal control that is carried out on-line is the optimisation process, which is bounded by the red dashed line in Figure 4.1. A modified Particle Swarm Optimisation (PSO) is developed to generate reliable signal plans efficiently. The same network used in Chapter 3 is used again for numerical studies in this chapter to evaluate the performance of the proposed signal control on network management in terms of total throughput, average delay and weighted throughput.

Chapter 4 is structured as follows. Section 4.1 presents the optimisation framework of the problem based on the surrogate model proposed in Chapter 3. Section 4.2 discusses the standard algorithm of PSO. Section 4.3 proposes a modified PSO specially designed for surrogate-based real time signal control. A numerical test is conducted in Section 4.4 to evaluate the performance and efficiency of the signal control developed in Section 4.1. Finally, Section 4.5 conducts multi-scenario analyses to validate the robustness of the proposed signal control to different levels of state representation, network saturation, demand variability and influence of sensitivity to estimation error and baseline signal.

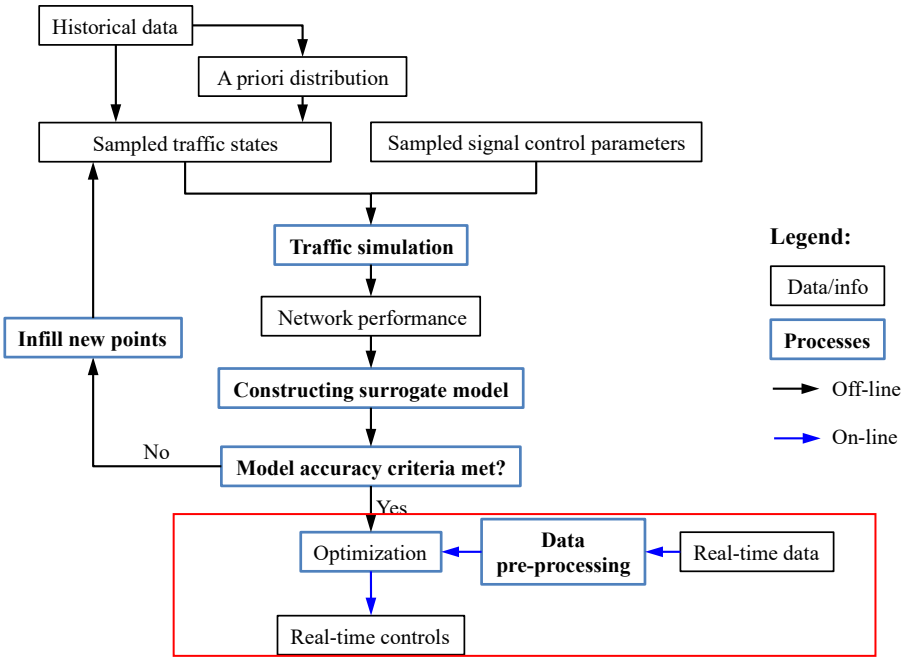


Figure 4.1 Framework of real time optimisation

4.1 Optimisation problem formulation

According to the definition given by Schneider & Kirkpatric (2006), '*optimisation*' is here defined as the approach to find a quite good (i.e. near-optimal solution), or optimal, solution to the proposed problem. Optimisation methods can be classified into two categories: exact and heuristic approaches. Compared to the exact approach, which always leads to a solution that can be proved mathematically to be optimal, the heuristic approach can only generate a near-optimal solution. The heuristic approach cannot prove the optimality of the solution and even does not know how far it is from the true optimum. In this thesis, the result obtained when the optimisation process finally converges is called the '*optimal solution*' or '*optimum*', even though, in fact, it might not be the true optimal solution of the problem. For the sake of distinction, the optimal solution that is obtained analytically and can be proved to be optimal is called the '*true optimal solution*' or '*true optimum*'.

When the real time traffic data (such as loop data from different sensors) are received, a pre-processing tool maps them to a given traffic state, s. The next step is to solve an optimisation problem, the preliminary form of which is given in Equation (4.1). The optimal

solution, u^*, is obtained by selecting the one with a minimum objective value under the given real time traffic state from a set of possible solutions. Here, for generality, the case of minimisation is considered.

$$u^* = \Theta(s) = \underset{u}{argmin}\{\widehat{\boldsymbol{\Psi}}(s,u)\} \tag{4.1}$$

The optimisation process of the real time signal control is slightly different from that of fixed-time signal control. For a fixed-time signal control, the traffic condition is known *a priori*, so the traffic states are not a part of the input variables of surrogate models. The optimisation process is usually integrated with the infill procedure (Osorio & Bierlaire, 2013), such that the infill points are selected on-the-fly to improve the accuracy while at the same time moving towards the global optimal. No conventional optimisation methods are involved in this kind of problem as there is no independent optimisation process. For real time signal control and optimisation, however, traffic states are an essential component of the input variables. Since the global optimum varies for different traffic states, on-line optimisation cannot be intertwined with infill as the traffic states change too frequently. As such, the on-line optimisation amounts to a regular nonlinear and non-convex optimisation problem. This can be solved easily by using a cheaply computable surrogate model in place of the simulation model.

To find a near-optimal solution within a reasonable amount of time, a heuristic method is employed for the problem for the following reason. Firstly, the heuristic approach only needs zeroth order information, thus it is suitable for this problem since the first and second order information of the objective function and constraints are unavailable. Secondly, the heuristic approach is problem independent, and hence can be applied to different problems flexibly and accommodate various constraints. Finally, the heuristic approach is chosen for its shorter converge time than the exact approach. Since the signal optimisation problem usually contains a large number of variables, it is not necessary, and even impossible, to find a 'true' optimal solution in the context of real time signal control optimisation. In addition, due to the stochasticity of the traffic network, the true optimal solution obtained by the optimisation process might not be the actual true optimum of the traffic system when it is implemented. In

short, it is not necessary to over-emphasise the optimality of the solution: for this particular problem, efficiency is more important than optimality. As a result, it is appropriate to use the heuristic approach for the real time signal control system.

4.2 Particle Swarm Optimisation

For a non-convex optimisation problem with a large searching space, population-based methods such as genetic algorithm (GA) and particle swarm optimisation (PSO) are more suitable. PSO (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995) is a population-based optimisation method, which emulates the social behaviour of animal swarms and searches for near-optimal solutions through swarm intelligence. By allowing each particle in the swarm to update their positions and velocities iteratively according to their own, and the group's, best positions, PSO does not require any gradient information and offers a flexible trade-off between optimality and solution efficiency (Banks et al., 2007). It starts at multiple points so as to increase the probability of finding the true optimum. Compared with GA, PSO only shares the best position within the swarm, therefore it converges faster than GA. In addition, it has few parameters to tune (Ercan, 2009). As a result, PSO is selected in this research, since it is simple, fast and easy to implement. The following section introduces the standard algorithm for PSO. Figure 4.2 shows the overall flowchart of the PSO process, which iterates until the stopping criteria are met. Each step of PSO is discussed in the following part.

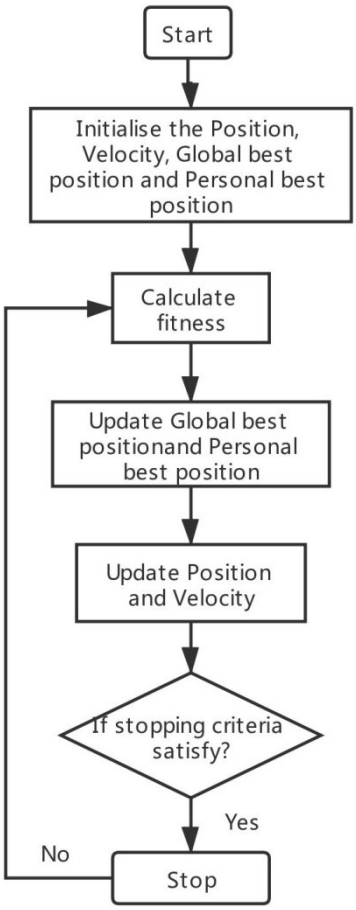


Figure 4.2 The flowchart of PSO

Notations of PSO equations:

X_i^k	Position of the *i*-th particle in iteration k
V_i^k	Velocity of the *i*-th particle in iteration k
$f(\cdot)$	Fitness function
ω	Inertia weight
C_1	Cognitive learning factor
C_2	Social learning factor
P_i	Personal best position of the *i*-th particle
P_g	Global best position
V_{max}	Maximum velocity

Given the notations above, the first step of PSO is to generate a swarm with m particles, of

which the initial positions and velocities are $\boldsymbol{X^0} = [X_1^0, X_2^0, \cdots, X_m^0]$ and $\boldsymbol{V^0} = [V_1^0, V_2^0, \cdots, V_m^0]$. The initial fitness value can then be calculated as $\boldsymbol{f^0} = [f(X_1^0), f(X_2^0), \cdots f(X_m^0)]$, where $f(\cdot)$ is the fitness function.

For each iteration, the position and velocity of each particle are updated with the following equations:

$$V_i^{k+1} = \omega V_i^k + C_1\gamma_1\left(P_i^k - X_i^k\right) + C_2\gamma_2\left(P_g^k - X_i^k\right) \tag{4.2}$$

$$V_i^k \in [-V_{max},\ V_{max}] \qquad \forall i, \qquad \forall k \tag{4.3}$$

where V_i^k and X_i^k are the current velocity and position of particle *i* in the *k*-th iteration, respectively. P_i^k is the personal best position of the *i*-the particle among all the positions it has visited within the previous *k* iterations. P_g^k is the global best position, which is the best one from all the personal best positions. C_1 and C_2 are the learning factors. The former is the cognitive learning factor, while the latter is the social learning factor. The relative value between C_1 and C_2 indicates the relative importance of the P_i^k and P_g^k on the update. γ_1 and γ_2 are both random values generated uniformly between $[0,\ 1]$. They are randomly generated every iteration to represent the stochasticity of the particle movement. Equation (4.2) is the velocity update equation, which has three terms. The first term represents the influence of the current velocity on the updated velocity, in which ω is the inertia weight. A large value of ω indicates the strong willingness of the particle to follow the current searching direction. In contrast, the particle with low ω tends to move towards the personal best position and global best position. The value of ω controls the exploration and exploitation of the searching. The second and the third parts represent the learning ability of the particle from its own experience and swarm experience. They represent the competition and cooperation between particles, respectively.

The velocity is limited within the range of $[-V_{max},\ V_{max}]$, where V_{max} is a constant that

denotes the maximum velocity at which the particles can move. The sign represents the direction of movement. V_{max} can also control the global and local searching capabilities. A large value of V_{max} allows a particle to move a long distance in each iteration, such that it has a good global searching ability. Although a high velocity means that a particle may escape easily from the local optimum, it also makes it easier for that particle to fly over and miss the global optimum. A small value of V_{max}, meanwhile, limits the area in which a particle can search, such that particles have high precision on searching but a high probability of being trapped at local optimum.

Once the velocity is updated, the position can be calculated with the equation:

$$X_i^{k+1} = X_i^k + V_i^k \tag{4.4}$$

With the new position, Equation (4.5) shows the update of the personal best position. The new personal best position P_i^{k+1} can be obtained by comparing the original personal best position P_i^k with the new position of the particle X_i^{k+1}. If the fitness of X_i^{k+1} is smaller than that of the P_i^k, X_i^{k+1} becomes the new personal best position, otherwise the original personal best is maintained. The update of the global best position follows a similar procedure, as shown in Equation (4.6). Algorithm 4.1 summarises the detailed procedures of PSO optimisation.

$$P_i^{k+1} = \begin{cases} P_i^k, & if\ f(P_i^k) \le f(X_i^{k+1}) \\ X_i^{k+1}, & if\ f(P_i^k) > f(X_i^{k+1}) \end{cases} \tag{4.5}$$

$$P_g^{k+1} = \begin{cases} P_g^k, & if\ f(P_g^k) \le \min\limits_{X_i^{k+1} \in \boldsymbol{X^{k+1}}} \left(f(X_i^{k+1})\right) \\ \underset{X_i^{k+1}}{argmin}\{f(X_i^{k+1})\}, & iff(P_g^k) > \min\limits_{X_i^{k+1} \in \boldsymbol{X^{k+1}}} \left(f(X_i^{k+1})\right) \end{cases} \tag{4.6}$$

Algorithm 4.1: Standard PSO Optimisation

Step 1 Set the parameter value of ω, C_1, C_2 and V_{max}. Define the number of particles.

Step 2 Initialise the position $\boldsymbol{X^0}$ and velocity $\boldsymbol{V^0}$ Let $k = 0$.

Step 3 Calculate the fitness value $\boldsymbol{f^k}$.

Step 4 Update the personal best position $\boldsymbol{P_i^{k+1}}$ and global pest position $\boldsymbol{P_g^{k+1}}$ with Equation (4.5) and (4.6).

Step 5 Update the position $\boldsymbol{X^{k+1}}$ and velocity $\boldsymbol{V^{k+1}}$ with Equation (4.2) and (4.4).

Step 6 Check if the stopping criterion is met. If yes, output $\boldsymbol{P_g^{k+1}}$ as the final solution, if not, go back to step 3 and $k = k + 1$.

4.3 Modified PSO for surrogate-based real time signal control

This section introduces the application of PSO in the surrogate-based real time signal control system. Several modifications are discussed in detail.

When PSO is applied to solve a real time signal optimisation problem, the particle represents the signal plan. For a green time optimisation problem, each dimension of the particle denotes the green time of a stage of a signalised intersection, as shown in Equation (4.7). The final target of PSO is to search for a signal plan good enough to minimise the specific objective function.

$$X_i = [g_1, g_2, \cdots g_l] \tag{4.7}$$

The problem of real time signal control has strict restrictions on computational time. With the rapid changes in traffic conditions mean that it is necessary to make the decision in a few seconds, such that the signal control can adapt to the current traffic condition, otherwise the traffic data collected will lose its representativeness. Even though PSO has advantages in its time efficiency, to further reduce the convergence time, two additional measures are taken.

The first measure is to sample the initial positions of the particles following a normal

distribution around the baseline signal plan, which is similar to the signal sampling process for surrogate model construction illustrated in Section 4.2.2. The baseline signal and the normal distribution of PSO need to be the same as for surrogate model construction. Due to the local sampling effect of the signal control variables, the sample density of the entire design space is not uniform. Using the same sampling approach for PSO as for the surrogate model construction ensures that the search for the optimal solution is concentrated in the region with sufficient samples and high accuracy. Moreover, initial particles of PSO are sampled in the region that is likely to find the optimal solution, which can shorten the convergence time.

The second measure is to use a dynamic inertia weight ω. The value of ω decreases in each iteration. In the beginning, the inertia weight is given a large value, such that particles tend to explore new regions rather than search the regions that have already been visited. The ability of local searching is improved with time, by gradually decreasing the value of the inertia weight to ensure that the searching is concentrated in the regions with better fitness values. In Equation (4.8), α is the parameter representing the decreasing rate of inertia weight, the lower limit $\omega_{\min}$ is set to limit the minimum value of ω.

$$\omega^{k+1} = \alpha\omega^{k} \tag{4.8}$$

Although these two measures can reduce the convergence time, they may also lead to the premature converge of PSO. The first measure reduces the diversity of particles, and the second one emphasises the local search. This, however, is a trade-off between optimality and efficiency. In the case of real time signal control, efficiency overweighs the optimality.

It is worth noting that PSO can only find the optimal solution of the surrogate model rather than that of the underlying problem. How close the optimal solutions of the surrogate model and the underlying problem are dependent on the accuracy of the surrogate model's interpretation of the underlying problem. The inevitable estimation error of the surrogate model thus propagates to the optimisation stage, causing the derived solution to be

sub-optimal or even un-desirable. Furthermore, in the case of surrogate-based optimisation, the reliability (i.e. estimation error) of the response surface also needs to be considered. The optimal solution is not the one with the best objectives only; it should also have a low estimation error. The revised fitness function is shown in Equation (4.9), where $\widehat{\boldsymbol{\Psi}}$ and σ are the estimated output and estimation error, respectively. Parameter A is the weighting parameter that balances optimality and reliability. It represents the sensitivity to the estimation error. When A decreases to zero, the optimal solution is selected purely based purely on the optimality.

$$f(x) = \widehat{\Psi}(x) + A \times \sigma(x) \tag{4.9}$$

The stopping criteria are that the maximum number of iterations has been reached or the relative improvement of fitness value is less than a pre-defined threshold for several successive numbers of iterations, as shown in Equation (4.10).

$$\frac{f\left(P_g^k\right) - f\left(P_g^{k+1}\right)}{f\left(P_g^k\right)} \leq \varepsilon \tag{4.10}$$

Algorithm 4.2 summarises the detailed procedures of the modified PSO designed for surrogate-based real time signal control.

Algorithm 4.2: PSO for surrogate-based signal control

Step 1 Set the parameter value of $\alpha, A\ C_1,\ C_2$ and V_{max}. Define the number of particles

Step 2 Initialise the position $\boldsymbol{X^0}$, velocity $\boldsymbol{V^0}$ and inertia weight w^0. Let $k = 0$.

Step 3 Project $\boldsymbol{X^k}$ to the feasible region Λ.

Step 4 Calculate the estimated output $\widehat{\boldsymbol{\Psi}}$ with the surrogate model and estimation error σ.

Step 5 Calculate the fitness value $\boldsymbol{f^k}$ with Equation (4.8).

Step 6 Update inertia weight w^{k+1} with Equation (4.7).

Step 7 Update the position $\boldsymbol{X^{k+1}}$ and velocity $\boldsymbol{V^{k+1}}$ with equation (4.2) and (4.4).

Step 8 Project $\boldsymbol{X^{k+1}}$ to the feasible region Λ

Step 9 Check if the stopping criterion is met. If yes, output P_g^{k+1} as final solution, if not, go back to step 3 and $k = k + 1$.

4.4 Simulation test

In this part, a simulation test is conducted on the same network as used in Chapter 3. A Kriging model is randomly selected from the 50 models constructed in Chapter 3 and used for fitness estimation. In Section 4.4.1, the experimental setup is introduced. Section 4.4.2 reviews the assumptions made to this simulation test. Section 4.4.3 assesses the effectiveness of the proposed real time signal control on the management of total throughput, average delay and weighted throughput and optimisation efficiency.

4.4.1 Set up of the experiment

Figure 4.3 shows the optimisation procedures of the real time signal based on surrogate modelling. In a given control period, with the known signal control and demands, the traffic states (i.e. RO) at the end of the control period can be estimated by the simulation model. Based on the estimated ROs, the modified PSO approach is used to generate the optimal signal control for the next control period. In this experiment, the optimisation via the modified PSO approach can be expressed as Equation (4.10), where $\widehat{\boldsymbol{\Psi}}$ is the estimated total throughput of the whole control period and σ is the estimation error, under the condition that traffic state is $s(p)$ and signal decision is u. When the weighting parameter, A, decreases to zero, Equation (4.11) is transformed into Equation (4.1). The estimation error can be easily estimated with the Kriging model. These procedures are repeated until the end of the experiment.

$$u^*(p+1) = \Theta(s(p)) = \underset{u}{argmin}\{\widehat{\boldsymbol{\Psi}}(s(p),u) + A \times \sigma(s(p),u)\} \tag{4.11}$$

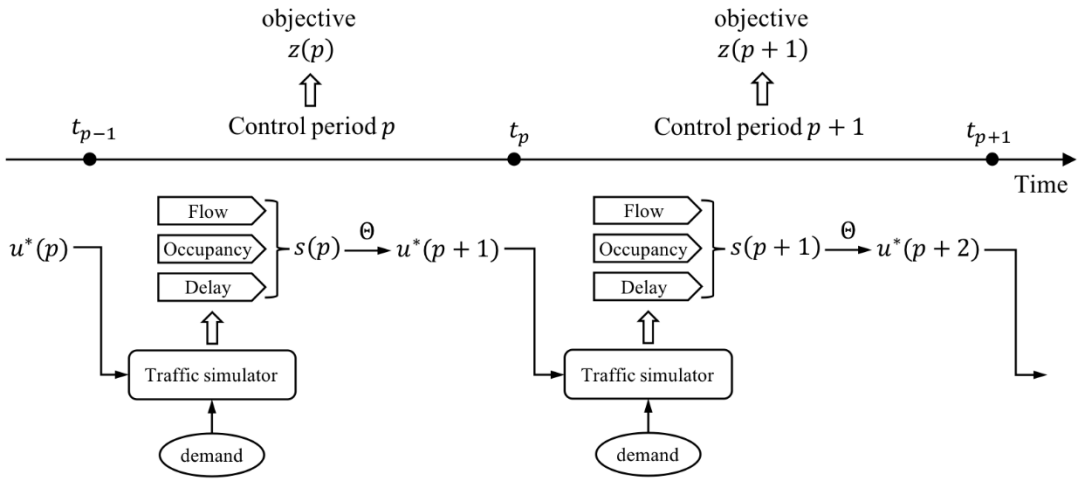


Figure 4.3 Real time signal control approach based on surrogate modelling

In this experiment, the simulation setup consists of 50 replications of the morning peak hour (8-9 am) on a typical working day, following a 400 s warming-up session. The one-hour session is divided into twelve control periods, each of which lasts for five minutes. The signal plan remains unchanged throughout the control period until a new control period begins. The traffic state is the average RO of the previous 20 time steps of eleven entrance links and eight main links to account for the temporal variation due to the change of signals, which is the same as the one used in Section 3.4.

In order to evaluate the performance of the traffic network over the entire analysis period T, the following three key performance indicators (KPIs) are considered:

$$Network\ throughput = \sum_{t\in T}\sum_{i\in O} f_i^{out}(t) \times dt \tag{4.12}$$

$$Weighted\ throughput = \sum_{t\in T}\frac{1}{t+1}\sum_{i\in O} f_i^{out}(t) \times dt \tag{4.13}$$

$$Average\ delay = \frac{\sum_{t\in T}\sum_i \left(N_i^{up}(t) - N_i^{down}(t)\right) \times dt}{\frac{1}{2}\left(\sum_{i\in I} N_i^{up}(T) + \sum_{i\in O} N_i^{down}(T)\right)} \tag{4.14}$$

where I denotes the set of entrance links, and $N_i^{up}(T)$ and $N_i^{dn}(T)$ are, respectively, the cumulative entering and exiting vehicle counts at time step T of link i. The weighted throughput assigns higher weights to vehicles leaving the network at an earlier time, and therefore indirectly encourages lower delays (Han & Gayah, 2015). In the expression of average delay (Equation (4.14)), the numerator denotes total link travel times (in veh·s), which is the integral of the subtraction between cumulative entering vehicles and cumulative exiting vehicles. The shaded area in Figure 4.4, bounded by the N_i^{up} and N_i^{down} curves, represents the total travel time of all the vehicles that have been in the link. The denominator denotes the average number of trips (in veh). It is the average of the cumulative entering and exiting vehicles counts of the link. If only exiting vehicles are considered, the average delay will be overestimated, as the vehicles remaining in the link also contribute to the total travel time. On the other hand, the average delay will be underestimated if all the vehicles that have entered the link are considered, as some of them have not finished their trips. Therefore, the average of N_i^{up} and N_i^{down} is used to estimate the average delay.

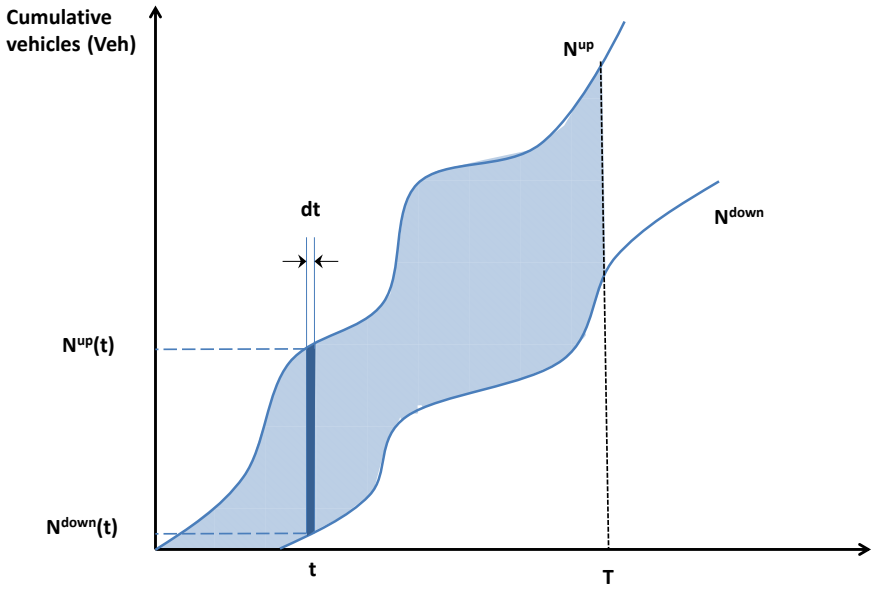


Figure 4.4 Demonstration of total travel time

TThe KPIs in Equations (4.13) – (4.14) are used to evaluate the network performance and compare the effectiveness of different signal control strategies. Note that they are not used directly as the objective of the surrogate-based optimisation, for the following reasons. The

weighted throughput, with time index in its denominator, is affected by the total number of time steps in the control period. If the length of the control period is relatively long, or the time interval is too short, the huge number of time steps will diminish the contribution of the last few time steps to the overall objectives. Then, in the latter stage of the control period, traffic control would not be able to manage the traffic networks effectively, potentially leading to undesirable local congestion within the network. The average travel time in Equation (4.14) cannot be easily estimated based on real-world measurements, limiting the model-free capabilities of surrogates. Moreover, Girianna & Benekohal (2004) pointed out that, under oversaturated conditions, queue formation and discharge should be set as the prime objective instead of delay minimisation, meaning that average delay is not chosen as the objective either.

In this experiment, the proposed real time signal control is compared with three benchmarks:

- **Scenario 1:** The fixed-time signal control based on the real-world test network in Glasgow, as illustrated in Liu et al. (2015), referred to here as *'GCC'*.
- **Scenario 2:** The fixed-time signal control where each intersection is controlled independently following the equi-saturation policy (Webster, 1958), referred to here as '*Webster*'.
- **Scenario 3:** The surrogate-based real time signal control with Genetic Algorithm as the optimiser, referred to here as *'GA'*.

The optimisation time of the Kriging-based signal control and simulation-based signal control for one control period is shown in order to verify the time efficiency of the Kriging-based signal control.

4.4.2 Simulation environment

The primary purposes of the simulation model are to simulate the traffic dynamics of the traffic network and evaluate the performance of the signal plan. In this experiment, LKWM is employed to evaluate signal control decisions. This is the same model that was used in

Section 3.4.3.1 to capture traffic dynamics for the construction of the surrogate model. The choice of LKWM as the simulation platform avoids introducing unnecessary uncertainties that are not related to the design of the signal control framework. Suppose a distinct simulation model rather than LKWM were to be used as the simulation environment for optimisation. In that case, the discrepancy between the two simulation models would add an extra source of uncertainty and it would be difficult to distinguish whether the performance differences between the designed signal control systems and benchmarks were due to the design of the algorithm or to the poor approximation of the surrogate model to the actual traffic dynamics.

Since the objective of this signal control system is to manage the network-level performance (i.e. total throughput), it is sufficient to use an LKWM that simulates the traffic dynamics at a macro level. If necessary, however, LKWM can be replaced by a model with a finer resolution that relaxes the assumption of homogeneous traffic flow. Replacing the simulation platform with a more refined model will not significantly change the findings, as long as the simulation models for both interpretation of the traffic dynamics and evaluation of signal control decisions are well calibrated with the same set of data, to ensure that the discrepancy between them is low enough to be neglected.

4.4.3 General assumptions

Besides the assumptions made in Section 3.4.3.1, the following assumptions are made for the empirical test, such that the key features to investigate can be highlighted, while some less critical features are neglected for simplicity.

1. Assume that only the stage green time is optimised.

For cyclic signal control, the signal plan is defined by the following concepts:

- **Cycle time**: the length of time for a complete signal plan
- **Stage green time**: the length of the green time of each stage
- **Offset**: the time difference between the start of two stages of adjacent intersections

- **Phase sequence**: the sequence of the stages

In this experiment, only the optimisation of stage green time is considered. All the other signal timing variables are assumed to be fixed. If all the signal time variables were optimised at the same time, the increase in the number of variables would increase the dimensions of the design space. In that case, real time optimisation becomes unavailable. The optimisations of other parameters can be achieved by using different optimisers running in parallel and realised following the same procedure. The optimisation of other signal timing variables will be discussed in a future study.

2. Assume that the traffic data can be collected and used directly.

In reality, the installation of detectors is restricted by practical constraints such as network geometry. Because of this, these constraints need to be taken into account when selecting input variables. In this research, due to the lack of information, it is assumed that the traffic data required can be collected directly, such that the selection of input variables is completely based on the analytical analysis. In practice, the relative occupancy can be estimated by installing detectors at the entrance and exit of the link. If an input variable cannot be collected directly by the sensors, then its importance to the approximation accuracy of the surrogate model needs to be evaluated. If it is not crucial, it can be removed from the set of input variables, or it needs to be estimated from other sources of information. The multi-scenario analysis in Section 4.5.1 shows that there is no need to install detectors at all the links.

In addition, it is assumed that the data collected are of high quality and can be used for optimisation directly. In a real-world application, the traffic data collected by detectors may contain faults. Since low-quality traffic data can affect the reliability of the signal control system, a data pre-processing system needs to be used to identify and correct faulty data before they are fed into the signal control system. Abnormal data can be identified and removed based on traffic flow characteristics. All missing data can be estimated based on the temporal and spatial characteristics of traffic data (Li et al., 2018). A large amount of missing or abnormal data suggests that the functionality of the sensors should be checked. Under this

condition, in order to avoid the adverse effects arising from poor quality data, all the data would need to be removed and a backup signal plan recalled until the problem has been solved.

3. Assume that updates to the signal plan happen instantaneously.

The assumption that signal updates happen instantaneously allows the switch to a new control period and the update of the signal plan to occur simultaneously. Under this condition, it is not necessary to conduct the optimisation before the end of the control period, as with SCOOT, to account for the optimisation time. In fact, the optimisation time is much shorter than the minimum green time. With the fixed phase sequence, the assumption is valid, since the new signal plan can be generated before the end of the first stage.

4. Assume that the signal control system is operated in normal conditions.

In reality, system failure is a significant threat to the real time signal control system, especially for centralised signal control. During the practical implementation, a back-up mode can be set and activated when a system failure is detected. In this experiment, however, it is assumed that the signal control system always works normally.

4.4.4 Input parameters and stopping criteria of PSO

The input variable of PSO is green split. Green split is the proportion of stage green time to the cycle time. The sum of the green splits of all the stages at each intersection is one. Due to the non-uniform approximation accuracy of the searching space, the maximum velocity is given with a low value in order to avoid the result deviation caused by the particles flying to the region with low accuracy. The large number of swarm particles guarantees the global searching ability.

Value of PSO Parameters

$\omega_0 = 0.8$	Initial inertia weight
$\alpha = 0.97$	Decreasing rate of inertia weight

$\omega_{min} = 0.3$	Minimum inertia weight
$A = 1$	Weighting parameter
$C_1 = 0.618$	Cognitive learning factor
$C_2 = 0.618$	Social learning factor
$m = 140$	Number of swarm particles
$V_{max} = 0.01$	Maximum velocity
$\varepsilon = 0$	Threshold of relative improvement

Stopping Criteria

- Maximum number of 20 iterations has been reached.
- The relative improvement of fitness is lower than the threshold for 15 successive iterations.

4.4.5 Result analysis

In this test, due to its ease of reference, the notion *'GCC'* (Glasgow City Council) is used to represent the fixed-time baseline signal. *'GCC'* is involved in the initial sampling of PSO to accelerate the convergence. The four Kriging models, trained by 201, 251, 351 and 451 training points respectively, are used for signal optimisation. In the following sections, Kriging-n represents the Kriging-based signal control with *n* training points, where n=201, 251, 351 and 451.

Table 4.1 presents the average performance of the four Kriging-based signal controls, *'GCC'* and *'Webster'* over 50 replications in terms of total throughput, average delay and weighted throughput. It can be seen that since *'Webster'* only generates a fixed-time signal plan for each individual intersection independently, the lack of coordination makes it cannot perform as well as *'GCC'*. Moreover, the results show that Kriging-based real time signal controls always perform better than the two foxed-time signal controls as they can adapt to the variation of the traffic condition. As shown in Table 4.2, the average relative improvements indicate that the improvement over *'GCC'* increases with the number of training points, due to the improvement in approximation accuracy. Taking Kriging-451 as an example, the total

throughput increases by 5.3%; at the same time, the average time is reduced by 8.1%. Since *'GCC'* always performs better than *'Webster'*, only *'GCC'* will be used as the benchmark in the following analysis.

	Total Throughput	Average Delay	Weighted Throughput
Kriging-201	3436.87 Veh	948.81s	5.8984
Kriging-251	3481.91 Veh	927.77s	5.9273
Kriging-351	3506.36 Veh	917.35s	5.9415
Kriging-451	3529.27 Veh	904.89s	5.9641
GCC	3351.36 Veh	984.21s	5.8377
Webster	3057.69 Veh	1066.88s	5.7497

Table 4.1 Average performance of *'GCC'*, *'Webster'* and Kriging-based real time signal controls with 201, 251, 351 and 451 training points (over 50 independent tests)

	Total Throughput	Average Delay	Weighted Throughput
Kriging-201	2.5470%	3.6026%	1.3099%
Kriging-251	3.8949%	5.7557%	1.5348%
Kriging-351	4.6278%	6.7087%	1.7772%
Kriging-451	5.3098%	8.0669%	2.1642%

Table 4.2 Average relative improvement of Kriging-based real time signal controls with 201, 251, 351 and 451 training points over *'GCC'* (over 50 independent tests)

In order to visualise the comparison between Kriging-based signal controls and *'GCC'* more clearly, the CDF plot of the relative improvement of Kriging-based real time signal controls on a total throughput of 50 replications over *'GCC'* is shown in Figure 4.5. All four Kriging-based signal controls have better distributions of total throughput compared with the

baseline signal *'GCC'*. As the number of training points increases, there are a higher proportion of replications for which the Kriging-based control yields higher total throughput compared with *'GCC'*. The performance of Kriging-351 and Kriging-451, in particular, is consistently better than that of *'GCC'*. Moreover, the curve becomes steeper as the Kriging model is trained by more points, which indicates that the variance of improvement also decreases.

Figure 4.6 displays a CDF plot of the relative improvement on average delay (left) and weighted throughput (right) of the same signal controls used to produce Figure 4.5. It can be seen that the Kriging-based controls have better traffic management ability in terms of average delay and weighted throughput. The performance tends to stabilise, however, since the performance gap between two adjacent curves reduces with the increase in training points.

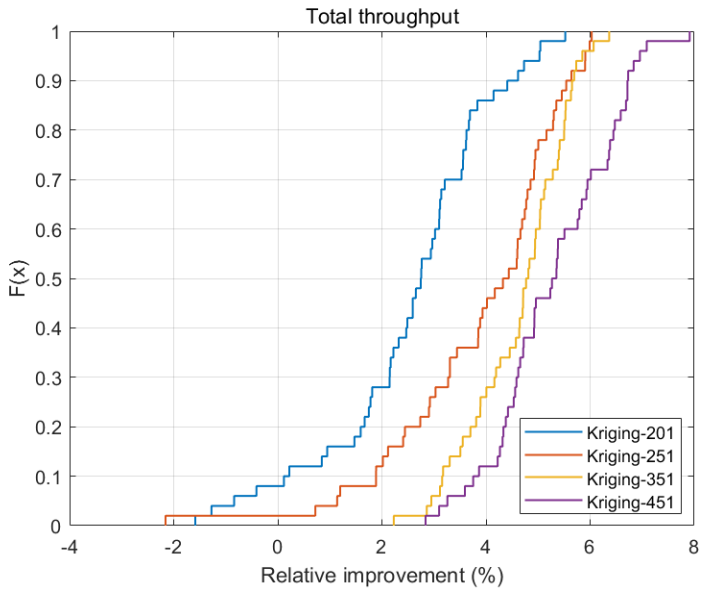


Figure 4.5 Empirical CDF plot of average relative improvement in total throughput

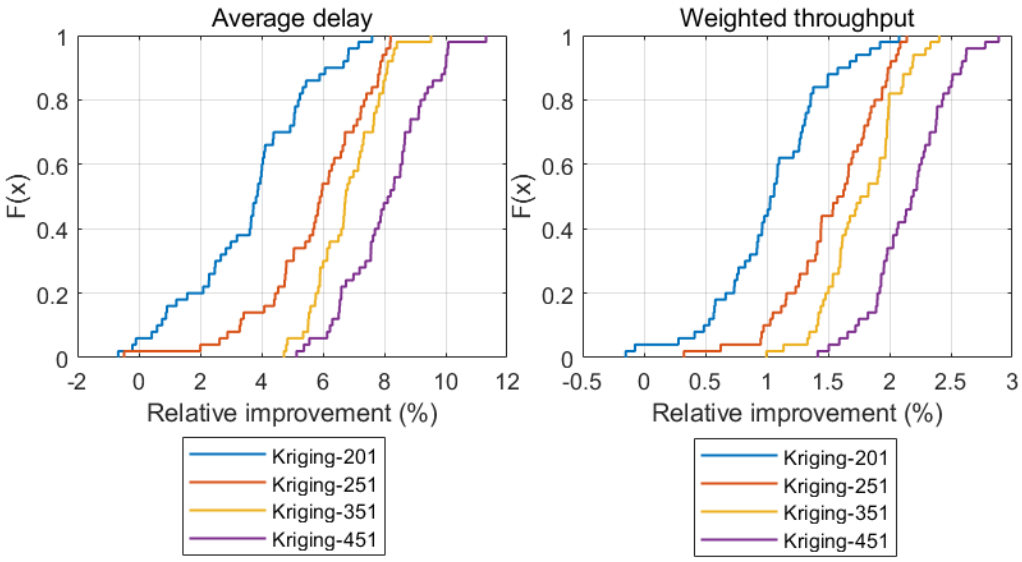


Figure 4.6 Empirical CDF plot of average relative improvement in average delay (left) and weighted throughput (right)

In order to prove whether the improvements over *'GCC'* are statistically significant, a one-tail paired t-test is performed. This can show whether the average performance of the Kriging-based signal control is statistically better than that of *'GCC'*. For total throughput and weighted throughput, a Left-tail t-test is adopted, while for average delay, a Right-tail t-test is adopted.

The null hypothesis and alternative hypothesis of Right-tail t-test are:

$$H_0: \mu_{GCC} \leq \mu_{Kriging} \tag{4.15}$$

$$H_1: \mu_{GCC} > \mu_{Kriging} \tag{4.16}$$

The null hypothesis and alternative hypothesis of Left-tail t-test are:

$$H_0: \mu_{GCC} \geq \mu_{Kriging} \tag{4.17}$$

$$H_1: \mu_{GCC} < \mu_{Kriging} \tag{4.18}$$

A 95% confidence interval is set here, such that if the p-value is lower than 0.05, the alternative hypothesis is accepted, which means Kriging-based signal control has better performance than *'GCC'*. The results in Table 4.3 show that the Kriging-based signal controls always have better performance than *'GCC'*, regardless of the number of training points.

		Total Throughput	**Average Delay**	**Weighted Throughput**
GCC & Kriging-201	p-value	1.0563×10^{-15} Better	3.4719×10^{-17} Better	1.8421×10^{-21} Better
GCC & Kriging-251	p-value	2.1466×10^{-22} Better	4.3480×10^{-28} Better	1.6246×10^{-31} Better
GCC & Kriging-351	p-value	1.2137×10^{-36} Better	7.4305×10^{-42} Better	2.1480×10^{-39} Better
GCC & Kriging-451	p-value	2.5443×10^{-35} Better	5.5823×10^{-40} Better	1.9431×10^{-42} Better

Table 4.3 T-test results between *'GCC'* and Kriging-based signal controls trained by 201, 251, 351 and 451 training points

During the optimisation, once the optimal solution is obtained, the total throughput throughout the control period is estimated by the corresponding Kriging model, and compared with the actual performance given by the simulation model. Table 4.4 presents the MAPE between the estimated and actual performance. This is averaged across the twelve solutions obtained in each replication, and across 50 replications. As expected, MAPE decreases as more points are added to the training set.

Figure 4.7 shows the CDF plots of the relative error between the estimated and actual performance, with a positive value indicating underestimation and a negative value indicating overestimation. It can be seen that, as the number of training points grows, the CDF curve becomes steeper and moves along the positive direction of the x-axis, which shows that the variance of relative error decreases and the proportion of underestimation increases. For a throughput maximisation problem, underestimation is preferred to overestimation, as it might lead to a point closer to the true optimal point. Both the MAPE and distribution of relative error show that approximation accuracy is the main factor affecting the performance of the Kriging-based signal control.

	Kriging-201	**Kriging-251**	**Kriging-351**	**Kriging-451**
MAPE (%)	8.0517	4.9238	3.7755	2.7427

Table 4.4 MAPE of Kriging models with 201, 251, 351 and 451 training points (over 50 independent tests).

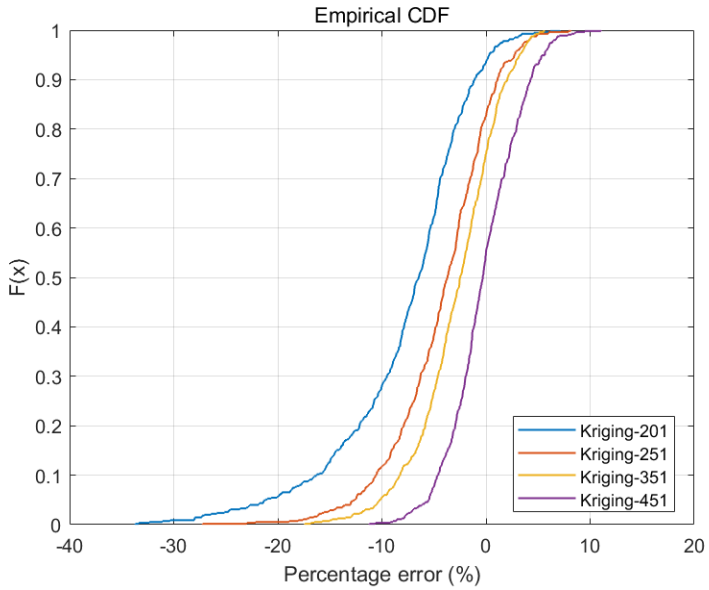


Figure 4.7 Empirical CDF plot of relative error of Kriging models with 201, 251, 351 and 451 training points (positive represents overestimation and negative represents underestimation)

In order to visualise the propagation of congestion with time, the total queue length is used as the indicator of network congestion. This is defined as the total number of vehicles queueing outside the network. The average queue length across 50 replications against time is plotted in

Figure 4.8. Two signal controls are considered: *'GCC'* and Kriging-451. The figure shows that Kriging-451 can relieve the congestion by allowing queues to dissipate faster compared with *'GCC'*. At the end of the simulation, Kriging-451 reduces the total queue length by more than 100 vehicles.

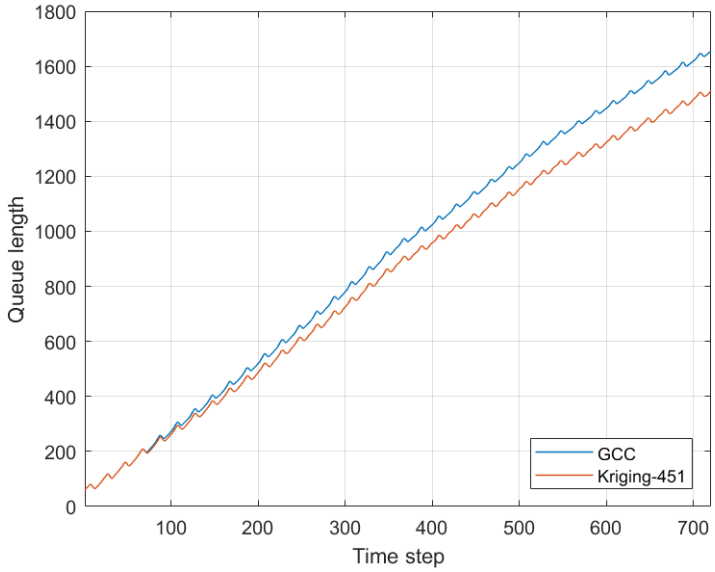


Figure 4.8 Total queue length of the network versus time

Table 4.5 shows the comparison between the performance of signal controls using PSO and GA as optimisers. Both of the two scenarios use the Kriging model trained with 451 points to estimate network performance of alternative plans. The results show that the one with PSO optimiser is slightly better than the one with GA optimizer.

	Total Throughput	**Average Delay**	**Weighted Throughput**
Kriging-451	3529.27 Veh	904.89s	5.9641
GA	3487.64Veh	930.53s	5.9202

Table 4.5 Average performance of 'Kriging--451' and *'GA'* (over 50 independent tests)

Table 4.6 shows the average optimisation time of the four Kriging-based signal controls. They are compared with *'GA'*, which represents the GA optimiser based on the Kriging model with 451 training points and the simulation-based optimisation, which is denoted as '*LKWM*', since the LKWM model is employed to simulate the traffic dynamics. The results show that

the application of the Kriging model dramatically reduces the optimisation time. The simulation-based optimisation spends more than 200 times longer than the Kriging-based optimisation. Moreover, from the computational efficiency perspective, *'GA'* almost doubles the time taken by the proposed PSO-based optimiser to generate a signal plan for the entire network.

The Kriging-based approach, however, exhibits an increasing trend in optimisation time with the increase in training points. All four cases have the same number of average iterations, since PSO terminates by satisfying the first stopping criteria. Thus, estimation time is the only factor affecting the optimisation time, which increases with the addition of more points, as discussed in Section 3.4.5.

	Kriging-201	**Kriging-251**	**Kriging-351**	**Kriging-451**	**LKWM**	**GA**
Optimisation Time (s)	6.9406	7.4183	7.7214	8.1077	2133.24	13.9343

Table 4.6 Average optimisation time of the Kriging-based PSO optimiser with a different number of training points, against *'LKWM'* and *'GA'*

4.5 Multi-scenario analysis

In this section, some important parameters are varied so as to draw additional insights into the proposed Kriging-based real time signal control. By employing Kriging-451, multi-scenario analyses are conducted on the dimension of the state space (4.5.1), demand level (4.5.2), demand variability (4.5.3), weighting parameter A (4.5.4) and baseline signal (4.5.5). This multi-scenario analysis allows an assessment of the robustness of the Kriging-based real time signal control to different traffic conditions and levels of knowledge. Moreover, the results can provide a reference for parameter design.

4.5.1 Dimension of the state space

For the construction of the model, the selection of input variables is critical to the

performance of the Kriging-based signal controls. One needs to capture network information essential for decision making, while eliminating insignificant and redundant information as much as possible to avoid unnecessarily high dimensionality. It is challenging to balance the dimensionality and approximation accuracy. The former influences the computational efficiency in both off-line training and real time decision making, while the latter affects the optimality of the real time controls. While increasing the number of input variables can contribute to the higher approximation accuracy of the surrogate model, since the features of the underlying network can be captured in more detail, this in turn increases the computational burden.

Using RO as the traffic state variable, four cases with a varying number of links are considered:

- **Case 1**: 25 links, which contains all the links in the network except outgoing links (as these outgoing links are not controllable)
- **Case 2**: 19 links, which are obtained from (1) by removing the six minor turning links at Byres/Great Western, hence leaving only the eleven inflow links and eight major arterials
- **Case 3**: 16 links, which are obtained from (2) by removing the three links with the smallest θ value
- **Case 4**: 13 links, which are obtained from (3) by further removing three links with the smallest θ value

The input variables can be selected according to the geometry and traffic dynamics of the network. This is the concept on which Case 2 is based. Owing to the Kriging model, the parameter θ of each dimension (see Equation (2.14)) can indicate the importance of each variable to the model output. According to this concept, some of the variables with low importance can be removed from the input set, as in Case 3 and Case 4.

Figure 4.9 illustrates the four cases, with the links used to extract state variables highlighted in red.

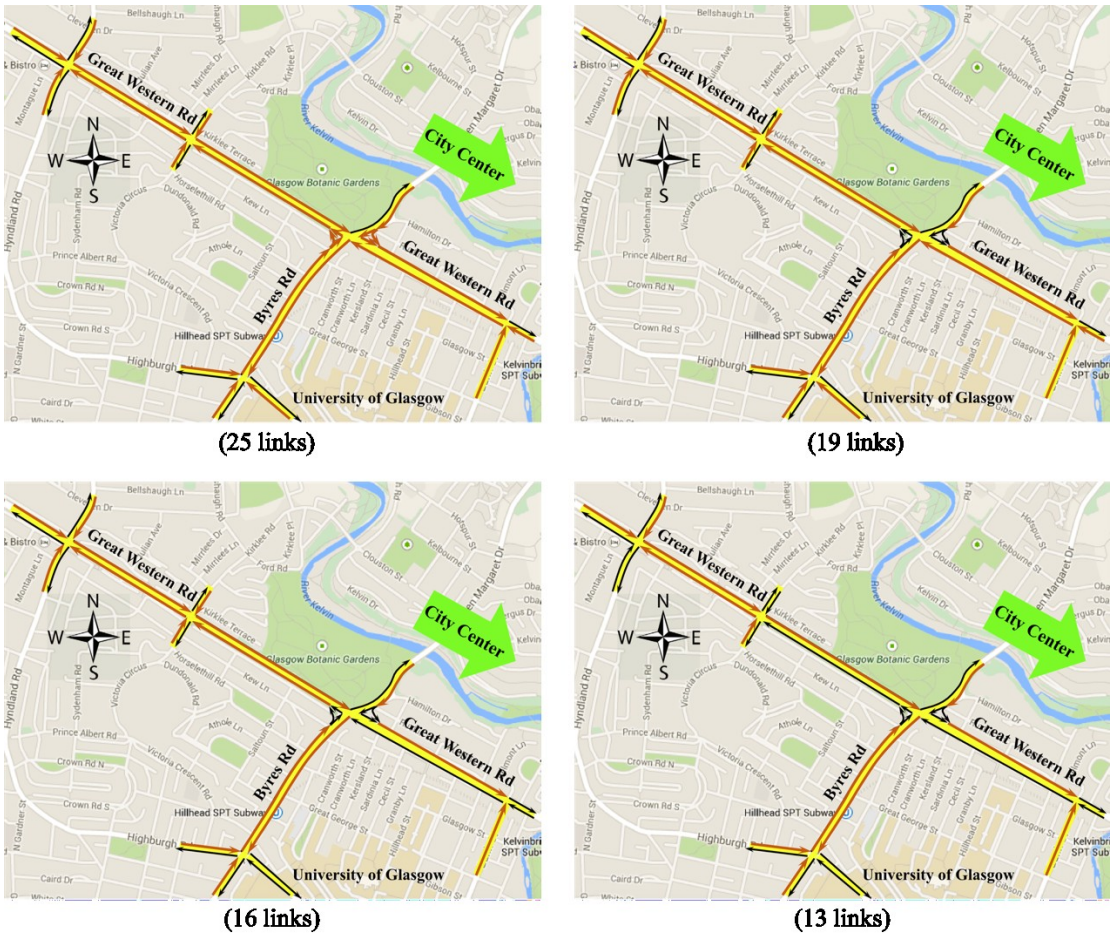


Figure 4.9 Links whose relative occupancy are used to comprise the state vector (highlighted in red) (Google Maps)

Figure 4.10 shows the variation of average estimation error, MAPE (%), versus the number of training points. For each scenario, 50 Kriging models are constructed and each tested with 11000 testing points, with MAPE averaged across these. For the cases with 13, 16, 19 variables, the approximation accuracy increases as the number of variables increases. This can be explained by the fact that when the number of variables decreases, traffic dynamics cannot be captured well, leading to larger estimation errors.

Interestingly, the case with 25 variables is not the one with the highest approximation accuracy. Even though, with the increase of training points, Case 1 (25 variables) surpasses Case 4 (13 variables) and Case 2 (16 variables) in terms of approximation accuracy, it does not outperform Case 2 (19 variables) when the number of training points reaches 601, although the gap between them is decreasing. This suggests that, in order to achieve the same

level of accuracy, the high-dimensional surrogate model needs more training points than the low-dimensional surrogate model. This is the reason why Case 1 (25 variables) does not perform well when the number of training points is low. The high-dimensional surrogate model will have higher accuracy when the number of training points is large enough, however. Thus, Case 1 (25 variables) has lower MAPE than Case 3 (16 variables) and Case 4 (13 variables) when they are trained by 601 points.

Comparing Case 2 (19 variables) with Case 1 (25 variables), six minor turning links on the Byres Road/Great Western Road are removed. These links are insignificant due to their size and high correlation with their upstream and downstream links, meaning that they can be well represented by the traffic states of the connected links. The low importance of these six links gives Case 2 (19 variables) a higher approximation accuracy than Case 1 (25), even when the number of training points reaches 601.

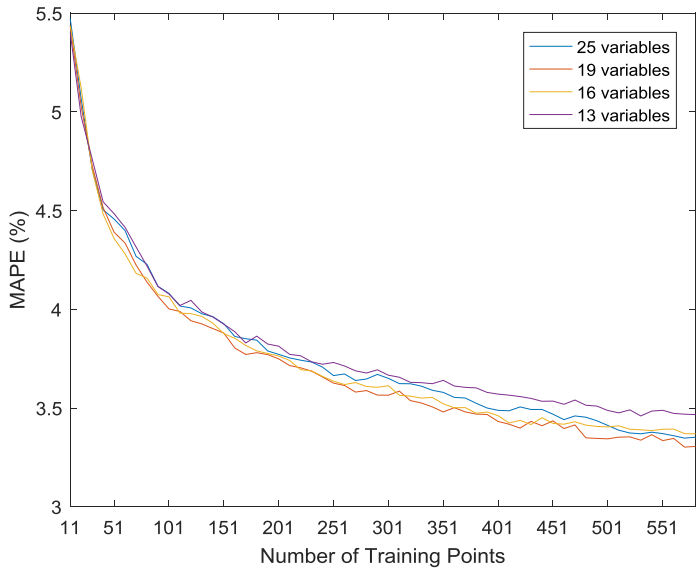


Figure 4.10 Average MAPE of the Kriging models with varying state spaces (# of links used as state variables) (based on 50 independent tests)

For each scenario, a Kriging model with 451 training points is used for real time signal control. The average performance in terms of total throughput, average delay and weighted throughput over 50 replications is presented in Table 4.7. Table 4.8 shows the relative improvement of the Kriging-based signal control over *'GCC'*. As shown in the tables, the

average performance and relative improvement follow the same trend as with approximation accuracy, in that Case 2 (19 variables) and Case 3 (16 variables) provide the best performance.

	Total Throughput	Average Delay	Weighted Throughput
25 variables	3480.40 Veh	926.25s	5.9313
19 variables	3529.27 Veh	904.89s	5.9641
16 variables	3528.00 Veh	907.54s	5.9577
13 variables	3505.80 Veh	921.67s	5.9322
GCC	3351.36 Veh	984.21s	5.8377

Table 4.7 Average performance of the Kriging-based signal controls with varying state variables (over 50 independent tests), compared to *'GCC'*

	Total Throughput	Average Delay	Weighted Throughput
25 variables	3.8506%	5.8913%	1.6024%
19 variables	5.3098%	8.0669%	2.1642%
16 variables	5.2713%	7.7971%	2.0548%
13 variables	4.6081%	6.3685%	1.6175%

Table 4.8 Average relative improvement of the Kriging-based signal controls with varying state variables over *'GCC'* (based on 50 independent tests)

Figure 4.11, Figure 4.12 and Figure 4.13 display the CDF of the relative improvement in total throughput, average delay and weighted throughput over *'GCC'*. It can be seen that even though Case 2 (19 variables) and Case 3 (16 variables) have similar average performance, the performance of Case 3 (16 variables) is more stable than that of Case 2 (19 variables) since its curve is steeper.

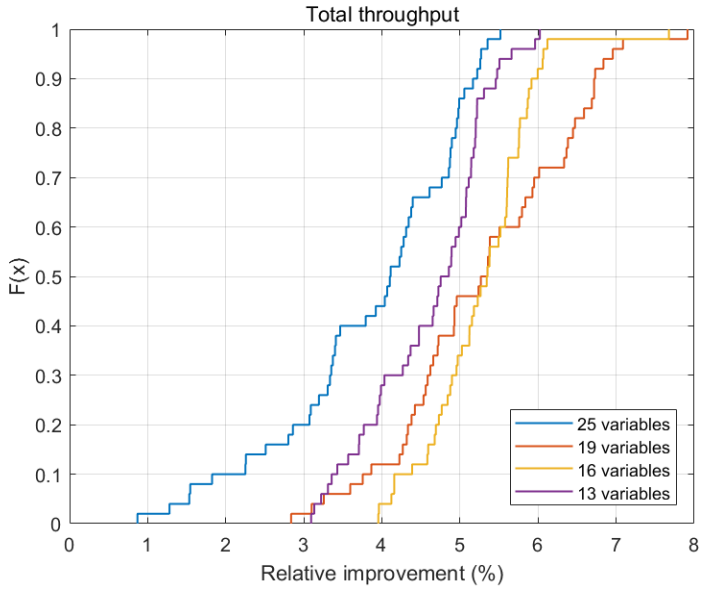


Figure 4.11 Empirical CDF plot of relative improvement in total throughput with varying state spaces (# of links used as state variables) (based on 50 independent tests)

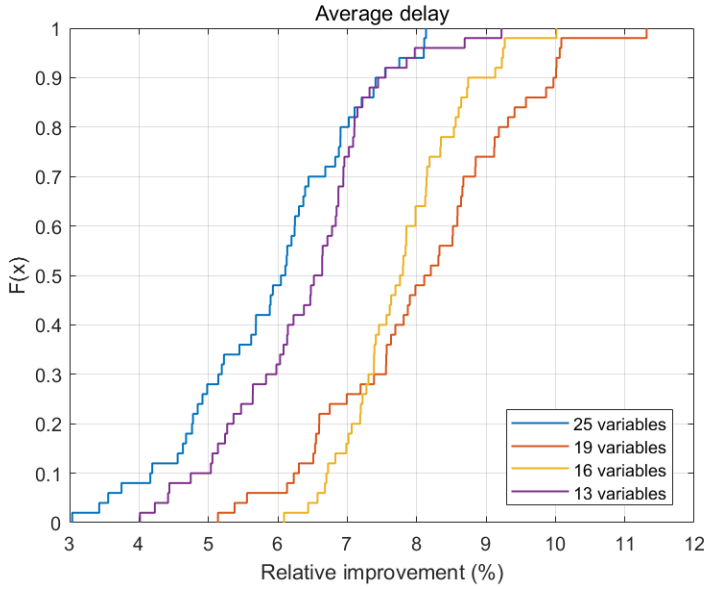


Figure 4.12 Empirical CDF plot of relative improvement in average delay with varying state spaces (# of links used as state variables) (based on 50 independent tests)

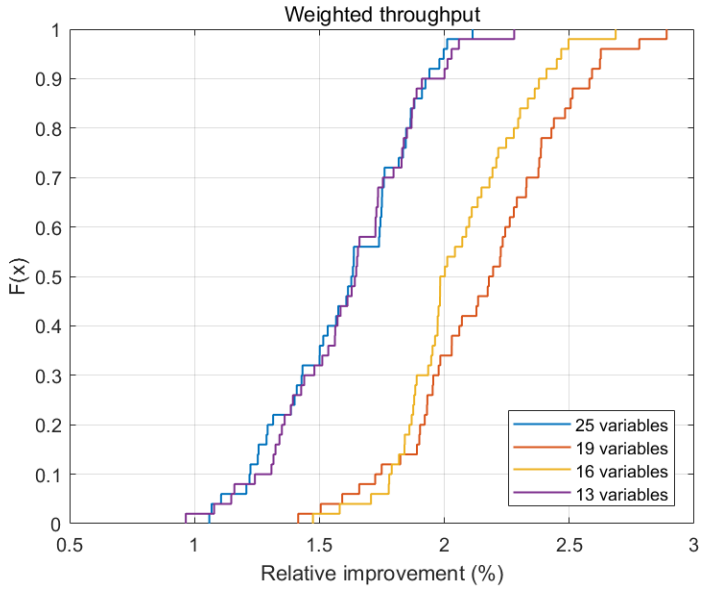


Figure 4.13 Empirical CDF plot of relative improvement in weighted throughput varying state spaces (# of links used as state variables) (based on 50 independent tests)

The average optimisation time in Table 4.9 exhibits an exponential growth with the increase of variables. Since they all have the same number of training points (i.e. 451 points), the only factor that may affect the optimisation time is the number of state variables. This is reasonable since the exponential correlation function needs to calculate the distance between two points along every dimension, and therefore an increase in the number of dimensions complicates the estimation of correlation, causing an increase in the estimation and optimisation time. This finding also highlights the importance of input variable selection.

	25 variables	19 variables	16 variables	13 variables
Optimisation Time	11.9816s	8.1077s	7.9465s	7.9355s

Table 4.9 Average optimisation time of Kriging-based real time signal controls with varying state variables

4.5.2 Demand level

In order to investigate how the demand level affects the performance of the proposed real time signal controls, four cases with different demand levels are generated and compared:

- **Case 1**: 50% of the baseline demand

- **Case 2**: 80% of the baseline demand
- **Case 3**: 100% of the baseline demand
- **Case 4**: 120% of the baseline demand

Figure 4.14 shows the MAPE of the Kriging model under these four demand levels. It indicates that once the decreasing trend has stabilised (i.e. over 300 points), MAPE decreases as the demand level increases. In addition, the figure shows that all four cases render low MAPE, with a difference within 1%. It can be concluded that the proposed surrogate model provides an accurate approximation of the target relationship that is not sensitive to the level of network saturation.

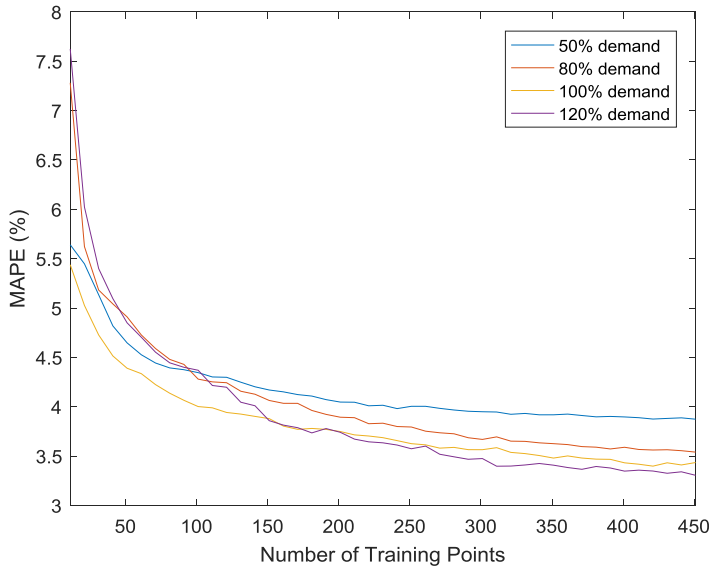


Figure 4.14 Average MAPE of Kriging models under varying demand levels (based on 50 independent tests)

To investigate the approximation accuracy under various demand levels in more detail, the RMSE for these four cases are plotted in Figure 4.15. This shows that Case 1 (50% demand) has a much lower RMSE than the other three cases. The reason for its low RMSE but high MAPE is that its actual total throughput is small due to its low demand level; hence the MAPE is sensitive to the difference between the actual and estimated performance. Besides Case 1 (50% demand), the other three cases have similar RMSE. This is because their demands are higher, which saturates the network and reduces the variability of total

throughput (i.e. the objective of Kriging), such that the increase of demand level does not have a significant impact on approximation accuracy.

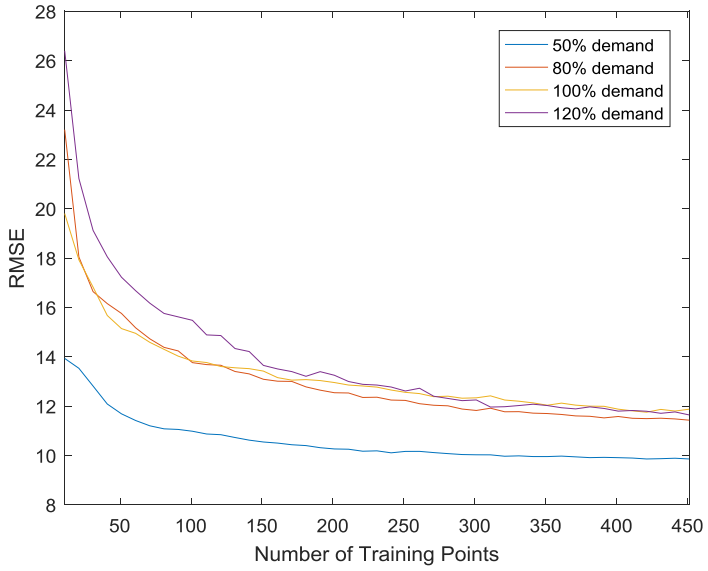


Figure 4.15 Average RMSE of Kriging models under varying demand levels (based on 50 independent tests)

The average network performance under Kriging-based real time signal control and *'GCC'* are shown in Table 4.10 and Table 4.11. In both cases, while the total throughput is positively related to the demand level, it tends to become more stable when the demand level is high (i.e. between 100% demand and 120% demand), which indicates that the network is saturated. With the increase of demand level, the increasing average delay also indicates that the network has become congested.

	Total Throughput	Average Delay	Weighted Throughput
50% Demand	2457.71 Veh	282.17s	4.4326
80% Demand	3259.76 Veh	648.98s	5.5931
100% Demand	3529.27 Veh	904.89s	5.9641
120% Demand	3607.58 Veh	1199.04s	6.2193

Table 4.10 Average performance of Kriging-based signal control under varying demand levels (based on 50 independent tests)

	Total Throughput	Average Delay	Weighted Throughput
50% Demand	2403.73 Veh	325.16s	4.4315
80% Demand	3137.23 Veh	708.06s	5.5056
100% Demand	3351.36 Veh	984.209s	5.8377
120% Demand	3458.26 Veh	1261.76s	6.1145

Table 4.11 Average performance of *'GCC'* under varying demand levels (based on 50 independent tests)

	Total Throughput	Average Delay	Weighted Throughput
50% Demand	2.2452%	13.2666%	1.0991%
80% Demand	3.9092%	8.3287%	1.5912%
100% Demand	5.3098%	8.0669%	2.1642%
120% Demand	4.3198%	4.9761%	1.7144%

Table 4.12 Average relative improvement of Kriging-based signal control over *'GCC'* under varying demand levels (based on 50 independent tests)

Due to the different demand levels in these four scenarios, the relative improvement is used to represent the performance of this real time signal control system. Table 4.12 shows the average relative improvement (%) of the Kriging-based real time signal control over the baseline '*GCC*' across varying demand levels, averaged over 50 independent tests. The following observations can be made from the table:

i. Under all the demand levels, the Kriging-based real time signal control outperforms '*GCC*' in terms of the three performance indicators.

ii. The improvement in average delay is more significant than that in total throughput for all four demand scenarios, especially for low demand levels. When the network is undersaturated, the room for reducing delay is greater than that for increasing throughput.

iii. The improvement in weighted throughput is far less than that in total throughput and average delay. The one-hour session contains too many time steps such that the last

few time steps have little contribution to the weighted throughput. The weighted throughput is mainly dominated by the throughput of the initial part of the whole period, such that the overall improvement in weighted throughput is not significant.

iv. For average delay, the efficacy of the proposed real time control decreases as the demand level increases. This is because when the network becomes saturated, adjusting signal timing has a weaker impact on network delays.

v. In terms of total throughput and weighted throughput, the improvement increases as the demand level grows from 50% to 100% but drops when the demand level is 120%. The indicates that the improvement offered by the Kriging-based control over the fixed-time plan peaks at a certain demand level, but reduces when the network is either under or oversaturated.

vi. When the traffic network is oversaturated (i.e. 120% demand), the improvements in these three performance indicators are reduced. The principle of signal control is to allocate the green time according to the traffic condition of each link in order to maximise the use of green time. In this experiment, the demand level of all the links increases simultaneously. When all the links are congested, the efficacy of signal control becomes insignificant.

Figure 4.16 depicts the queue length against time under the Kriging-based signal control and '*GCC*'. Four subplots represent four cases with different demand levels. As the demand level increases, longer queues are formed outside the network. For all the cases, the Kriging-based signal control leads consistently to shorter queues than '*GCC*', which indicates that the Kriging-based signal control can relieve congestion. Moreover, the queue length increases with time due to the oversaturation, except for the case of 50% demand under the Kriging-based signal control. In the last subplot (i.e. 50% demand), the curve of Kriging-451 stabilises, which indicates that the Kriging-based signal control is effective in stopping the formation of queues.

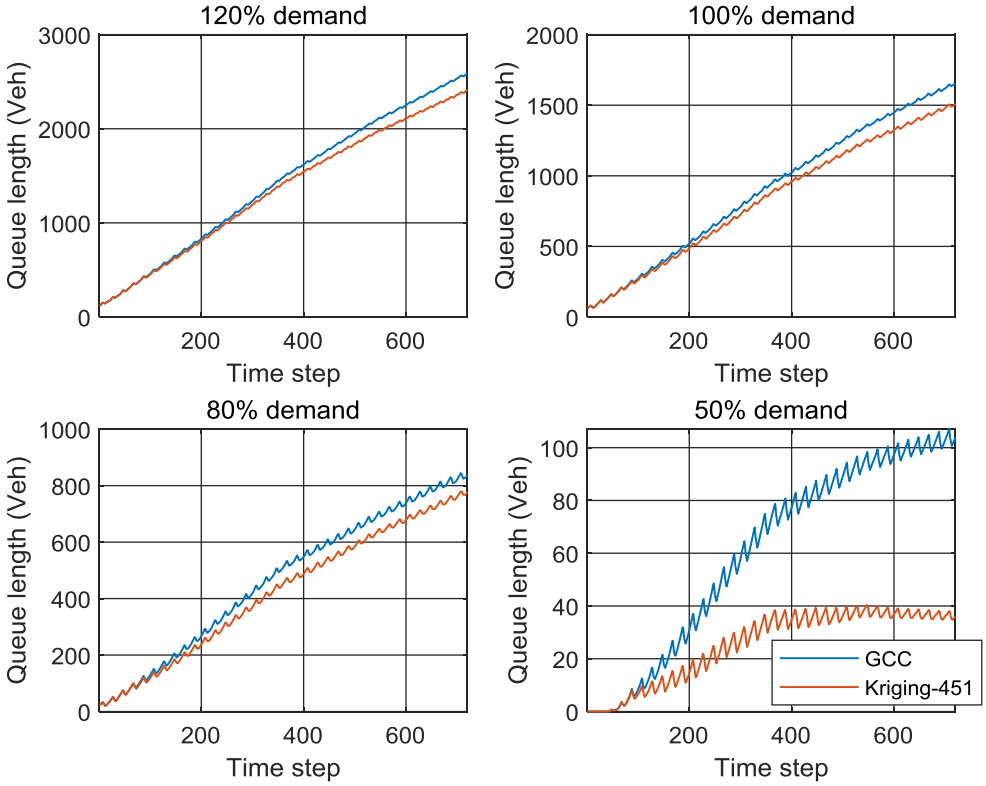


Figure 4.16 Average queue length against time under the control of 'Kriging-451' and *'GCC'* under varying demand levels (based on 50 independent tests)

4.5.3 Demand variability

The stochasticity of the demand poses a challenge to real time signal controls in general. For the Kriging model, higher demand variability means a larger state space and a more complex state-control-objective relationship. In order to verify the adaptability of the proposed Kriging-based signal control to demand variability, three demand cases with low, medium and high variability are generated. In this research, the variability of the demand is controlled by the value of *r*, which is essentially the noise-to-signal ratio in the generation of demand. Here, three scenarios are generated, with *r* equal to 0.1, 0.2 and 0.3, respectively:

- **Case 1**: High variability, $r = 0.3$
- **Case 2**: Medium variability, $r = 0.2$
- **Case 3**: Low variability, $r = 0.1$

The approximation accuracy of the Kriging models is illustrated in Figure 4.17. This shows

that MAPE increases with demand variability, which indicates that it is more challenging to estimate the performance when the demand is highly variable. Taking the MAPE of 4.5% as a reference, both the low variability and medium variability cases need less than 50 points to achieve that, while in the high variability case it needs around 200 points. This is because an increase in the variability of demand enlarges the design space, such that more training points are needed to achieve the same level of approximation accuracy for Case 1 (High variability) than Case 2 (Medium variability) and Case 3 (Low variability).

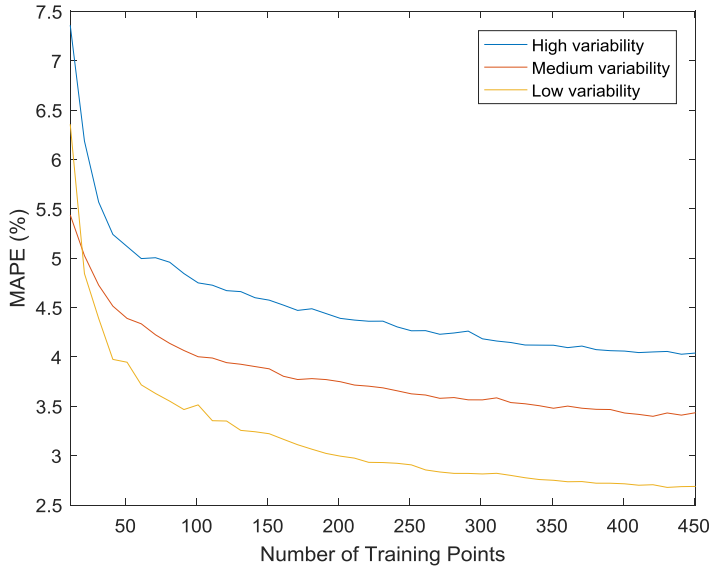


Figure 4.17 Average MAPE of Kriging models under varying demand variabilities (based on 50 independent tests)

Table 4.13 and Table 4.14 show the average performance (i.e. total throughput, average delay and weighted throughput) of the Kriging-based signal control and '*GCC*' under varying demand variabilities. For '*GCC*', the performance deteriorates as the demand variability increases. This can be explained by the fact that it cannot adapt to traffic variation well due to its fixed-time nature, which limits its application in conditions with high demand variability. The Kriging-based signal control has similar characteristics, which is due to the decrease in the approximation accuracy of the corresponding Kriging model with the increase in demand variability.

	Total Throughput	Average Delay	Weighted Throughput
High Variability	3486.42 Veh	936.56s	5.8983
Medium Variability	3529.27 Veh	904.89s	5.9641
Low Variability	3540.32 Veh	899.68s	5.9899

Table 4.13 Average performance of Kriging-based signal control under varying demand variabilities (based on 50 independent tests)

	Total Throughput	Average Delay	Weighted Throughput
High Variability	3339.65 Veh	998.51s	5.8051
Medium Variability	3351.36 Veh	984.21s	5.8377
Low Variability	3367.42 Veh	978.48s	5.8688

Table 4.14 Average performance of *'GCC'* under varying demand variabilities (based on 50 independent tests)

Table 4.15 shows the relative improvement of the Kriging-based control over '*GCC*'. The positive improvements show that the Kriging-based signal control always performs better than '*GCC*' regardless of the demand variability. Interestingly, while, as expected, High Variability renders less improvement than Medium Variability, Low Variability also leads to less improvement than Medium Variability. Table 4.14 suggests that this is due to the fact that the performance of the Kriging-based signal control also deteriorates under high variability, while '*GCC*' performs relatively well under low demand variability.

	Total Throughput	Average Delay	Weighted Throughput
High Variability	4.3971%	6.2182%	1.6058%
Medium Variability	5.3098%	8.0669%	2.1642%
Low Variability	5.1337%	8.0557%	2.6039%

Table 4.15 Average relative improvement of Kriging-based signal control over *'GCC'* under varying demand variability (based on 50 independent tests)

4.5.4 Weighting parameter A

As discussed in Section 4.3, weighting parameter *A* is introduced to control the balance between the optimality and the reliability of Kriging-based real time signal control. It

represents the sensitivity to the estimation error. When A is zero, the decision is made under the assumption that the estimation from a surrogate model is reliable enough for it to be used directly for optimisation. In order to investigate how the weighting parameter *A* affects the performance of the Kriging-based signal control, three cases with varying *A* are considered:

- **Case 1**: Weighting parameter $A = 0$
- **Case 2**: Weighting parameter $A = 1$
- **Case 3**: Weighting parameter $A = 2$

It is also necessary to realise that the Kriging model has different degrees of approximation accuracy with an increasing number of training points. Therefore, six Kriging-n models with $n = 151, 201, 251, 351, 451$ and 551 are constructed, where *n* is the number of training points. This helps to identify how the influence of weighting parameter *A* behaves under different approximation accuracies.

Figure 4.18, Figure 4.19 and Figure 4.20 show how the three performance indicators: total throughput, average delay and weighted throughput vary when Kriging models with different numbers of training points are used for signal optimisation. The three curves represent the cases with different *A* values. The Figures show that when *n* is low (i.e. below 351), a larger *A* value leads to better network performance for the same Kriging-n model. This is because the approximation accuracy of the Kriging model is low when *n* is small, such that the influence of estimation error on signal control optimisation is significant. The estimated outputs generated may therefore have a large estimation error and, if they were to be applied, the traffic network may not work as expected, resulting in the deterioration of performance. Under this condition, a large value of *A* can ensure that a solution with low estimation error is selected such that the actual network performance is not far from what is estimated. When *n* exceeds 351, however, the case of *A*=2 has the worst performance. This is due to the fact that the overall approximation accuracy at this time is high, therefore overemphasising the estimation error may generate a conservative optimal solution. When *n* is 551, the three cases have a similar level of performance, showing that with high approximation accuracy, the performance of the signal control is not sensitive to the value of *A*.

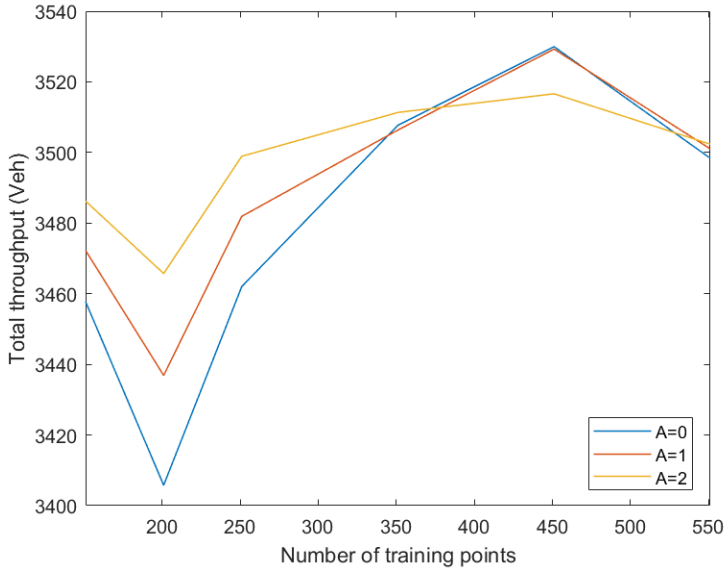


Figure 4.18 Average total throughput versus the number of training points under different values of weighting parameter A (based on 50 independent tests)

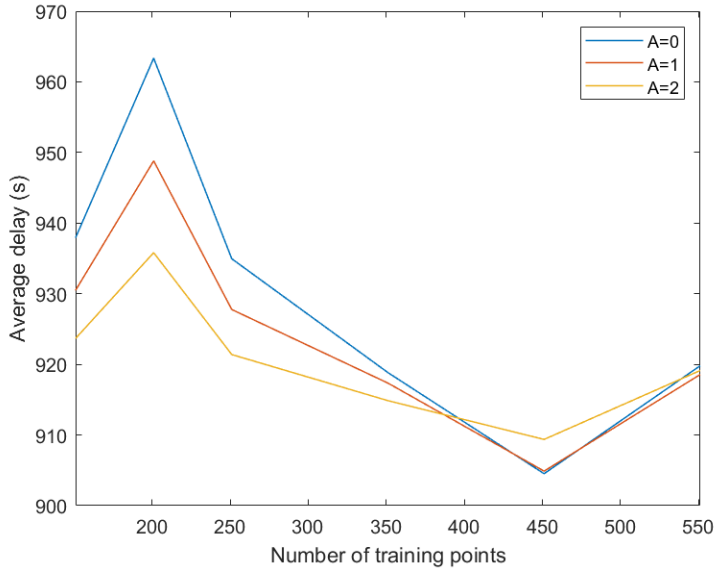


Figure 4.19 Average delay versus the number of training points under different values of weighting parameter A (based on 50 independent tests)

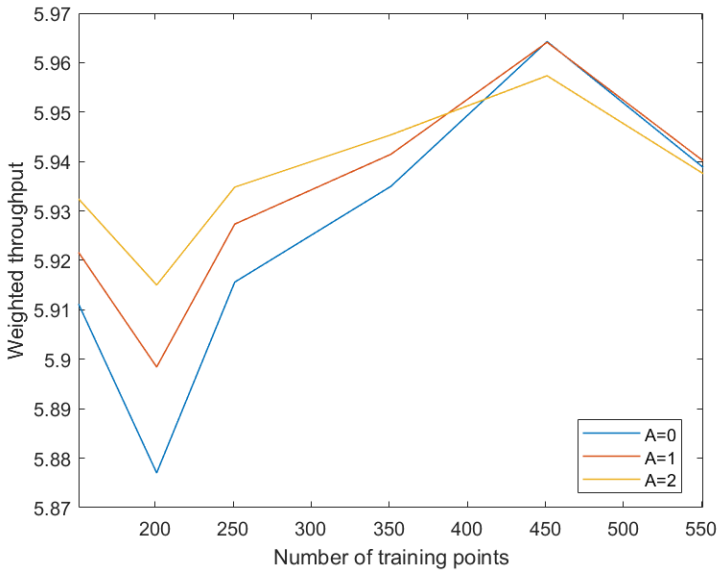


Figure 4.20 Average weighted throughput versus the number of training points under different values of weighting parameter A (based on 50 independent tests)

Figure 4.21 presents the CDF plots of the total throughput (i.e. objective function) when the traffic network is controlled by the Kriging-based signal control with different values of weighting parameter *A*. Each subplot represents a Kriging model with a different number of training points. When *n* is 151, the three curves are apart from each other. As *n* increases, except for the cases of *n*=201 and 551, the three curves move along the positive direction of the *x*-axis, indicating that their performance has been improved. The curve representing a low value of *A* moves faster than the curve representing a high value of *A*, meaning that the curves get closer to each other. When *n*=351, the three curves almost coincide, and when n is larger than 351, curves representing *A*=0 and *A*=1 surpass the curve representing *A*=2 for the first time. It is worth noting, however, that when *n* is large enough, the distance between the curves decreases gradually, which indicates that the selection of the optimal solution is not sensitive to the value of *A* due to the small estimation error. Ideally, when the estimation errors of all the points in the design space are zero, the value of *A* will not affect the optimisation at all; although this ideal condition can never be achieved.

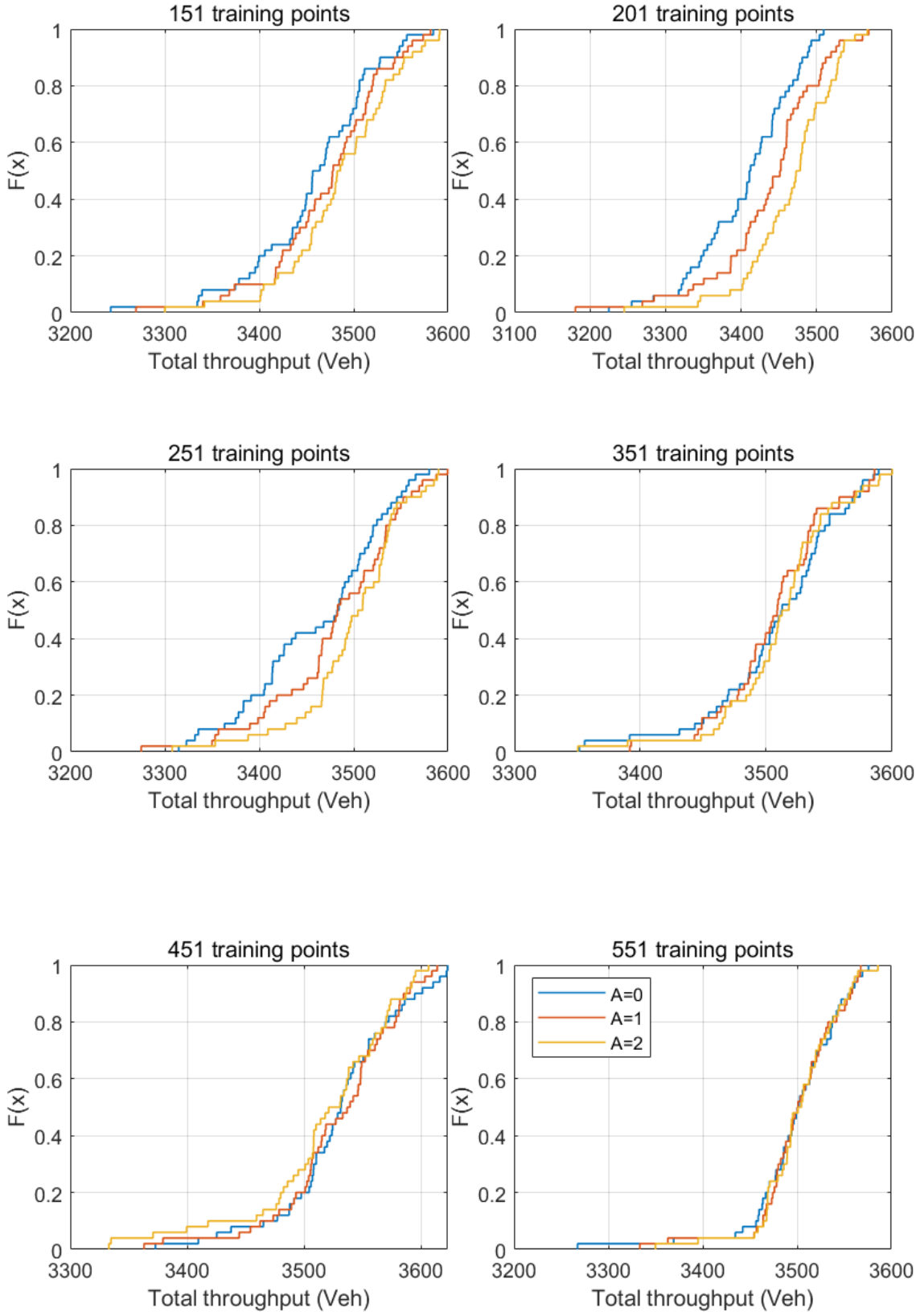


Figure 4.21 Empirical CDF plot of total throughput for Kriging-based signal controls with different values of weighting parameter A (based on 50 independent tests)

In the optimisation process, the network performance (i.e. total throughput) of the next control period is estimated before the generated signal plan is applied. In order to assess the reliability of the generated optimal solution, the error that exists between the estimated performance and the corresponding actual performance is calculated. Table 4.16 shows the MAPE of the data collected at the end of each control period. Several cases with varying values of A and the number of training points are considered. Comparing the results in each column, it is evident that MAPE shows a decreasing trend, except for the case of n=201 and n=551. This indicates that the approximation accuracy of the Kriging models improves as the number of training points increases. When comparing the results in each row, meanwhile, it can be seen that MAPE decreases as the value of A increases, which shows that the PSO with a high value of A tends to select a solution with a low estimation error.

	$\mathbf{A = 0}$	$\mathbf{A = 1}$	$\mathbf{A = 2}$
$\mathbf{n = 151}$	7.5170%	6.7268%	5.7988%
$\mathbf{n = 201}$	9.0772%	8.0517%	6.5634%
$\mathbf{n = 251}$	5.3683%	4.9238%	4.1180%
$\mathbf{n = 351}$	4.0472%	3.7755%	3.5009%
$\mathbf{n = 451}$	3.0334%	2.7427%	2.6516%
$\mathbf{n = 551}$	4.1855%	4.1693%	4.1923%

Table 4.16 Average MAPE of the Kriging-based signal controls with varying weighting parameter A and number of training points (based on 50 independent tests)

4.5.5 Baseline signal

A fixed-time baseline (i.e. '*GCC*') signal is adopted to enhance the efficiency in the surrogate model construction and on-line optimisation. The test results have proved that the Kriging-based surrogate model can behave better than its corresponding fixed-time baseline signal '*GCC*'. In order to investigate how different baseline signals affect the performance of Kriging-based signal control, the following three baseline scenarios are considered:

- **Scenario 1:** The fixed-time signal control based on the real-world test network in

Glasgow, as illustrated in Liu et al. (2015), referred to as '*GCC*'.

- **Scenario 2:** The fixed-time signal control generated by simulation-based optimisation, referred to as '*TOD*'.

- **Scenario 3:** The fixed-time signal control where each intersection is controlled independently following the equi-saturation policy (Webster, 1958), referred to as '*Webster*'.

'*GCC*' is a fixed-time signal control provided by Glasgow City Council to approximate the SCOOT system in the morning peak, and was used in the previous numerical tests. '*TOD*' is a fixed-time signal control specially designed for the one-hour morning peak. It is solved by PSO, with LKWM to estimate the fitness of each particle. '*Webster*' is an isolated fixed-time signal control, where each intersection makes its own decision independently. The green time of each link is allocated to balance the red time costs of each link, which is a function of the degree of saturation and green split. For the detailed procedures refer to **Appendix C**.

Signal samples for the construction of the Kriging model and the PSO are generated around these three baseline scenarios, respectively. Based on these three baseline signals, three Kriging-based real time signal controls are obtained, denoted as '*Kriging-GCC*', '*Kriging-TOD*' and '*Kriging-Webster*' for ease of reference. Table 4.17, Table 4.18 and Table 4.19 show the comparison of the fixed-time signal controls and their corresponding Kriging-based real time signal controls in terms of total throughput, average delay and weighted throughput.

From the three tables, it can be seen that the two coordinated fixed-time signal controls: '*GCC*' and '*Kriging fixed*' perform much better than the isolated fixed-time signal control: '*Webster*'. The lack of coordination between intersections means that '*Webster*' fails to mitigate congestion at the network level.

In addition, '*Kriging-GCC*', '*Kriging-TOD*' and '*Kriging-Webster*' have an entirely different level of performance, positively correlated with the performance of the corresponding

baseline signal controls. For example, '*TOD*' is the one with the best performance among all the fixed-time signal controls; hence '*Kriging-TOD*' performs the best among all the Kriging-based signal controls. This is because the sampling of signal control variables for surrogate model construction and swarm particles for PSO are both conducted locally. Hence improper selection of the baseline signal will degrade the performance of the Kriging-based real time signal control.

Comparing the relative improvement reveals that the Kriging-based signal controls always outperform their corresponding baseline signal. Moreover, the better the performance of the baseline signal, the smaller the improvement that the Kriging-based real time signal controls can achieve. In other words, if the performance of the baseline signal control is good, there is little room left for improvement, such that the relative improvement will be small. The improvement of '*Kriging-TOD*' is small. This is similar to the finding in Shelby et al. (2008) that real time signal control cannot provide significant improvement if the fixed-time signal plan is well optimised.

	Total Throughput	**Average Delay**	**Weighted Throughput**
GCC	3351.36 Veh	984.21s	5.8377
Kriging-GCC	3529.27 Veh	904.89s	5.9641
Improvement (%)	5.3098%	8.0669%	2.1642%

Table 4.17 Comparison between *'GCC'* and its corresponding Kriging-based real time signal control (based on 50 independent tests)

	Total Throughput	**Average Delay**	**Weighted Throughput**
TOD	3598.20 Veh	874.12s	6.1006
Kriging-TOD	3632.71 Veh	861.55s	6.1114
Improvement (%)	0.9600%	1.4389%	0.1757%

Table 4.18 Comparison between *'TOD'* and its corresponding Kriging-based real time signal control (based on 50 independent tests)

	Total Throughput	Average Delay	Weighted Throughput
Webster	3057.69 Veh	1066.88s	5.7497
Kriging-Webster	3495.64 Veh	920.41s	5.9896
Improvement (%)	14.3372%	13.6358%	4.1905%

Table 4.19 Comparison between *'Webster'* and its corresponding Kriging-based real time signal control (based on 50 independent tests)

4.6 Conclusions

This chapter has developed a real time signal control framework based on surrogate modelling. A modified Particle Swarm Optimisation, a meta-heuristic method, was applied to the real time optimisation. The standard framework of Particle Swarm Optimisation was introduced in Section 4.1 and a few modifications were then made to it so that it could be integrated more effectively with the surrogate-based real time signal control. These modifications are summarised as follows:

- The initial swarm particles are sampled around a baseline signal generated in advance rather than being evenly distributed in the whole design space. The baseline signal guides the searching area to prevent the final solution from being trapping at the local optimum, with consequent undesirable performance. Additionally, this modification narrows down the searching space by generating the particles around the region that has a high probability of finding a near-optimal solution, thus reducing the optimisation time.

- The dynamic inertia weight ω is helpful to enhance the optimisation efficiency. The value of ω decreases iteration by iteration, such that the searching is gradually dominated by local searching. The rate of descent was controlled by the parameter α, which has a positive value less than 1. This can accelerate the convergence speed of PSO.

- Since the surrogate model has an inevitable estimation error this needs to be taken into account in any optimisation based on it. Thus, the weighting parameter A was included in

the fitness function in order to balance the optimality and reliability.

An empirical experiment was carried out on the same test network as used in Chapter 3. The results showed that the efficacy of Kriging-based real time signal control is positively correlated with the approximation accuracy of the corresponding Kriging model. The Kriging model with more training points has higher global accuracy, such that the signal plan generated based on it has better capability to manage the traffic flow. The t-test results and the figure of queue length against time also prove that Kriging-based real time signal control is better than '*GCC*' in terms of mitigating congestion.

Multi-scenario analysis was then conducted in order to investigate the robustness of this Kriging-based real time signal control to different traffic conditions and parameters,. The key findings are summarised as follows:

- The Kriging model indicates the importance of each variable with the parameter θ. It guides the selection of input variables. Four cases with different numbers of traffic state variables were generated and it was evident that their ranking in terms of approximation accuracy varied with the number of training points. This is due to the fact that a Kriging model with more input variables can capture more detailed characteristics of the underlying network but needs more training points to achieve the same level of accuracy as a Kriging model with fewer input variables. The optimisation results also confirmed this finding. Since the growth of the design space dimension leads to an increase in optimisation time, the decision on which variables should be involved is a trade-off between performance and efficiency.

- Both the performance of the Kriging-based signal control and '*GCC*' deteriorated when the demand level increased. Comparing the Kriging-based signal control with '*GCC*', the improvement in total throughput increased along with the demand level, but began to decline after peaking at a specific demand level. In contrast, the improvement in average delay was negatively correlated with the demand level. This indicates that the efficacy of Kriging-based signal control is insignificant when the network is oversaturated.

- The variability of demand can be controlled by the parameter *r*. The Kriging model showed a decreasing approximation accuracy as the demand variability increased. The average performance, however, indicated that the Kriging-based signal control has strong robustness to demand variability, especially in the case of low variability and medium variability. Based on the relative improvement over '*GCC*', this leads to the conclusion that Kriging-based signal control maximises its efficacy when the variability is at the medium level.

- In order to ensure the reliability of the signal control, the estimation error needs to be taken into account for signal optimisation. Parameter *A* represents the sensitivity to the estimation error. It was proved that, with a large value of *A*, a solution with low estimation error tends to be selected. Thus, if the surrogate model has low approximation accuracy, the signal generated when *A* is large leads to better network performance. The signal control generated by a PSO with a large value of *A* may, however, be too conservative when the approximation accuracy of the surrogate model is high.

- The performance of Kriging-based signal control was affected by the baseline signal used for surrogate model construction and on-line optimisation. The improper selection of a baseline signal degrades the performance of the Kriging-based real time signal control developed based on it.

The last finding highlighted that the performance of the Kriging-based real time signal control is sensitive to the baseline signal. This means that it would not be sensible to select the baseline signal randomly. In most cases, however, there is no alternative signal plan, and it is time-consuming to generate a new fixed-time signal plan. The following chapter investigates how to eliminate the influence of the baseline signal using iterative updates. In addition, it also discusses how the Kriging-based signal control can be adapted to systematic changes of demand through a series of update procedures.

Chapter 5

Real time signal control with surrogate model updates

The surrogate-based real time signal control proposed in Chapter 4 was constructed based on relatively stationary demands, although short-term variations were considered. Therefore, the adaptability of this proposed signal control to demand change needs to be tested. Moreover, the proposed signal control has no learning ability. Due to the virtue of being data-driven, however, it can be integrated with an adaptive response surface, such that the surrogate-based real time signal control system can be re-trained repeatedly based on the latest information relating to the traffic network.

In this chapter, an adaptive baseline signal updating strategy is proposed to solve the problem of performance degradation caused by the improper choice of the baseline signal found in Section 4.5.5. The procedures of adaptive sampling with different baseline signals are described in Section 5.1. Section 5.2 introduces the real time signal control with an adaptive response surface aiming to handle the evolution of demand level. Its performance is tested and validated under the systematic change of demand. In Section 5.3, the design plan for an adaptive response surface is investigated by carrying out sensitivity analyses on two major aspects of the update plan: update frequency, and the number of infill points.

5.1 Adaptive sampling strategy with different baseline signals

Section 4.5.5 highlights that the baseline signal affects the performance of the surrogate-based real time signal control system. The optimisation of a fixed-time signal control is typically time-consuming, however, and it is not sensible to spend a large computational budget on baseline signal optimisation. It is therefore crucial to consider how to eliminate the adverse effects of an improper baseline signal. In this section, an off-line adaptive updating strategy for the baseline signal is developed to ensure that the surrogate-based real time signal control can always achieve a reasonable level of performance (i.e. total throughput, average delay and

weighted throughput) eventually, regardless of the initial baseline signal. The whole process is carried out off-line before the system is actually implemented. The aim of this off-line adaptive updating strategy is to address the problem of performance degradation due to the improper baseline signal.

The principle underlying updating the adaptive baseline signal is to replace the old baseline signal with a new one that leads to better network performance. The new baseline signal can be estimated on the basis of the optimal signal plans generated by the optimisation tests. Section 4.5.5 has proved that the Kriging-based real time signal control is always superior to the corresponding baseline signal. The involvement of a better baseline signal should therefore lead to further performance improvement in the Kriging-based real time signal control.

The adaptive baseline signal update procedure is shown in Figure 5.1. First, a surrogate-based real time signal control system is established based on an initial baseline signal, and applied for optimisation. According to the signal decisions generated, a new baseline signal with improved quality can be derived, since an optimal signal plan obtained from the surrogate-based real time signal control always leads to better performance than its corresponding baseline signal. Each signal decision is only optimal under a specific condition, however. Therefore, a number of optimised signals are generated and averaged in order to capture their macroscopic features and form a new baseline signal. Finally, the surrogate-based real time signal control system is re-trained using this new baseline signal.

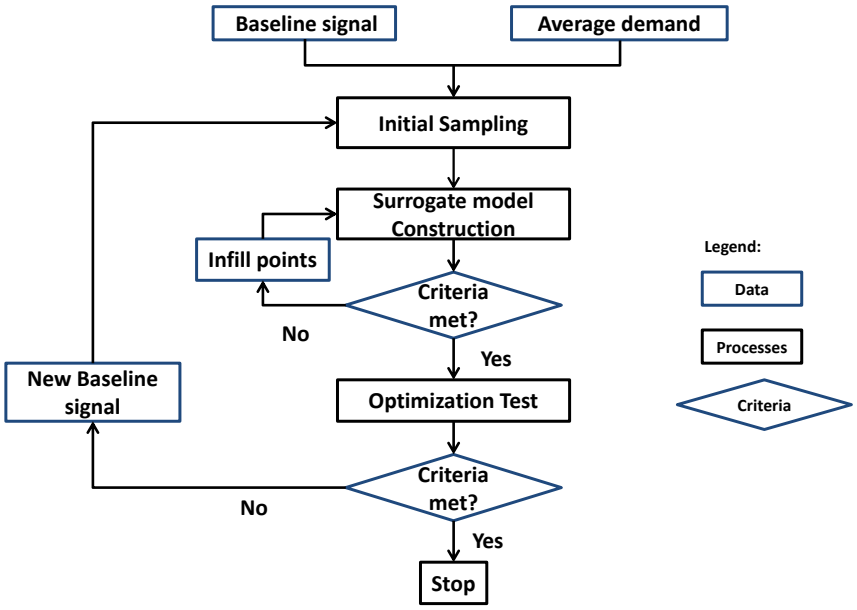


Figure 5.1 Framework of the adaptive baseline signal updating strategy

An empirical test, an extension of that in Section 4.5.5, is conducted to test whether this adaptive baseline signal updating strategy can eliminate the adverse impacts of improper baseline signals. Three cases are considered, with their initial baseline signals being: '*GCC*', '*TOD*' and '*Webster*', respectively. In order to obtain a new baseline signal, 50 optimisation tests are carried out. After five updates, the performance of the initial baseline signal and six Kriging-based real time signal controls (one based on initial baseline signal and five based on updated baseline signals) are evaluated under the same conditions.

Table 5.1, Table 5.2 and Table 5.3 show the performance of these signal controls in terms of the total throughput, average delay and weighted throughput, which are averaged across 50 replications. The first row of each table displays the performance of the initial fixed-time baseline signal, while the second row depicts the performance of the Kriging-based real time signal control system developed based on the initial baseline signal. The subsequent five rows present the performance of the Kriging-based real time signal control system after one to five adaptive updates, respectively.

	Total Throughput	**Average Delay**	**Weighted Throughput**
GCC	3351.36 Veh	984.21s	5.8377

Kriging-GCC	3511.61 Veh	918.73s	5.9375
Adaptive update1	3624.44 Veh	855.17s	6.2111
Adaptive update2	3671.99 Veh	832.59s	6.3241
Adaptive update3	3649.50 Veh	838.09s	6.2521
Adaptive update4	3652.46 Veh	836.72s	6.4219
Adaptive update5	3646.79 Veh	839.96s	6.3405

Table 5.1 Average performance of *'GCC'* and *'Kriging-GCC'* before and after baseline signal updates (based on 50 independent tests)

	Total Throughput	**Average Delay**	**Weighted Throughput**
TOD	3598.20 Veh	874.12s	6.1006
Kriging-TOD	3632.71 Veh	861.55s	6.1114
Adaptive update1	3650.41 Veh	840.95s	6.2791
Adaptive update2	3658.27 Veh	836.59s	6.2919
Adaptive update3	3587.24 Veh	861.19s	6.2645
Adaptive update4	3652.66 Veh	832.16s	6.3656
Adaptive update5	3607.16 Veh	854.77s	6.2877

Table 5.2 Average performance of *'TOD'* and *'Kriging-TOD'* before and after baseline signal updates (based on 50 independent tests)

	Total Throughput	**Average Delay**	**Weighted Throughput**
Webster	3057.69 Veh	1066.88s	5.7497
Kriging-Webster	3495.64 Veh	920.41s	5.9896
Adaptive update1	3660.20 Veh	840.55s	6.2950
Adaptive update2	3654.32 Veh	836.94s	6.3082
Adaptive update3	3621.09 Veh	844.61s	6.3102
Adaptive update4	3622.42 Veh	844.25s	6.2989
Adaptive update5	3662.52 Veh	831.02s	6.3261

Table 5.3 Average performance of *'Webster'* and *'Kriging-Webster'* before and after baseline signal updates (based on 50 independent tests)

For the case based on '*Webster*', which has the worst performance, all three performance

indicators are improved as the baseline signal is updated. After five updates, when compared with '*Kriging-Webster*', the total throughput increases by 4.8%, the average delay is reduced by 9.7% and the weighted throughput increases by 5.6%. The overall performance of the other two cases is also improved. This shows that the adaptive baseline signal updating strategy can enhance the performance of Kriging-based real time signal control.

Performance is not improved continuously with the updates of the baseline signal, however. As shown in the three tables, the three performance indicators fluctuate around the peak once that is reached. Due to the inherent uncertainties in the Kriging model and PSO, the obtained solution may deviate from the *'true'* optimum. Moreover, when the baseline signal is already near the *'true'* optimum, the improvement caused by the update of the baseline signal is not statistically significant. The superimposition of these two effects results in the fluctuation seen here.

Comparing the performance of '*GCC*' between Adaptive update 2 and Adaptive update 3, where the first performance degradation is observed, the total throughput decreases by 0.6%. Meanwhile, the total throughput of '*TOD*' is reduced by 1.9% from Adaptive update 2 to Adaptive update 3, and for '*Webster*' the total throughput decreases by 0.16% from Adaptive update 1 to Adaptive update 2. The performance deterioration is small; hence the conclusion that the adaptive baseline signal updating can improve the performance overall is still valid.

Moreover, it can be seen from these three tables that all the cases outlined eventually have a similar level of performance after five updates. This proves that the influence of the initial baseline signal can be eliminated by the adaptive baseline signal updating strategy.

5.2 Real time signal control based on adaptive response surface

The surrogate-based real time signal control system can only interpret a known and fixed state-control-objective relationship (i.e. response surface) around the region of interest. In practice, however, systematic changes may occur in the traffic system, such as seasonal demand shifts and changes in flow characteristics, thereby violating the approximation

accuracy of the response surface. In order to adapt to these systematic changes, the signal control system needs to update itself. A response surface updating strategy is therefore designed to construct an adaptive response surface.

5.2.1 Adaptive response surface

A fixed response surface that cannot adapt to any systematic changes in the traffic system may result in performance deterioration. Although the mechanism of the deterioration is complex, some possible factors are identified and summarised below:

- The centre of the state space shifts. As mentioned in previous sections, the state variables are not evenly distributed throughout the entire design space. In order to reduce the computational burden, the response surface is constructed around a predefined central area, which is determined according to the average traffic pattern and the baseline signal. In other words, a local surrogate model is constructed only in the region with a high probability of occurrence. Since traffic patterns are not static over time, however, the new centre of interest may not be covered by enough samples in the original surrogate model. The resulting estimation errors may hinder the PSO from finding the *'true'* optimal results, which consequently results in the performance of the real time signal control degrading.

- The performance of the baseline signal deteriorates. Systematic changes in the traffic system may mean that the baseline signal is no longer be suitable for the current traffic conditions. Since the optimisation is conducted around the baseline signal, the *'true'* optimum may be missed if it is outside of the search region of the baseline signal.

- Traffic dynamics change, and hence the original state-control-object relationship cannot represent the current network interactions accurately. The factors leading to the invalidation of the original state-control-object relationship include a change in exogenous factors and traffic rules. Such a change of network dynamics leads to the failure of the response surface and thereby results in an increase in the approximation

error.

In order to address the problems mentioned above, an adaptive response surface is developed. The adaptive update of the response surface can be achieved by learning from the latest information relating to the traffic network. The entire updating process relies on the data collected during optimisation, and there is no need to invoke simulation models.

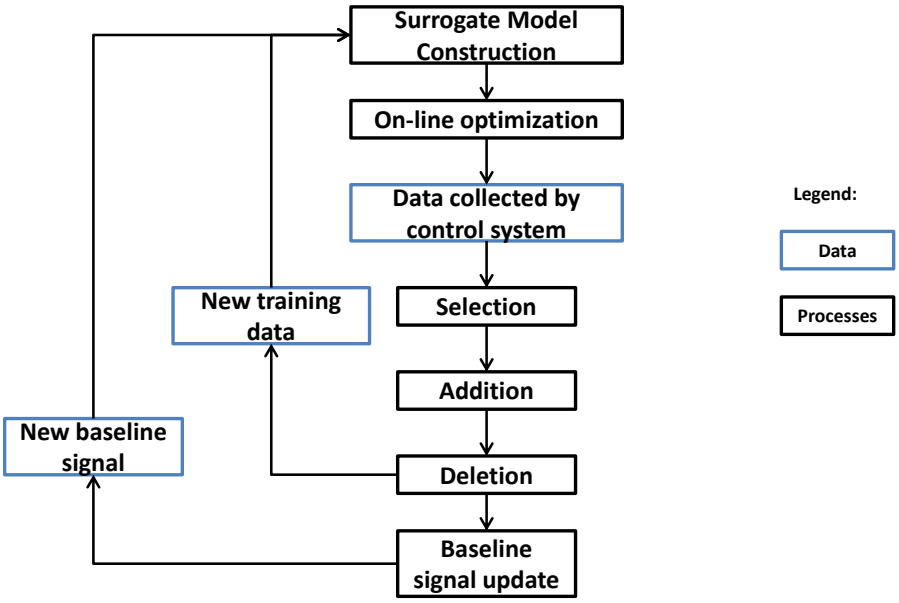


Figure 5.2 Framework of on-line adaptive update

As illustrated in Figure 5.2, the on-line update of the response surface contains four main stages, including: 1) selection, 2) addition, 3) deletion and 4) baseline signal update. A new surrogate model can be constructed and implemented after these four stages, which are introduced below:

- **Selection:** Implement the surrogate-based real time signal control for a period (e.g. several epochs), during which another set of data can be collected. These data are stored in the candidate set *C*. For each point in *C*, calculate the absolute relative error between the estimated outputs from the surrogate model and the true observations. *M* points with the largest errors are selected from *C* to construct the infill set *I*. Points with large errors indicate that the local accuracies around them are low.

- **Addition:** Construct a new surrogate model based on the updated training points set, including the current training points set *X*, and *M* points selected from the previous step. The addition of new points increases the sample density of the region with low approximation accuracy. In addition, the relationship between training points is reconstructed by re-training a new surrogate model.

- **Deletion:** Based on the new surrogate model proposed above, construct the correlation matrix $R \in \mathrm{R}^{/X/\times M}$ between the original training samples and the *M* added points. Calculate the average value of each row, which represents the average correlation of each original training point with *M* added points. *M* points from the current training set *X* with the lowest average correlation values are deleted from the training point set *X*. Then the points in infill set *I* are added to set *X* to form a new training set. A new surrogate model is trained with the new training point set. The points in *I* indicate the new centre of interest in the design space. Since the deleted points have low contributions to the estimation of the points around the new centre, their deletion should have little influence on the approximation accuracy.

- **Baseline signal update:** Update the baseline signal on the basis of the optimal signal plans generated. Together with the new surrogate model, a new surrogate-based real time signal control is developed and implemented.

The detailed algorithm of the on-line update is presented below.

Algorithm 5.1: On-line update of the surrogate model

Input Surrogate model trained with sample set *X*.

Step 1 In an on-line environment, implement the surrogate-based real time signal control and collect a set of points *C*.

Step 2 For every candidate point $c \in C$, calculate the absolute relative error between the estimated output and the actual observation. Construct the infill set I by selecting *M* points with the largest errors from the candidate set *C*.

Step 3 Train a new surrogate model with set $X \cup I$ and construct the correlation matrix

$R \in R^{|X| \times M}$ between the points in X and the points in I.

Step 4 Find M rows of R with the smallest row averages and form the set E from the corresponding samples.

Step 5 Let $X = X \cup I \backslash E$, and retrain the surrogate model

Step 6 Update the baseline signal and go back to step 1

5.2.2 Systematic change in demand

The traffic volume of a network is not stationary over time and Sunkari (2004) mentioned that demand change is the main reason for the degradation of signal control. This is regarded as a sign that the signal control needs to be retimed.

There is little research on the performance deterioration of real time signal control under systematic changes in demand, since it is believed that real time signal control can adapt to demand variation effectively. In fact, the only research on the deterioration of adaptive signal control is Stevanoic (2006). That study shows that, for a performance minimisation problem, the ageing effect of SCOOT can be quantified by comparing the gap between the growth of the performance of SCOOT and that of the fixed-time signal. If the gap increases with increasing demand, deterioration exists. The results show that the traffic model of SCOOT cannot estimate the approaching vehicles accurately when demand changes. Moreover, SCOOT does age in several cases. Stevanoic (2006) also emphasises that this quantitative method may underestimate the ageing of SCOOT.

Chapter 4 has shown that the surrogate-based real time signal control can handle short-term variations of demand, but the behaviour of surrogate-based real time signal control under systematic changes of demand, and the requirement for regular updates have not been studied. The change of performance indicators (PI) in the case of changing systematic demand arises from two main reasons. One is directly related to the change of demand, while the other is due to the deterioration of the signal control. The changes of PI caused by these two reasons may be additive or they may cancel out.

Table 5.4 shows the change in direction of total throughput and average delay with a change of demand, with '+' for an increase and '-' for a decrease. The other measure used in the previous chapter, weighted throughput, is a function of the total throughput and the average delay and cannot be detected directly. Since it is challenging to predict the change of weighted throughput with demand it is not included in the table. For increasing demand, if the total throughput increases and the average delay decreases, signal control may deteriorate. Similarly, for decreasing demand, signal control may deteriorate if the total throughput decreases and average delay increases. Besides these two cases, it is impossible to judge whether the signal deteriorates with a change in demand according to its performance, given the cancellation and additive effect.

		Increasing demand	**Decreasing demand**
Total throughput	Demand change	+	-
	Signal deterioration	-	-
Average delay	Demand change	+	-
	Signal deterioration	+	+

Table 5.4 The direction of performance change due to demand change and signal deterioration

5.2.3 Simulation test

An empirical experiment is conducted to test the adaptability of the adaptive Kriging-based real time signal control under a systematic change in demand. The initial demand profile is defined as TD_0, which is the average demand profile in Section 3.4.2., and it evolves to the terminal demand profile TD_1 after 100 epochs (i.e. days). During this period, the demand profile in each epoch changes gradually and linearly. On top of this, short-term perturbations are also considered by adopting the demand generation Equation (3.7). Three cases of systematic change are considered:

- **Case 1:** From TD_0 to TD_1, demand profile increases by 50%

- **Case 2:** From TD_0 to TD_1, demand profile decreases by 50%
- **Case 3:** From TD_0 to TD_1, the demand profile of each link changes independently in the range of $[-50\%\ 50\%]$

In the first two cases, the demand profile of all the inflow links changes simultaneously. In Case 3, the demand profile of each link is allowed to change independently by up to 50%, either increasing or decreasing.

Five demands are generated for each case, and three signal controls are presented and compared:

- **'*Fixed*':** The baseline signal of '*Kriging-GCC*' after two updates introduced in Section 5.1 is adopted, which is generated based on the initial demand profile TD_0. This is a fixed-time signal control and cannot react to the demand fluctuation.

- **'*Kriging*':** To be consistent with '*Fixed*', '*Kriging-GCC*' after two updates is selected, which is also introduced in Section 5.1. This can react to the demand fluctuation, but the response surface is fixed throughout the whole test period.

- **'*Adaptive Kriging*':** This is a Kriging-based real time signal control with an adaptive response surface. The initial response surface is the same as that of '*Kriging*' but is updated every two days, with ten points replaced in each update, following the Algorithm 5.1.

Analysis of results of Case 1

Figure 5.3, Figure and Figure show the evolution of the total throughput, average delay and weighted throughput over 100 days with the increase of demand shown on the left. The subplot on the right represents the CDF plots of the difference between signal controls. The difference is positive if the initial model in the legend has a better performance than the later model, while it is negative in the reverse case. Without considering the deterioration effect, the total throughput should grow with demand until the network reaches its capacity. The

increasing trends of '*Kriging*' and '*Adaptive Kriging*' indicate that the network is still not fully saturated. It can be seen, however, that in the case of '*Fixed*', the growth of total throughput stagnated after Day 20. This is because the decline of total throughput caused by the signal deterioration counteracts the increase in total throughput caused by the demand growth. From the aspect of average delay, the influences of demand growth and deterioration are additive; hence the increase in average delay cannot indicate whether these three signal control strategies deteriorate. It can therefore be concluded that '*Fixed*' may deteriorate with the growth of demand, and thus it cannot handle the increase in demand well.

In comparison, '*Kriging*' outperforms '*Fixed*' in terms of total throughput, average delay and weighted throughput, and the difference between them increases over time. As a result, '*Fixed*' cannot maintain its performance as demand increases. This is because when the network is saturated, the interactions between intersections are enhanced due to the spillback of the intersection. If spillback occurs, there is no room for any entering vehicles and the traffic condition is complex and highly variable. Under this condition, it is crucial to adjust the signal plan according to the traffic conditions of adjacent intersections, and this cannot be realised by a fixed-time signal control.

For the comparison between '*Adaptive Kriging*' and '*Kriging*', between Day 1 and Day 40, '*Adaptive Kriging*' does not perform as well as '*Kriging*'. This is because when the demand change is small *'Kriging'* still has good performance. In contrast, for '*Adaptive Kriging*', each update of the response surface is based on a limited number of samples, which may bring uncertainties in themselves due to the randomised demand, and hence affect the performance. From Day 40, however, the performance of '*Adaptive Kriging*' becomes better than that of '*Kriging*'. Moreover, with the growth of demand, the difference is even more prominent. These findings highlight the need for an adaptive response surface, and the effectiveness of Algorithm 5.1 in maintaining the signal control's performance in response to the increase of demand.

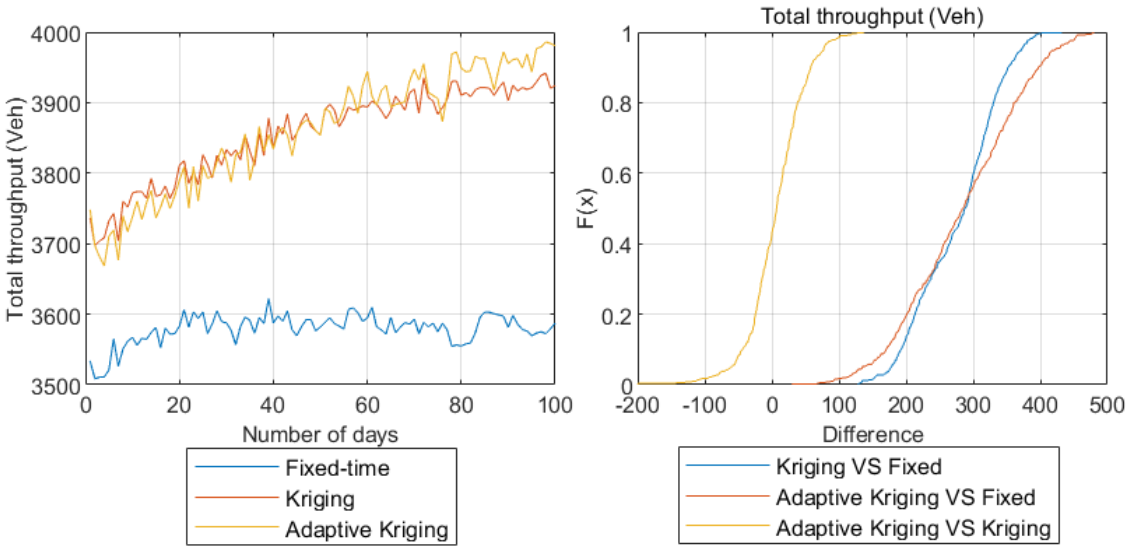


Figure 5.3 Evolution of total throughput (Left), and CDF plot of the difference in total throughput under the increase of demand

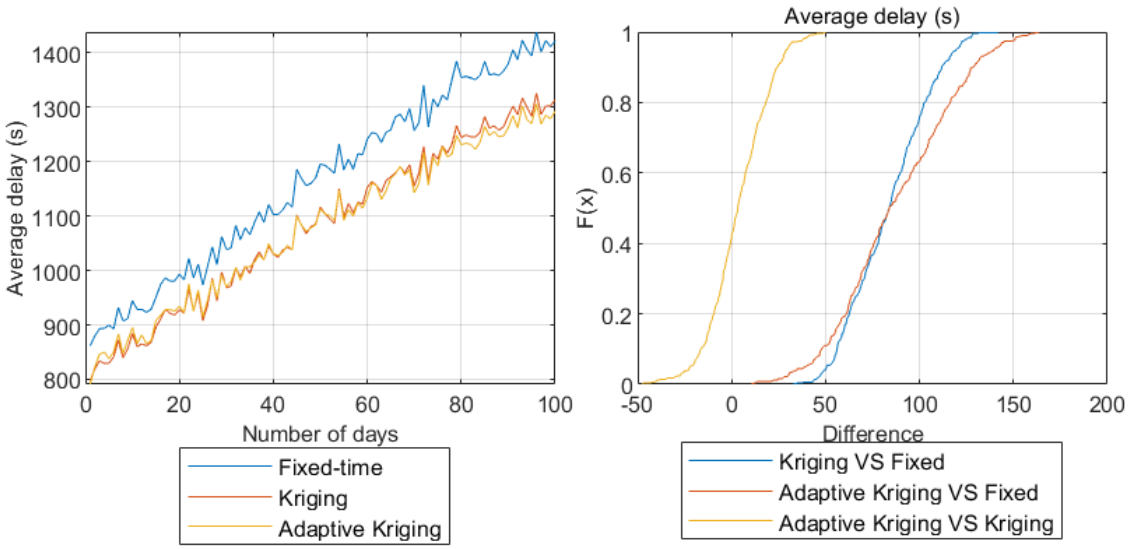


Figure 5.4 Evolution of average delay (Left), and CDF plot of the difference in average delay under the increase of demand

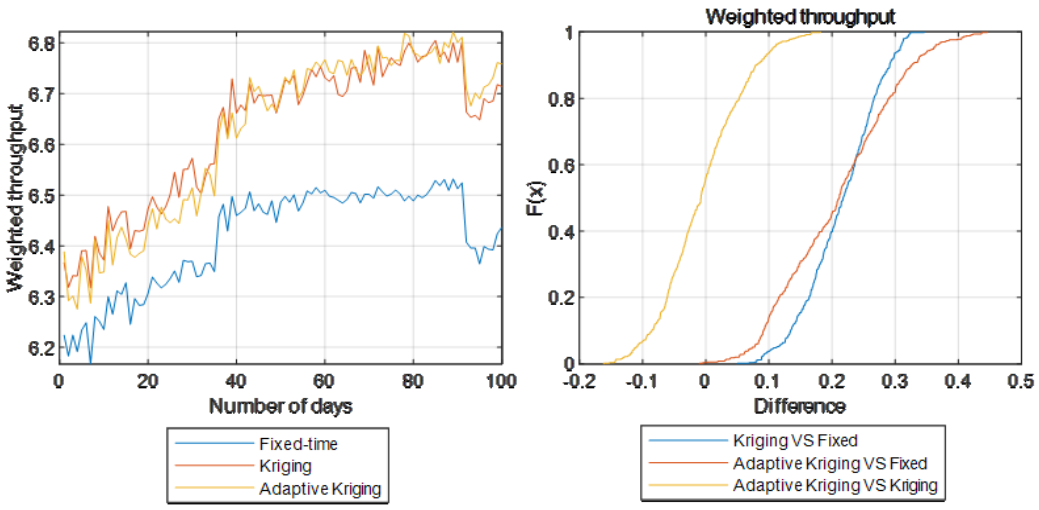


Figure 5.5 Evolution of weighted throughput (Left), and CDF plot of the difference in weighted throughput under the increase of demand

In order to investigate the factor that causes the difference between '*Adaptive Kriging*' and '*Kriging*', the approximation error (i.e. MAPE) and performance of baseline signals are plotted in Figure 5.6 and Figure 5.7. It can be seen from Figure 5.6 that the MAPE of '*Kriging*' increases over time, which indicates that the approximation accuracy of '*Kriging*' degrades. This shows that the fixed response surface cannot interpret the time-varying traffic dynamics accurately if the demand level increases. Figure 5.7 compares the performance of the initial baseline signal, which is the baseline signal of '*Kriging*', and the adaptive baseline signal of '*Adaptive Kriging*'. It is obvious from the figure that the adaptive baseline signal leads to a much higher total throughput than the fixed baseline signal. This confirms the effectiveness of the baseline update strategy.

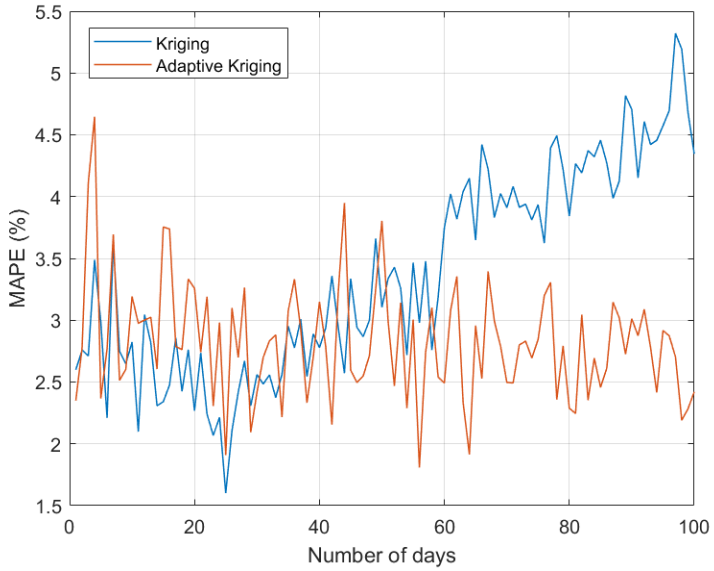


Figure 5.6 Evolution of approximation error under an increase of demand

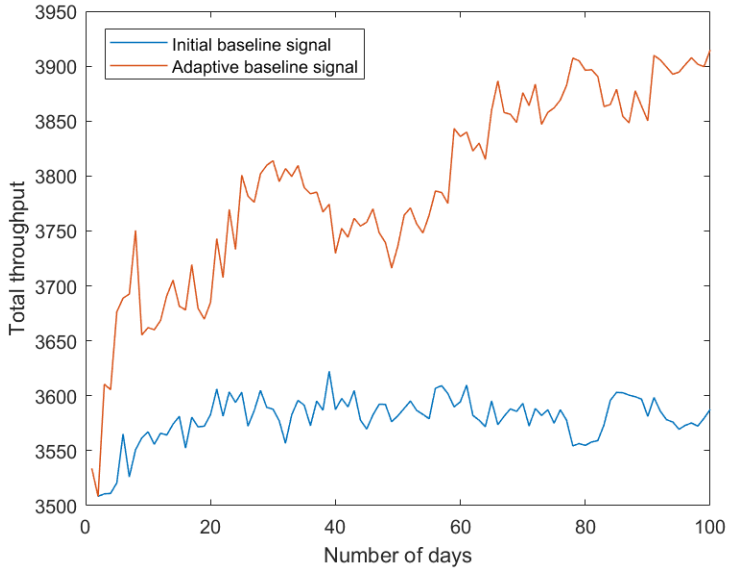


Figure 5.7 The performance of baseline signals against time under an increase of demand

Analysis of results of case 2

Figure 5.8, Figure 5.9 and Figure 5.10 show the evolution of three performance indicators and CDF plots as demand level decreases. It can be seen from the figures that, at the beginning of the test, '*Kriging*' and '*Adaptive Kriging*' have better performance than '*Fixed*'. The difference between them decreases with time, however, and all three signal controls

eventually have a similar level of performance, this is because the performance degradation of fixed-time signal controls when demand decrease is less significant than that when demand increases (Park et al., 2000). The CDF plot shows that, from an overall perspective, '*Kriging*' and '*Adaptive Kriging*' outperform '*Fixed*' since in most of the cases they lead to better performance indicators. While between '*Kriging*' and '*Adaptive Kriging*', the latter is slightly better.

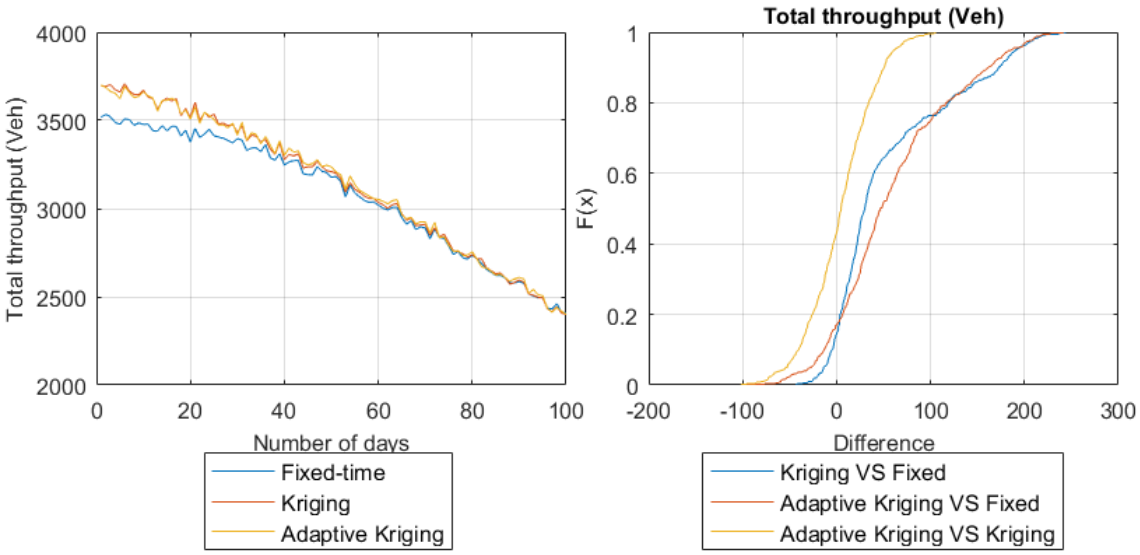


Figure 5.8 Evolution of total throughput (Left) and CDF plot of the difference in total throughput under a decrease of demand

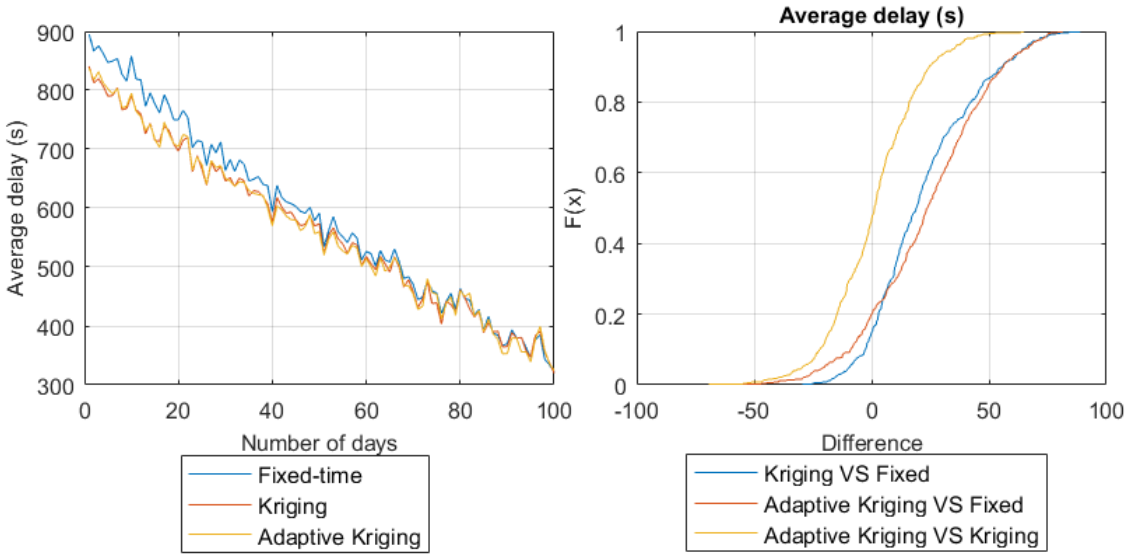


Figure 5.9 Evolution of average delay (Left) and CDF plot of the difference in average delay under a decrease of demand

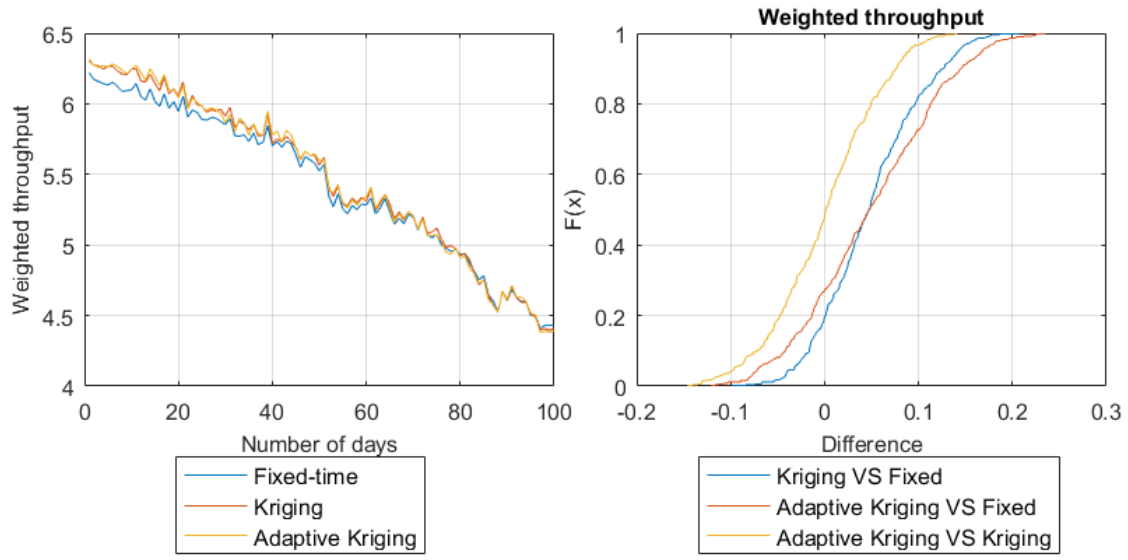


Figure 5.10 Evolution of weighted throughput (Left) and CDF plot of the difference in weighted throughput under a decrease of demand

Figure 5.11 shows the MAPE of total throughput in '*Kriging*' and '*Adaptive Kriging*'. Interestingly, the MAPE of *'Kriging'* increases over time, even reaching 45% on Day 100. Nevertheless, it can still find signal controls that are as good as, or even better than, those

generated by '*Adaptive Kriging*'. This indicates that, for a given traffic state, the factor affecting the performance of real time signal control is the similarity between the shapes of the estimated signal-objective relationship and the actual signal-objective relationship. Therefore, '*Kriging*' can still make a reasonable decision even though it cannot predict the outputs accurately. Under the initial condition (i.e. TD_0), the network is already congested, and thus the variation of total throughput is small. This means that the Kriging model constructed under TD_0 may underestimate the decline of total throughput due to decreasing demand, and this is the reason for the large estimation error of '*Kriging*'.

Moreover, the MAPE of '*Adaptive Kriging*' also increases over time. Comparing three performance indicators (i.e. total throughput, average delay and weighted throughput) on Day 1 and Day 100, a huge difference can be observed, indicating the rapid change in demand level in this experiment. Due to the low update speed, '*Adaptive Kriging*' cannot respond to the time-varying demand level in time, leading to the increase in approximation error.

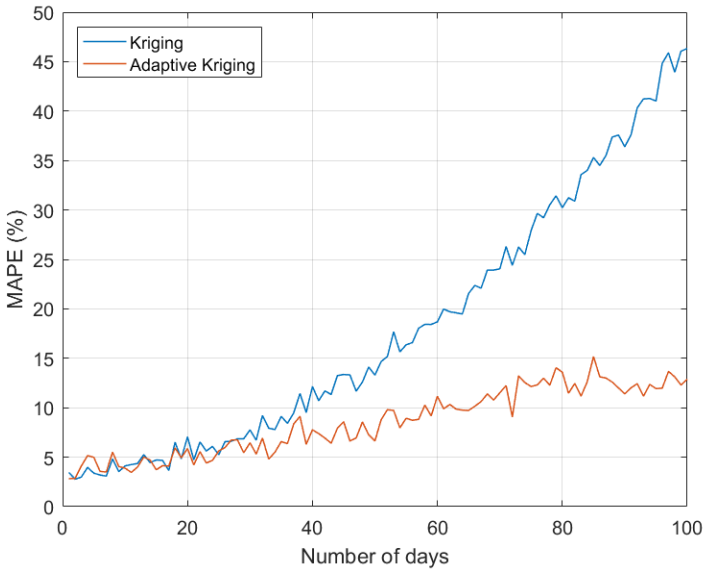


Figure 5.11 Evolution of approximation error under a decrease of demand

Figure 5.12 shows the performance of the initial and adaptive baseline signals over time. The adaptive baseline signal always leads to a larger total throughput than the initial baseline signal, which validates the effectiveness of the baseline signal update strategy.

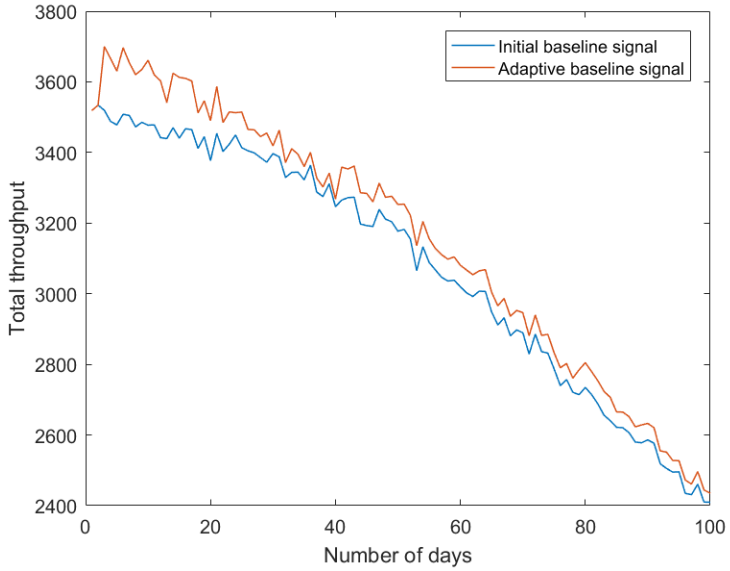


Figure 5.12 The performance of baseline signals against time under a decrease of demand

Analysis of results of case 3

For case 3, the relative demand change of each link is randomly selected from the uniform distribution between $[-0.5, 0.5]$. Figure 5.13 shows the relative change of each inflow link in the demand profile from TD_0 to TD_1. A positive value indicates an increase in demand, while a negative value indicates the reverse.

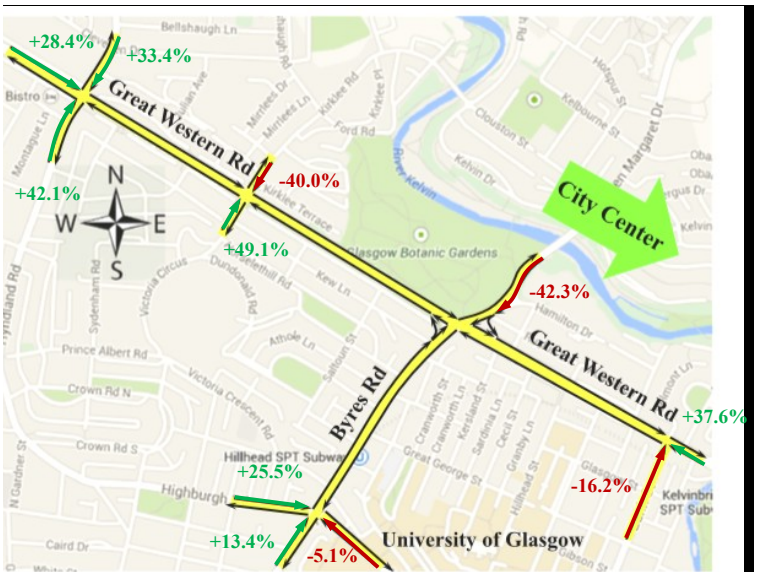


Figure 5.13 Illustration of links with increasing demand (Red) and decreasing demand (Green) (Google Maps)

Figure 5.14, Figure 5.15 and Figure 5.16 compare the average performance of the three signal control strategies. It can be seen that from Day 40, the total throughput of '*Kriging*' begins to decline, while the average delay is still increasing. Theoretically, when no deterioration exists, the total throughput and the average delay should show the same trend as the demand change, as illustrated in Table 5.4. The opposite evolution of the trends between total throughput and average delay indicates that '*Kriging*' suffers from performance deterioration when the change of demand is irregular. The reason that *'Adaptive Kriging'* also deteriorates is that the change in traffic demand is rapid, and the irregular change in demand poses a challenge. *'Adaptive Kriging*' therefore does not have enough samples and time to adapt to the change effectively.

'*Kriging*' and '*Adaptive Kriging*' outperform '*Fixed*' in all three of the performance indicators. This shows that real time signal control can accommodate the irregular change of demand better than the fixed-time signal plan, although the difference between them reduces with time since the two real time signal controls deteriorate.

Additionally, '*Adaptive Kriging*' performs even better than '*Kriging*'. The irregular change of demand leads the traffic network to behave differently and thus establishes a new relationship between traffic states, signal control and network performance. As stated in the previous section, '*Kriging*' cannot arrive at a proper decision when the shape of the actual relationship has changed, since it can no longer capture the network behaviour of the upcoming control period. This means that the Kriging model needs to be updated regularly to adapt to the new relationship. It is also noted that, in the first 40 epochs, '*Kriging*' and '*Adaptive Kriging*' have similar performance. This is because each update in '*Adaptive Kriging*' is only based on limited samples, which may bring uncertainties in themselves due to randomised demands. Hence, the response surface requires a certain number of epochs to accumulate a statistically significant sample set to capture the systematic change. From Day 40, however, '*Adaptive Kriging*' becomes superior to '*Kriging*' and the difference increases over time.

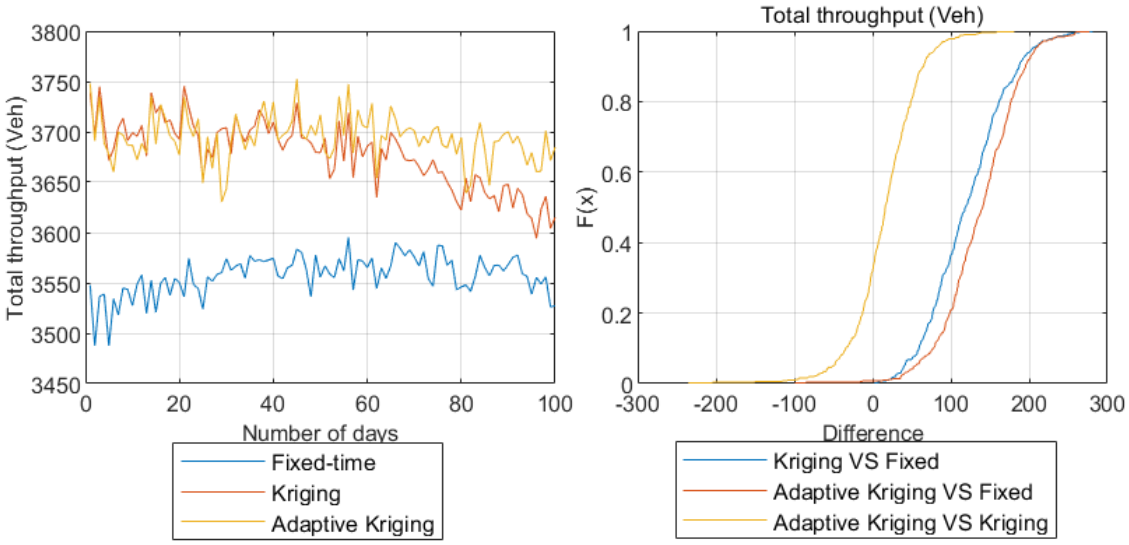


Figure 5.14 Evolution of total throughput (Left) and CDF plot of the difference in total throughput under an irregular change of demand

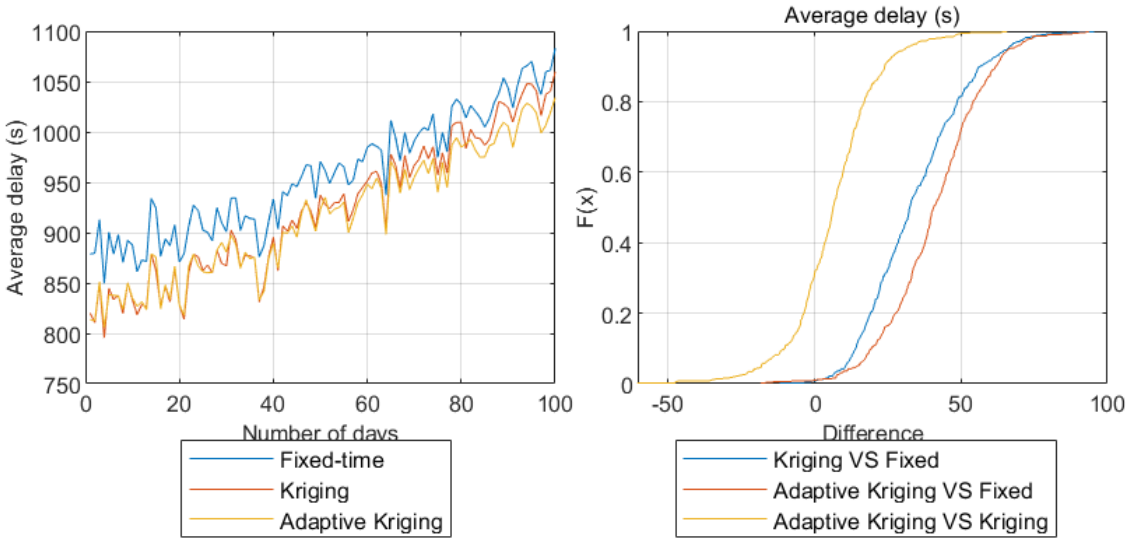


Figure 5.15 Evolution of average delay (Left) and CDF plot of the difference in average delay under an irregular change of demand

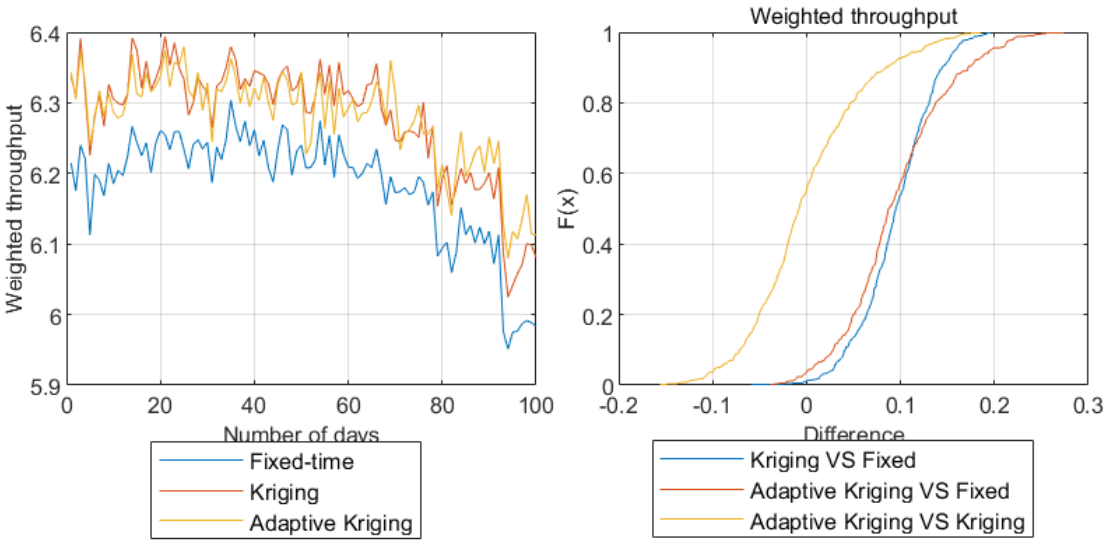


Figure 5.16 Evolution of weighted throughput (Left) and CDF plot of the difference in weighted throughput under an irregular change of demand

Figure 5.17 indicates that '*Kriging*' has a lower MAPE than '*Adaptive Kriging*' between Day 20 and Day 80. Under this condition, therefore, the efficacy of the adaptive response surface in terms of an improvement in approximation accuracy cannot be validated. The adaptive baseline signal leads to larger total throughput than the initial baseline signal from Day 40, as shown in Figure 5.18, however, hence showing the effectiveness of the baseline signal update strategy.

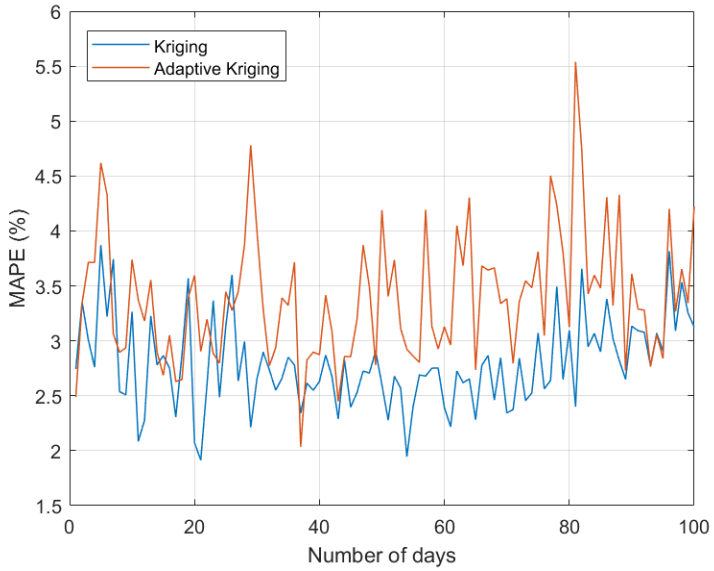


Figure 5.17 Evolution of approximation error under an irregular change of demand

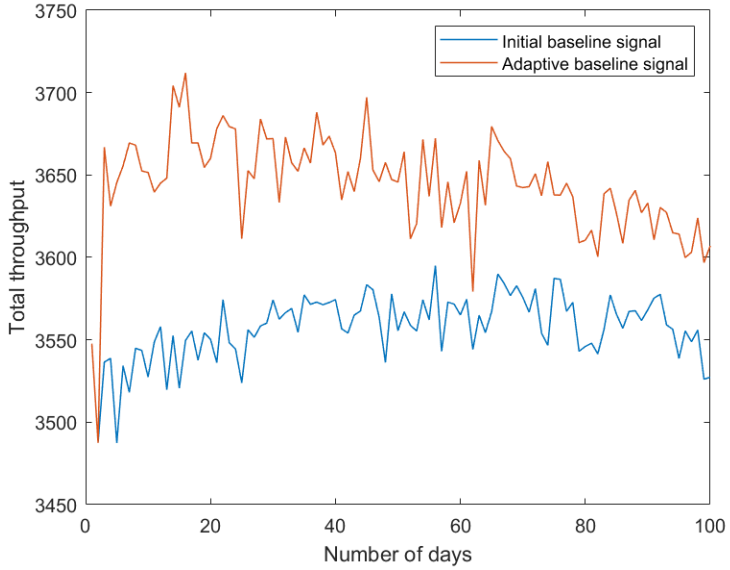


Figure 5.18 The performance of baseline signals against time under an irregular change of demand

5.2.4 Sensitivity analysis

The previous section has shown the effectiveness of '*Adaptive Kriging*' in maintaining network performance under both an increasing demand and an irregular change in demand. This section further explores how the update plan affects the performance of '*Adaptive Kriging*' when demand increases over time by 50%. Update frequency and the number of

infill points are two major factors controlling the adaptability of the response surface. Four update plans are therefore designed with different update frequencies and numbers of infill points. These are summarised as follows:

- **Case 1**: Update every two days and replace ten points each time.
- **Case 2**: Update every day and replace five points each time.
- **Case 3**: Update every two days and replace five points each time.
- **Case 4**: Update every four days and replace ten points each time.

Thirteen replications are conducted for each case. In order to investigate how the update frequency and the number of infill points affect the adaptability of the real time signal control to the increase in demand, the first two sensitivity analyses are designed. In sensitivity analysis 3, a new term, '*update speed*' is introduced, defined as the average number of points updated per day. With the same '*update speed*', the Case with the greater impact on the performance of the proposed signal control between update frequency and the number of infill points can be identified. The results and analysis of each test are discussed in detail in the following part.

- **Sensitivity analysis 1:** Compares Case 2 & Case 3 and Case 1 & Case 4 to investigate the influence of update frequency when the number of infill points is fixed.

- **Sensitivity analysis 2:** Compares Case 1 & Case 3 to investigate the influence of the number of infill points when update frequency is fixed.

- **Sensitivity analysis 3:** Compares Case 1 & Case 2 and Case 3 & Case 4 to identify the one with the greater impact between update frequency and the number of infill points when '*update speed*' is fixed.

Sensitivity analysis 1

The first sensitivity test focuses on the influence of update frequency. Two pairs of update plans are compared: Case 2 vs Case 3 and Case 1 vs Case 4. The two plans in each pair have

the same number of infill points, but differ in their update frequency. Case 1 and Case 2 update more frequently than the corresponding plan in the same pair.

Figure 5.19 and Figure 5.20 show the evolution of total throughput and average delay when the network is controlled by '*Adaptive Kriging*' with an update strategy following Case 2 and Case 3. It can be seen that Case 2 leads to a larger total throughput and smaller average delay than Case 3. Figure 5.21 and Figure 5.22 present the comparison between Case 1 and Case 4 in terms of total throughput and average delay; from which it can be seen that Case 1 has better performance than Case 4. It can therefore be concluded that the plan with a higher update frequency leads to better performance.

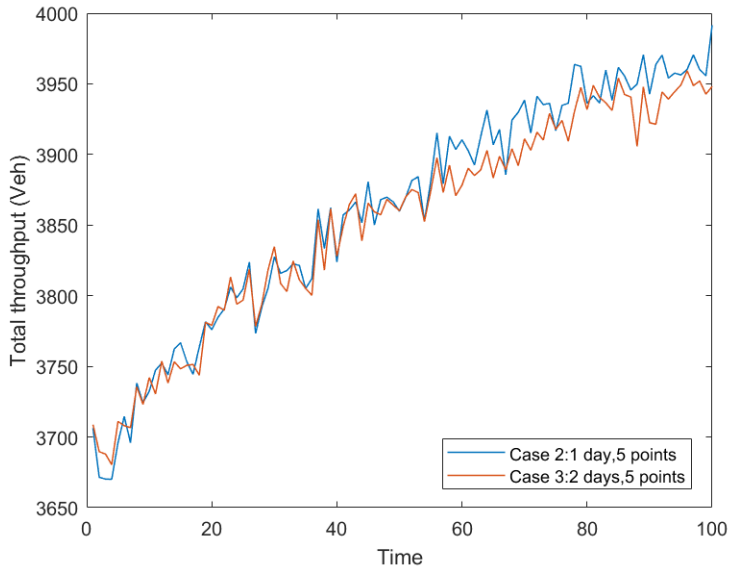


Figure 5.19 Comparison between Case 2 and Case 3 in terms of total throughput under a systematic change of demand

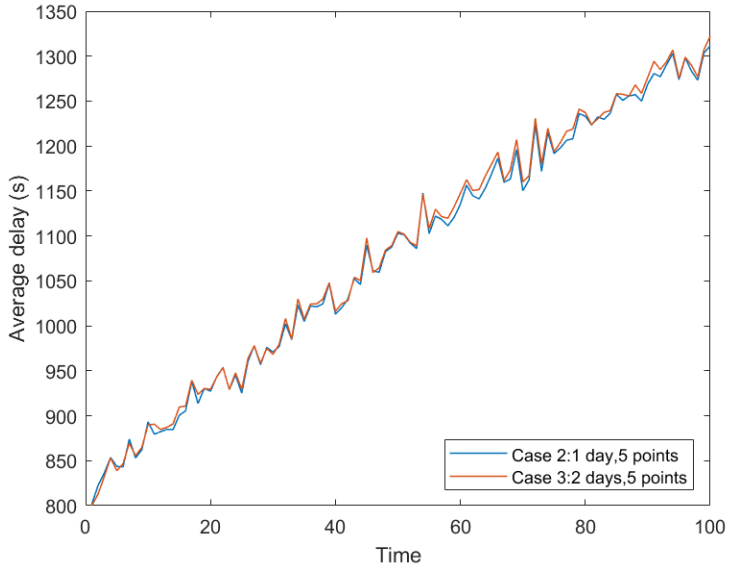


Figure 5.20 Comparison between Case 2 and Case 3 in terms of average delay under a systematic change of demand

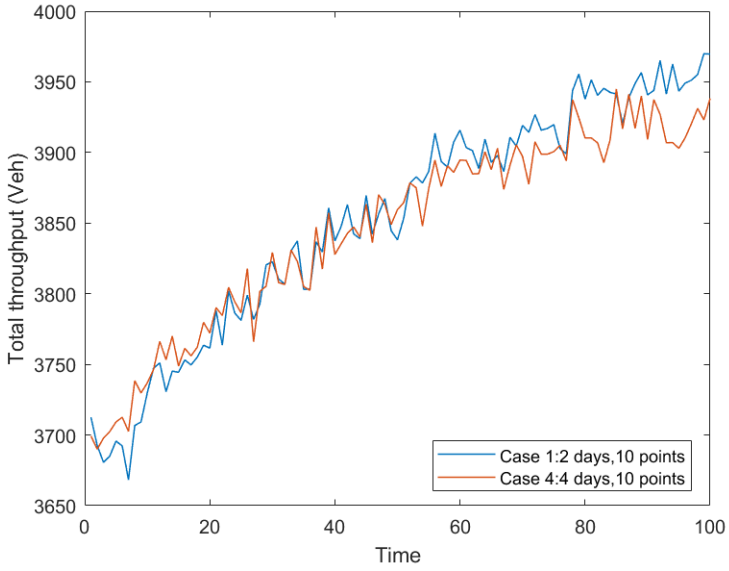


Figure 5.21 Comparison between Case 1 and Case 4 in terms of total throughput under a systematic change of demand

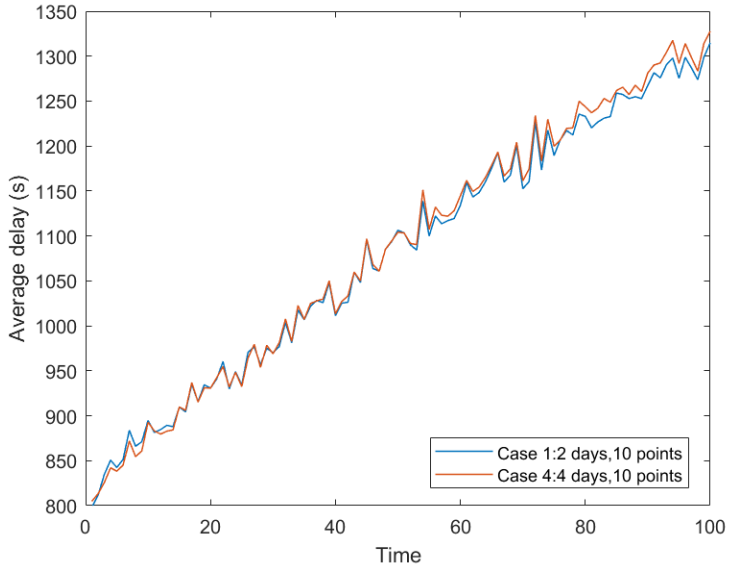


Figure 5.22 Comparison between Case 1 and Case 4 in terms of average delay under a systematic change of demand

In order to investigate the influence of the update frequency further, the differences in total throughput and average delay of each pair of plans are calculated, and CDF plots of them are plotted, as shown in Figure 5.23. If the former plan has better performance than the latter one (i.e. larger total throughput or smaller average delay), the difference is positive, and negative for the reverse. CDF plots help to identify the probability that one is superior to the other by observing the F(x) value when the difference is zero. The blue curve represents the comparison between Case 2 and Case 3, and the red curve represents the comparison between Case 1 and Case 4. For total throughput, both curves have an F(x) value less than 0.5 when the difference is zero, such that Case 2 and Case 1 have a higher probability of having larger total throughput than Case 3 and Case 4 under the same condition. There is a similar trend for the average delay, where the asymmetric curve confirms that the cases with higher update frequency have a larger probability of leading to a lower average delay.

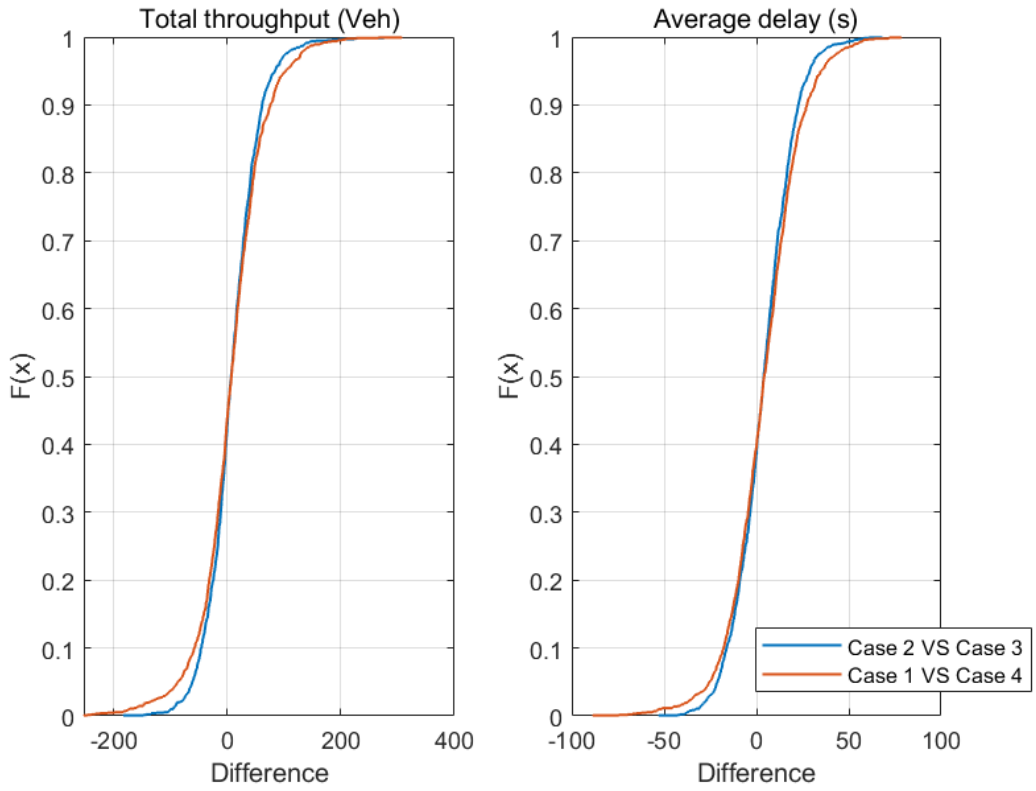


Figure 5.23 Empirical CDF plot of differences in total throughput (Left) and average delay (Right) between Case 2 & Case 3 and Case 1 & Case 4 (positive means the former is better than the latter)

In each simulation, 100 network performance indicators were collected. Therefore 1300 samples were collected after 13 simulations. The paired t-test can show whether the difference between the means of the two populations is significantly different. The *ttest* function in MATLAB was used to calculate this. For two paired populations with mean μ_a and μ_b respectively, the null hypothesis test is set to be:

$$H_0: \mu_a = \mu_b \tag{5.1}$$

While the alternative hypothesis is:

$$H_1: \mu_a \neq \mu_b \tag{5.2}$$

Whether the null hypothesis test is rejected is determined by the p-value. Give the 95%

confidence interval, when the p-value is smaller than 0.05, then the null hypothesis is rejected and the difference between the means of two populations is determined to be statistically significant. When it is proved that the two populations are different, then a t-test with Left tail or Right tail is carried out to examine which one is better.

The null hypothesis and alternative hypothesis of Right-tail t-test are:

$$H_0: \mu_a \leq \mu_b \tag{5.3}$$

$$H_1: \mu_a > \mu_b \tag{5.4}$$

The null hypothesis and alternative hypothesis of the Left-tail t-test are:

$$H_0: \mu_a \geq \mu_b \tag{5.5}$$

$$H_1: \mu_a < \mu_b \tag{5.6}$$

Similarly, when the p-value is smaller than 0.05, the alternative hypothesis is accepted.

		p-value	Reject/Uphold
Total throughput	Case 2 vs Case 3	1.7853×10^{-11}	Reject
	Case 2 vs Case 3 (Right Tail)	8.9265×10^{-12}	Reject
	Case 1 vs Case 4	2.0231×10^{-5}	Reject
	Case 1 vs Case 4 (Right Tail)	1.0115×10^{-5}	Reject
Average delay	Case 2 vs Case 3	1.6235×10^{-19}	Reject
	Case 2 vs Case 3	8.1173×10^{-20}	Reject

	(Left Tail)		
	Case 1 vs Case 4	1.2468×10^{-15}	Reject
	Case 1 vs Case 4 (Left Tail)	6.2338×10^{-16}	Reject

Table 5.5 Hypothesis test results of Case 2 vs Case 3 and Case 1 vs Case 4

Based on the hypothesis results presented in Table 5.5, it can be seen that the differences between each pair in both total throughput and average delay are statistically significant. This confirms the finding that there is an improvement in network performance when the update frequency increases.

In summary, the update frequency can affect the performance of the proposed real time signal control, especially when the number of infill points is large.

Sensitivity analysis 2

Test 2 focuses on the influence of the number of infill points on the adaptability of the proposed real time signal control under a systematic change of demand. Case 1 and Case 3 with the same update frequency (i.e. every two days) are compared. The former updates five points each time while the latter updates ten points. Figure 5.24 and Figure 5.25 show that Case 1 and Case 3 have similar performance in terms of the total throughput and average delay. The difference in total throughput and average delay is plotted in Figure 5.26. The subplot of average delay shows that the probability that Case 3 has a lower average delay than Case 1 is greater than 50%.

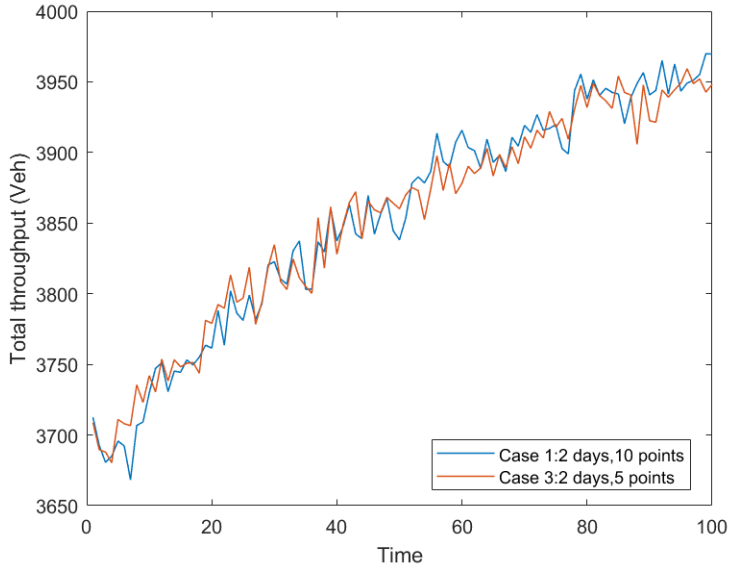


Figure 5.24 Comparison between Case 1 and Case 3 in terms of total throughput under a systematic change of demand

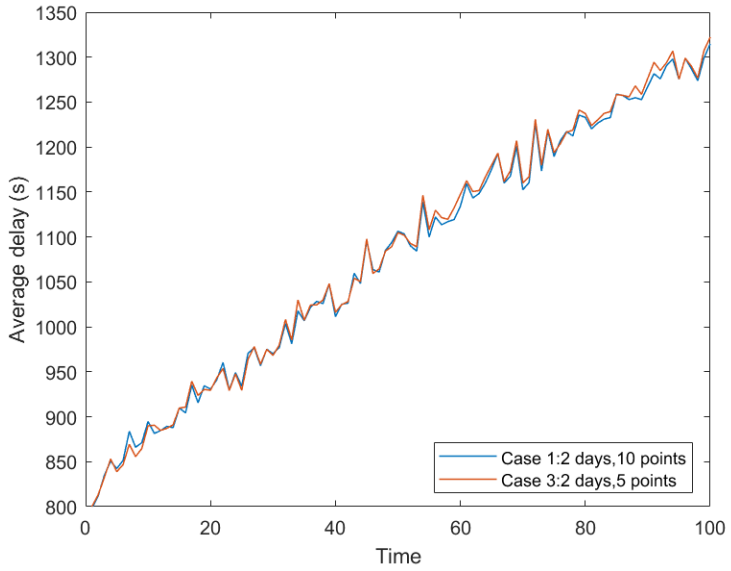


Figure 5.25 Comparison between Case 1 and Case 3 in terms of average delay under a systematic change of demand

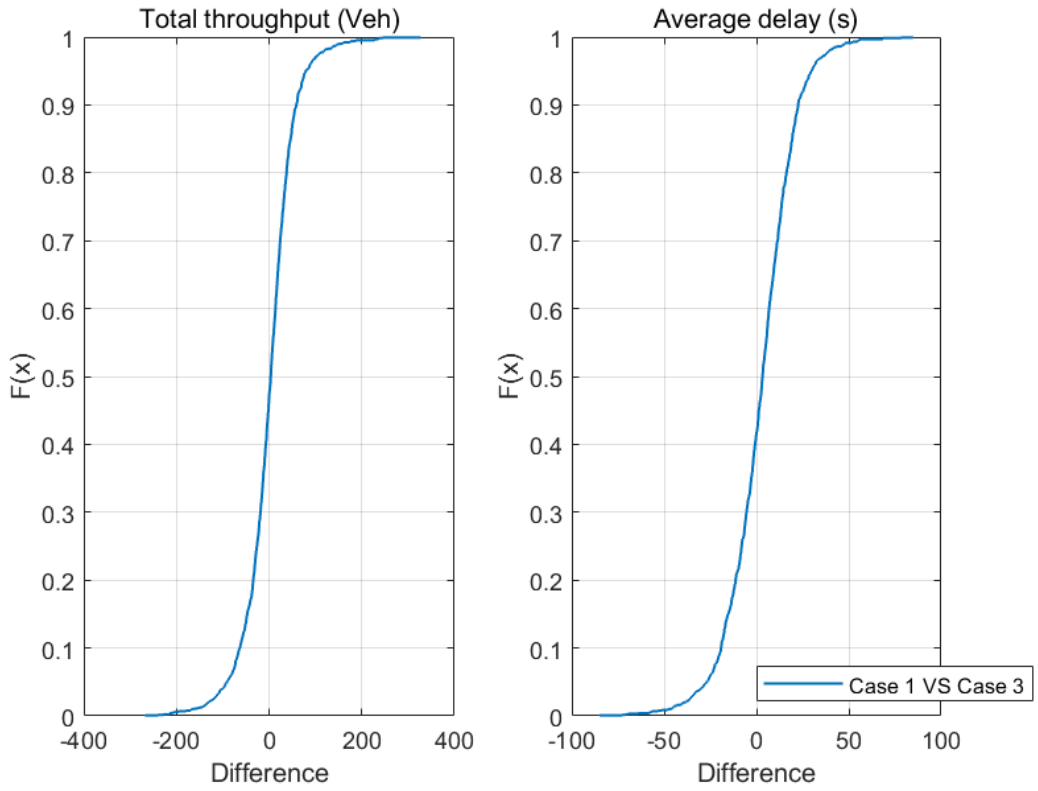


Figure 5.26 Empirical CDF plot of differences in total throughput (Left) and average delay (Right) between Case 1 & Case 3 (positive means the former is better than the latter)

The t-test results in Table 5.6 also show that there are no statistically significant differences in the total throughput, whereas the average delay of Case 3 is statistically smaller than that of Case 1. As a consequence, the performance can be improved if a larger number of points are replaced each time, but in this test, the improvement is small.

		p-value	Reject/Uphold
Total throughput	Case 1 vs Case 3	0.5425	Uphold
Average delay	Case 1 vs Case 3	1.3718×10^{-6}	Reject
	Case 1 vs Case 3 (Left Tail)	6.8591×10^{-7}	Reject

Table 5.6 Hypothesis test results of Case 1 vs Case 3

Sensitivity analysis 3

The previous two analyses have shown that both the update frequency and the number of

infill points can affect the adaptability of the proposed real time signal control. It is worth noting, however, that for each pair of tests, the '*update speed*' is not the same. This sensitivity analysis was therefore conducted under a fixed '*update speed*' in order to investigate whether update frequency or the number of infill points has the greater influence on the adaptability of the proposed real time signal control.

Figure 5.27 and Figure 5.28 present the evolution of total throughput and average delay over time when the network is controlled under Case 1 and Case 2. Both cases add 500 new points to the surrogate model in total within 100 days, with the update speed being five points per day. With the same update speed, Case 2 outperforms Case 1. Similarly, in Figure 5.29 and Figure 5.30, it can be seen that Case 3 leads to better network performance than Case 4. This confirms the finding that the update frequency is the main factor affecting the performance of the proposed real time signal control, compared to the number of infill points. Moreover, the difference between Case 3 and Case 4 is greater than that between Case 1 and Case 2, which indicates that the performance of the proposed real time signal control is more sensitive to the design of the update plan when the update speed is low.

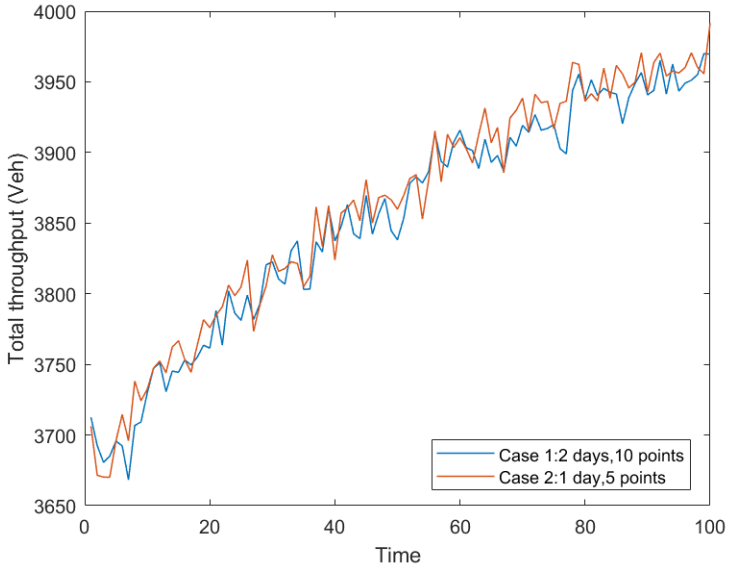


Figure 5.27 Comparison between Case 1 and Case 2 in terms of total throughput under a systematic change of demand

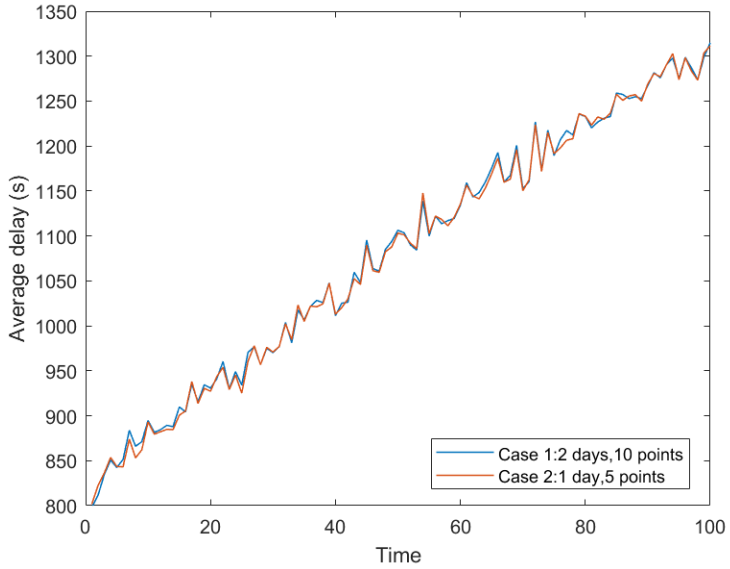


Figure 5.28 Comparison between Case 1 and Case 2 in terms of average delay under a systematic change of demand

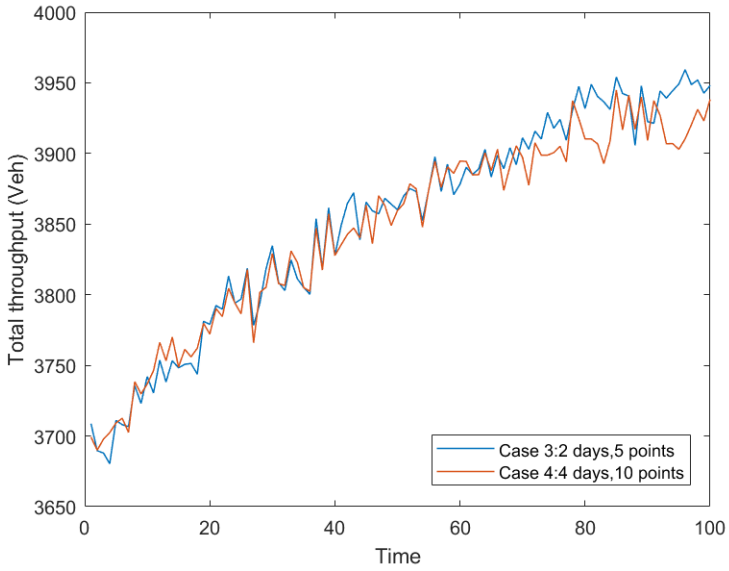


Figure 5.29 Comparison between Case 3 and Case 4 in terms of total throughput under a systematic change of demand

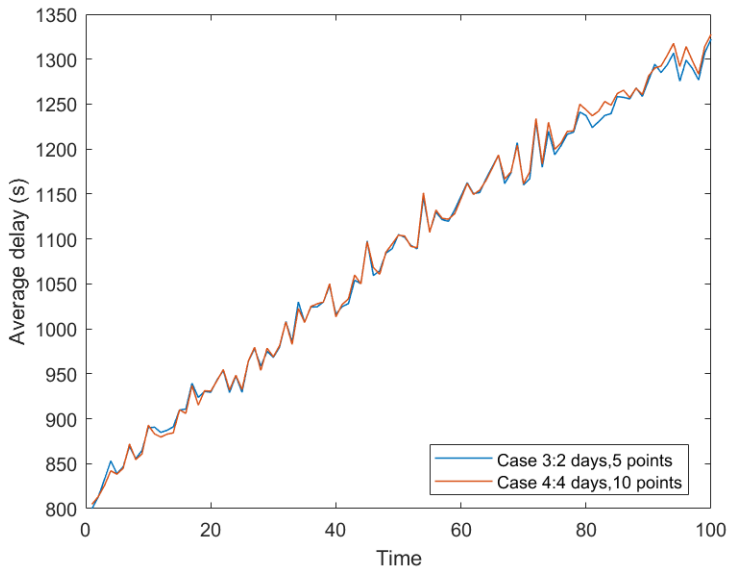


Figure 5.30 Comparison between Case 3 and Case 4 in terms of average delay under a systematic change of demand

The CDF plots in Figure 5.31 show that the probabilities that Case 2 and Case 3 lead to larger total throughput and smaller average delay than Case 1 and Case 4 are higher than 50%. In addition, the t-test results in Table 5.7 show that this difference is statistically significant.

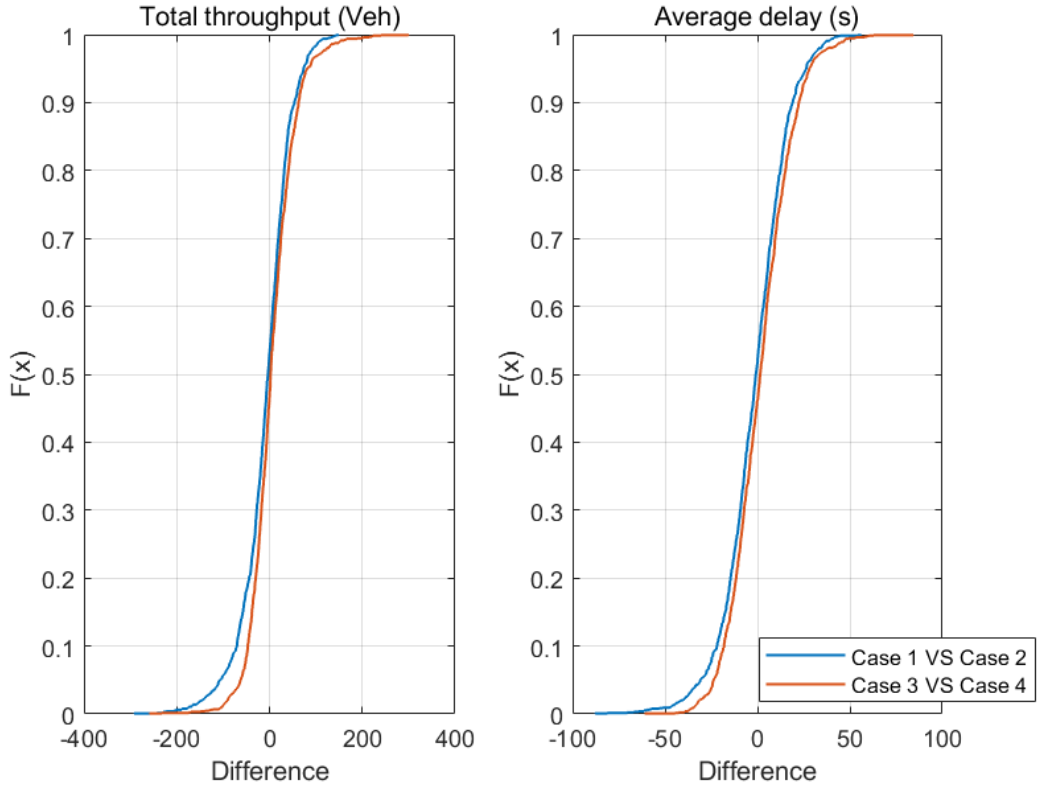


Figure 5.31 Empirical CDF plot of differences in total throughput (Left) and average delay (Right) between Case 1 & Case 2 and Case 3 & Case 4 (positive means the former is better than the latter)

		p-value	Reject/Uphold
Total throughput	Case 1 vs Case 2	2.5944×10^{-7}	Reject
	Case 1 vs Case 2 (Left Tail)	1.2972×10^{-7}	Reject
	Case 3 vs Case 4	7.5154×10^{-6}	Reject
	Case 3 vs Case 4 (Right Tail)	3.7577×10^{-6}	Reject
Average delay	Case 1 vs Case 2	0.0019	Reject
	Case 1 vs Case 2 (Right Tail)	9.4496×10^{-4}	Reject
	Case 3 vs Case 4	4.3515×10^{-5}	Reject
	Case 3 vs Case 4 (Left Tail)	2.1757×10^{-5}	Reject

Table 5.7 Hypothesis test results of Case 1 vs Case 2 and Case 3 vs Case 4

5.3 Conclusions

This chapter has proposed an off-line adaptive baseline signal update strategy to protect the surrogate-based real time signal control from performance degradation due to an inappropriate baseline signal. In the proposed strategy, the baseline signal was updated based on the optimal signals generated during the optimisation process. Test results showed that the performance of the real time signal control was improved by iterative updating. The difference between three surrogate-based real time signal controls with different initial baseline signals decreased, which indicates that the off-line adaptive updates can reduce the influence of the initial baseline signal effectively.

Furthermore, the chapter developed a surrogate-based real time signal control with an adaptive response surface. Comparing the performance between the real time signal controls with fixed and adaptive response surfaces in conditions of increased, decreased and irregular changes of demand, the latter was shown to have better performance than the former when

demand level increased or changed irregularly. Under the decreasing demand level, both of them had a similar level of performance eventually. This indicated that the adaptive response surface is effective in adapting to non-stationary demands, and its efficacy in helping the surrogate-based real time signal control to retain the desired level of performance under the demand change was confirmed.

The sensitivity analyses on the design of the update plan showed that the adaptability and performance of a surrogate-based real time signal control system can be improved if it is updated more frequently or replacing more points each time. The update frequency, however, is the more dominant of these two factors.

Chapter 6

Conclusion and future work

Signal control is an effective way to manage traffic flow. Compared with fixed-time signal control, real time signal control is more effective in accommodating unpredictable traffic conditions. As a basic requirement, real time signal control should be efficient enough to make decisions within seconds with satisfactory optimality. Moreover, it needs to be robust to various traffic conditions and have the ability to adapt to the evolution of the traffic network. In order to ensure flexibility and transferability, few assumptions and restrictions should be posed in respect to the form of traffic states, control variables and performance indicators. These requirements are satisfied by a number of objectives which are revisited in Section 6.1. Section 6.2 summarises the contributions of this thesis. Finally, Section 6.3 presents a set of recommendations for future research.

6.1 Conclusions

This thesis aimed to propose a centralised model-free real time signal control, based on surrogate models, which can self-update so as to adapt to systematic changes in demand. In order to achieve this aim, seven objectives were defined in Section 1.2. The findings and contributions related to these objectives are discussed in turn in the following section.

1. Review the state-of-the-art of real time signal controls.

Six of the most widely-used real time signal controls systems were reviewed in Chapter 2. All of these are model-based and therefore lack the flexibility to accommodate user-defined input variables and objectives. Moreover, due to their computational burden, they are either non-optimised or operated in a distributed or hierarchical manner.

The optimisation methods of real time signal controls were classified into the exact

approach, heuristic approach, rule-based approach and the learning-based approach in Chapter 2. The basic features of the literature reviewed in Chapter 2 are summarised in Appendix A.

The exact approach and heuristic approach need to be integrated with a traffic model to estimate the performance of different signal control decisions, such that their real-world applications are dependent on the validity and accuracy of the underlying traffic model. The rule-based approach is model-free, but is non-optimised. The RL-based approach is model-free, but it is not interpretable and is usually operated in a distributed manner. Although real time signal control based on decision rules is time efficient, since the time-consuming optimisation is carried out off-line, it still requires a traffic model for off-line optimisation.

In the literature reviewed, there is no model-free real time signal control operated in a centralised manner.

2. ***Review the state-of -the-art of surrogate modelling.***

Three surrogate models were reviewed in Chapter 2. The selection of the appropriate surrogate model depends on the complexity of the relationship that needs to be approximated. Moreover, the construction of the surrogate model is a trade-off between approximation accuracy and efficiency.

This review of surrogate-based traffic control problems showed that such problems are capable of transferability since their construction does not need any background knowledge of the underlying problem. Surrogate models have been applied to solve a wide range of problems, and their use can reduce the computational burden, especially for time-consuming optimisation problems. It is the first that review both real time signal control and surrogate modelling in one literature.

3. ***Develop a framework for surrogate modelling that can interpret the state-control-objective relationship in an accurate and robust way.***

Chapter 3 identified that surrogate models are currently only applied to fixed-time control problems. The chapter therefore developed a surrogate model that can be applied to real time optimisation problems. In order to enable real time decisions, the notion of a response surface was extended by incorporating the traffic condition as state variables, and constructing surrogate models that interpret and approximate the state-control-objective relationship.

In order to ensure the estimation efficiency of those surrogate models, the traffic state variables and signal control variables were sampled around the given baseline scenario, respectively. It was shown that the surrogate model can be improved through an infill strategy, which re-trains the model with an additional set of sample points. A multi-point candidate sampling approach was developed to select the infill points with the largest estimation error from a set of candidate points. In addition, the distances between the selected infill points were controlled to ensure the diversity of samples.

The numerical studies in Section 3.4 showed that the Kriging model with 451 training points has satisfactory approximation accuracy, with a testing error (MAPE) less than 3.5%. Furthermore, it can make the estimation within 1.9432×10^{-4} s.

4. ***Develop a surrogate-based real time signal control system based on the surrogate model proposed in objective three.***

Chapter 4 developed a surrogate-based real time signal control with the surrogate model proposed in Chapter 3. The optimisation problem was solved by a modified Particle Swarm Optimisation (PSO). In order to ensure optimisation efficiency, the initial particles of the PSO were sampled around the baseline signal. The inertia parameter was decreased over iterations so as to let the local search dominate gradually.

The fitness of PSO is the combination of the objective value and the estimation error, and thus the inherent estimation error was considered in the optimisation process. The balance between optimality and reliability was controlled by a weighting parameter.

Section 4.5.4 shows that the real time signal control with a large weighting parameter was reliable in cases where the surrogate model was not accurate enough.

5. ***Test, validate and assess the surrogate-based real time signal control, and compare it with the benchmarks***.

The proposed surrogate-based real time signal control was tested in Chapter 4. The test results showed that, compared with the fixed-time baseline signal, the proposed real time signal control increased the total throughput and weighted throughput by 5.3% and 2.2% respectively, while reducing the average delay by 8.1%. Furthermore, it could make decisions in just eight seconds, which reduces around 50% and 90% time taken by GA optimiser and simulation-based optimiser respectively.

In addition, the robustness of the proposed real time signal control was validated under scenarios with a variable number of state variables, demand levels and demand variabilities. The choice of state variables is vital for the performance of the surrogate model, and the Kriging model is capable of identifying important input state variables via its inherent mathematical structure. This helped to maintain, or even enhance, the performance of the surrogate model by actively selecting the input state variables. Furthermore, with different levels of saturation of the network and variability of the uncertain traffic demand, the proposed surrogate-based real time signal control outperformed the benchmark signal controls consistently.

6. ***Develop an adaptive baseline signal update strategy to mitigate the influence of the baseline signal.***

In Chapter 4, it was found that the initial baseline signal affects the performance of surrogate-based real time signal control. In order to eliminate the influence of the initial baseline signal, an adaptive baseline signal update strategy was developed in Chapter 5. By applying the surrogate-based real time signal control, the surrogate model can be updated based on the signal decisions made in the optimisation process. Numerical

studies showed that, with the adaptive update of the underlying surrogate model, the performance of surrogate-based real time signal control was significantly improved, for all the baseline signal controls. More importantly, the update process was able to bring the controller to the same level of performance regardless of the baselines chosen at the beginning of the construction of the surrogate.

7. ***Develop a surrogate-based real time signal control with an adaptive response surface.***

Chapter 5 summarised the factors that may cause performance degradation of the surrogate-based real time signal control under a systematic change of demand. The proposed surrogate-based real time signal control in Chapter 5 was thus extended to include an adaptive response surface mechanism to accommodate systematic changes of demand. The surrogate model and baseline signal were updated according to the feedback collected during the real time deployment of the controller. Numerical experiments showed that the real time signal control with adaptive response surface can maintain its performance better than the one with a fixed response surface when the demand changes systematically. In addition, it was found that the update frequency is the main factor affecting the performance of the adaptive surrogate-based real time signal control.

6.2 Contributions of the research

This thesis makes the following contributions to the literature:

- This work is the first to apply surrogate modelling to the real time control problem with both state and control as input variables. Compared with the surrogate model for a strategic design problem, key challenges lie in the inclusion of traffic state variables, which complicates the construction of the surrogate model and estimation of outputs. The test results show that non-uniform sampling following the probability distribution of the traffic state can reduce the number of training points while at the same time ensuring the approximation accuracy.

- A multi-point infill strategy is proposed in this thesis. This infill strategy is the first to adopt an adaptive distance threshold to avoid the clustering of infill points.

- This research develops a new algorithm to solve surrogate-based optimisation problems. It modifies the Particle Swarm Optimisation to take the estimation error of surrogate modelling into account, thus enabling the trade-off between optimality and reliability to be controlled. Moreover, non-uniform sampling of initial swarm particles is adopted to enhance computational efficiency. Since PSO has no requirements as to the properties of the problem, this new algorithm can be used to solve all kinds of surrogate-based optimisation problems that have a strict time constraint.

- Based on the surrogate modelling technique and PSO-based optimisation algorithm that have been developed in this work, a novel surrogate-based real time signal control system is proposed. With the surrogate model, this system is purely model-free with no need of *a posteriori* validation. It can control multiple intersections in a centralised manner to manage the network-wide performance, and generate decision plans for all the intersections simultaneously within eight seconds. Moreover, the proposed system is flexible to be applied to various traffic networks since few assumptions are imposed on it and the parameters of the surrogate model help to gain insight into the traffic dynamics, which makes the proposed system interpretable.

- This thesis proposes a surrogate-based real time signal control strategy with an adaptive response surface, and where the surrogate model and baseline signal can be updated continuously during the implementation according to feedback from the traffic network.

6.3 Recommendations for future work

This research is the first attempt to apply the surrogate model to real time signal control. There are, however, a number of limitations to this research, and thus the following recommendations are made for future research:

- The surrogate model in Chapter 3 is constructed without considering the uncertainty of data. Moreover, training points are obtained from a deterministic simulation model. In practice, however, the traffic network is stochastic, and the data quality may not be high. Further research can investigate the construction of a surrogate model with data that are subject to uncertainties and assess the robustness of the surrogate model to data quality.

- The proposed real time signal control is only tested in a deterministic macroscopic simulation environment, which has a low level of similarity to the real network. While this can act as a benchmark, further research can test the proposed signal control system in a stochastic microscopic simulation environment and modify the surrogate-based optimisation to adapt to the stochasticity of the traffic network.

- The proposed surrogate-based real time signal control system is only tested on a small network with five signalised intersections. Although, theoretically, it can be extended to larger networks, the growth in the dimension of the response surface will result in an exponential increase in computational burden. One possible approach to tackle this problem is to divide the network into several subsystems and operate traffic signals in a hierarchical manner. This can be investigated in detail in future work.

- For Kriging and RBF, the estimation time can be further reduced by constructing a correlation matrix between the point to estimate and only those training points that are close to it. Based on this, the limitation in the number of training points can be relaxed, such that the surrogate model can cover a larger design space.

- The proposed real time signal control only considers the adjustment of the green split. Other signal control parameters can be considered as well in future research.

- This research only considers a systematic change of demand. In reality, however, travel behaviours (e.g. turning ratio, wave speed) and network parameters (e.g. capacity of the link) also vary with time. The research can be further extended to assess the performance of the real time signal control system under changes in other parameters.

References

Abdoos, M., Mozayani, N. & Bazzan, A. (2013) Holonic multi-agent system for traffic signals control. *Engineering of artificial intelligence.* 26(5-6), 1575-1587.

Abdulhai, B., Pringle, R. & Karakoulas, G. (2003), Reinforcement learning for true adaptive traffic signal control. J*ournal of transportation engineering.* 129(3), 278-285.

Aboudolas, K., Papageorgiou, M., & Kosmatopoulos, E. (2009). Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C Emerging Technologies, 17*(2), 163-174.

Aslani, M., Mesgari, M. & Wiering, M. (2017) Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C*. 85, 732-752.

Bagloee, S. & Sarvi, M. (2017) A modern congestion pricing policy for urban traffic: subsidy plus toll. *Journal of modern transportation.* 25(3), 133-149.

Banks, A., Vincent, J. & Anyakoha, C. (2007) A review of particle swarm optimisation. Part I: background and development. *Nat Comput* 6(4), 467-484

Barriere, J., Farges, J. & Henry, J. (1987). Decentralization vs hierarchy in optimal traffic control. *IFAC Proceedings Volumes, 20*(3), 209-214.

Barton, R. & Meckesheimer, M. (2006) Metamodel-Based Simulation Optimisation. In: Henderson, S. & Nelson, B. (eds.) *Handbooks in Operations Research and Management Science.* Volume 13. Amsterdam, Elsevier Science, pp. 535-574.

Bellman, R. (1957) *Dynamic programming*, Princeton N.J.: Princeton University Press.

Bingham, E. (2001) Reinforcement learning in neurofuzzy traffic signal control. *European Journal of Operational Research*. 131(2), 232-241.

BITRE (2015) T*raffic and congestion cost trends for Australian capital cities*. Bureau of Infrastructure, Transport and Regional Economics. Information Sheet:74.

Box, G. & Wilson, K. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society B*. 13, 1–45.

Cai, C. (2009) *Adaptive traffic signal control using approximate dynamic programming*. PhD thesis. University College London, UK.

Cai, C., Wong, C. & Heydecker, B. (2009) Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C*. 17(5), 456-474.

Cai, X., Qiu, H., Gao, L., Yang, P. & Shao, X. (2017) A multi-point sampling method based on Kriging for global optimisation. *Structural and Multidisciplinary Optimisation.* 56(1), 71-88.

Castro-Neto, M., Jeong, Y., Jeong, M. & Han, L. (2009) On-line SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications.* 36(3), 6164-6173.

Chen, X., Xiong. C., He, X., Zhu, Z. & Zheng, L. (2016) Time-of-day vehicle mileage fees for congestion mitigation and revenue generation: A simulation-based optimisation method and its real-world application. *Transportation Research Part C.* 63, 71-95.

Chen, X., Yin, M., Song, M., Zhang, L. & Li, M. (2014b) Social welfare maximisation of multimodal transportation: theory, metamodel, and application to Tianjin Ecocity China. *Transportation research record.* 2451(1), 36-49.

Chen, X., Zhang, L., He, X., Xiong, C. & Li, Z. (2014a) Surrogate-Based Optimisation of Expensive-to-Evaluate Objective for Optimal Highway Toll Charges in Transportation Network. *Computer-Aided Civil and Infrastructure Engineering.* 29(5), 359-381.

Chen, X., Zhang, L., He, X., Xiong, C. & Zhu, Z. (2017) Simulation-based pricing optimisation for improving network-wide travel time reliability. *Transportmetrica.* 14(1-2),155-176.

Chen, X., Zhu, Z. & Zhang, L. (2015) Simulation-based optimisation of mixed road pricing policies in a large real-world network. *Transportation Research Procedia.* 8, 215-226.

Chen, X., Zhu, Z., He, X. & Zhang, L. (2015) Surrogate-Based Optimisation for Solving a Mixed Integer Network Design Problem. *Journal of Transportation Research Board: Transportation Research Record.* 2497, 124-136.

Chiu, S. & Chand, S. (1993) Adaptive traffic signal control using fuzzy logic. *Second IEEE International Conference on Fuzzy Systems*. 2, 1371-1376.

Chong, L. & Osorio, C. (2018) A simulation-based optimisation algorithm for dynamic large-scale urban transportation problems. *Transportation Science.* 52(3), 497-737

Choudhary, A. & Gokhale, S. (2016) Urban real-world driving traffic emissions during interruption and congestion. *Transportation Research Part D.* 43(2016), 59-70.

Christofa, E. & Skabardonis, A. (2011) Traffic signal optimisation with application of transit signal priority to isolated intersection. *Transportation Research Record: Journal of the Transportation Research Board.* 2259, 192-201.

Christofa, E., Papamichail, I. & Skabardonis, A. (2013) Person-based traffic responsive signal control optimisation. *IEEE Transactions on Intelligent Transportation Systems.* 14(3) 1278-1289.

Chu, T., Wang, J., Codeca, L. & Li, Z. (2020) Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems.* 21(3), 1086-1095.

Conn, A. R., Gould, Nicholas I. M., & Toint, Ph. L. (2000) *Trust-Region Methods.* MPS-SIAM Series on Optimisation, Society for Industrial and Applied Mathematics and Mathematical Programming Society. Philadephia, PA, USA.

Conn, A., Scheinberg, K. & Vicente, L. (2009) *Derivatice-Free Optimisation.* MPS-SIAM series on optimisation, SIAM, Philadephia, PA.

De Vliger, I., De Keukeleere, D. & Kretzschmar, J. (2000) Environmental effects of driving behaviour and congestion related to passenger cars. *Atomospheric Environment.* 34(27), 4649-4655.

Dell'Olmo, P. & Mirchandani, P. (1995) REALBAND: An Approach for Real-Time Coordination of Traffic Flows on a Network. *Transportation Research Record.* 1494, 106-116.

Dell'Olmo, P. & Mirchandani, P. (1996) A Model for Real-Time Traffic Coordination Using Simulation Based Optimisation. In: Bianco, L. & Toth, P. (eds.) *Advanced Methods in Transportation Analysis*. Springer-Verlag, Germany, pp. 525-546.

Diakaki, C. (1999) *Integrated Control of Traffic Flow in Corridor Networks.* Ph.D. Thesis, Technical University of Crete.

Diakaki, C. , Dinopoulou, V. , Aboudolas, K. , Papageorgiou, M. , Ben-Shabat, E. , & Seider, E. (2003). Extensions and new applications of the traffic-responsive urban control strategy: coordinated signal control for urban networks. *Transportation Research Record Journal of the Transportation Research Board, 1856*, 202-211.

Diakaki, C. , Papageorgiou, M. , & Mclean, T. . (1999). Application and evaluation of the integrated traffic-responsive urban corridor control strategy IN-TUC in Glasgow. *78th Annual Meeting of the Transportation Research Board*.

Diakaki, C., Papageorgiou, M. & Aboudolas, K. (2002) A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Eng. Practice*. 10, 183–195.

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

Dion, F., & Hellinga, B. (2002). A rule-based real-time traffic responsive signal control system with transit priority: application to an isolated intersection. *Transportation Research Part B,* 36(4), 325-343.

Dion, F. & Yagar, S. (1996) Real-time control of signalised networks-different approaches for different needs. *Proceedings of the eighth IEE international conference on road traffic monitoring and control.* Pp. 56-60. Savoy Place, London, UK

Dong, H., Song, B., Wang, P. & Dong, Z. (2018) Surrogate-based optimisation with clustering-based space exploration for expensive multimodal problems. *Structural and*

Multidisciplinary Optimisation. 57, 1553-1577

Eberhart, R. & Kennedy, J.(1995) A New Optimiser Using Particles Swarm Theory, Proc. *Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan),* IEEE Service Center, Piscataway, NJ, pp. 39-43,

El-Tantawy, S. & Abdulhai, B. (2010) Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers (MARLIN-OTC). *Transportation letters.* 2(2), 89-110.

Ercan, M. (2009) Particle swarm optimisation and other metaheuristic methods in hybrid flow shop scheduling problem. In: Lazinica, A. (eds.) *Particle Swarm Optimisation.* InTech. Pp.155-168.

Farges, J., Khoudour, L. & Lesort, J. (1990) PRODYN: on site evaluation. In: Proc. 3rd IEE Conf. Road Traffic Control. London, U.K. IEE, pp. 62–66.

Forrester, A. & Keane, A. (2009) Recent advances in surrogate-based optimisation. *Progress in aerospace sciences*. 45(1-3),50-79.

Forrester, A.., Sobester, A. & Keane, A. (2007) Multi-fidelity optimisation via surrogate modelling. Proceedings of the Royal Society. 463(2688), 3251-3269.

Forrester, A.., Sobester, A. & Keane, A. (2008) *Engineering Design via Surrogate Modelling: A Practical Guide.* Chichester, John Wiley & Sons, Ltd.

Friesz, T. (2010) *Dynamic optimisation and differential games.* Nternational series in operations research & management, v35. Newyork, London. Springer.

Garitselov, O., Mohanty, S. & Kougianos, E. (2012) A Comparative Study of Metamodels for Fast and Accurate Simulation of Nano-CMOS Circuits. *IEEE Transactions on Semiconductor Manufaturing.* 25(1), 26-36.

Gartner, N. (1983) OPAC: a demand-responsive strategy for traffic signal control. *Transportation Research Record.* 906, 75-81.

Gartner, N., Pooran, F. & Andrews, C. (2001) *Implementation of the OPAC adaptive control*

strategy in a traffic signal network. In: Proc. 4th IEEE Conf. Intelligent Transportation Systems, pp. 197–202

Gartner, N., Pooran, F., & Andrews, C. (2002) Optimised policies for adaptive control strategy in real-time traffic adaptive control systems: implementation and field testing. *Transportation Research Record Journal of the Transportation Research Board.* 1811, 148-156.

Gartner, N., Stamatiadis, C. & Tarnoff, P. (1995) Development of advanced traffic signal control strategies for intelligent transportation system multilevel design. *Transportation Research Record.* 1494, 98-105.

Girianna, M. & Benekohal, R. (2004) Using genetic algorithm to design signal coordination for oversaturated networks. *Journal of Intelligent Transportation Systems*. 8(2), 117-129.

Goodwin, P. & Noland, R. (2003) Building new roads really does create extra traffic: a response to Prakash et al.. *Applied Economics.* 35(13), 1451-1457.

Han, K. & Gayah, V. (2015) Continuum signalized junction model for dynamic traffic networks: Offset, spillback, and multiple signal phases. *Transportation Research Part B: Methodological*. 77, 213-239.

Han, K. Gayah, V., Piccoli, B. Friesz, TL. & Yao, T. (2014) On the continuum approximation of the on-and-off signal control on dynamic traffic networks. *Transportation Research Part B.* 61, 73-97

Han, K., Eve, G. & Friesz, TL. (2019) Computing dynamic user equilibria on large-scale networks with software implementation. *Networks & Spatial Economics.* 19(3), 869-902.

Han, K., Liu, H., Gayah, V.V., Friesz, T.L. & Yao, T. (2016a) A robust optimisation approach for dynamic traffic signal control with emission considerations. *Transportation Research Part C.* 70, 3-26.

Han, K., Piccoli, B. & Friesz, TL. (2016b) Continuity of the path delay operator for dynamic network loading with spillback. *Transportation Research Part B: Methodological*. 92 (B), 211-233.

Han, K., Piccoli, B. & Szeto, W.Y. (2015) Continuous-time link-based kinematic wave model: formulation, solution existence, and well-posedness. *Transportmetrica B: Transport Dynamics*. 4 (3), 187-222.

He, X., Chen, X., Xiong, C., Zhu, Z. & Zhang, L. (2016) Optimal time-varying pricing for toll roads under multiple objectives: A simulation-based optimisation approach. *Transportation science.* 51(2), 412-426.

Head, K. (1995) An Event-Based Short-Term Traffic Flow Prediction Model. *Transportation Research Record.* 1510, 45-52.

Henry, J.J., Farges, J.L. & Tuffal, J. (1983) The PRODYN real time traffic algorithm. *Proceedings of the fourth IFAC-IFIP-IFORS conference on Control in Trasnportation Systems.* pp. 307-311.

Hochreiter, S. & Schmidhuber, J. (1997) Long short-term memory. Neural computation. 9(8), 1735-1780.

Hunt, P., Robertson, D. & Bretherton, R. (1982) The SCOOT on-line traffic signal optimisation technique. *Traffic Engineering and Control.* 25, 14-22.

Hunt, P., Robertson, D., Bretherton, R. & Winton, R. (1981) SCOOT-A Traffic Responsive Method of Coordinating Signals. Transport and Road Research Laboratory Report number: Report LR 1041.

Jakobsson, S., Patriksson, M., Rudholm, J. & Wojciechowski, A. (2010) A method for simulation based optimisation using radial basis functions. *Optimisation and Engineering*, 11(4), 501–32.

Jones, D., Perttunen, C., Stuckman, B. (1993) Lipschizian optimization without the Lipschitz constant. Journal of optimization theory and application. 79(1), 157-181.

Jones D., Schonlau M., & Welch W. (1998) Efficient Global Optimisation of Expensive Black-Box Functions. *Journal of Global Optimisation*. 13(4), 455-492.

Kellner, F. (2016) Exploring the impact of traffic congestion on CO2 emissions in freight

distribution networks. *Logistics Research.* 9(1), 1-15.

Kennedy, J. & Eberhart, R.(1995) Particle Swarm Optimisation. *IEEE Conference on Neural Networks (Perth, Australia)*, Piscataway, NJ, pp. 1942-1948.

Khamis, M. & Gomma, W. (2014) Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence.* 29, 134-151.

Kosonen, I. (2003) Multi-agent fuzzy signal control based on real-time simulation. *Transportation Research Part C*. 11(5), 389-403.

Krige, D. (1951) A Statistical Approach to Some Basic Mine Valuations Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa.* 52(6), 119-139.

Le Thi, H., Vaz, A. & Vicente, L. (2012) Optimizing radial basis functions by d.c. programming and its use in direct search for global derivative-free optimisation. *Top*, 20(1), 190-214.

Lee, J., Abdulhal, B., Shalaby, A. & Chung, E. (2005) Real-time optimisation for adaptive traffic signal control using genetic algorithm. J*ournal of Intelligent Transportation Systems.* 9(3), 111-122.

Li, L., Lv, Y. & Wang, F. (2016) Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica.* 3(3), 247-254.

Li, M., Lin, X. & Chen, X. (2017) A surrogate-based optimisation algorithm for network design problems. *Frontiers of Information Technology & Electronic Engineering.* 18(11), 1693-1704.

Li, X., Chen, J., Yu, X., Zhang, X., Lei, F., Zhang, P. & Zhu, G. (2018) Data Fault Identification and Repair Method of Traffic Detector. Computational Science-ICCS 2018, 10862, pp.567-573.

Li, X., Li, G., Pang, S., Yang, X. & Tian, J. (2004) Signal timing of intersections using

integrated optimisation of traffic quality, emissions and fuel consumption: a note. *Transportation Research Part D.* 9, 401-407.

Lighthill, M. & Witham, F. (1955) On kinematic waves: II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London.* 229(178), 317-345.

Lin, W. & Wang, C. (2004) An enhanced 0-1 mixed-integer LP formulation for traffic signal control. *IEEE Transactions on Intelligent Transportation*. 5(4), 238-245.

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

Lophaven, S. N., Nielsen, H. B., & Søndergaard, J. (2002). *DACE - A Matlab Kriging Toolbox, Version 2.0*.

Lowrie, P. (1982) The Sydney coordinated adaptive traffic system-principles, methodology, algorithms. *Proceedings of the IEE International Conference on Road Traffic Signalling*. pp. 67-70.

Lu, X. (2015) Effectiveness of government enforcement in driving restrictions: a case in Beijing, China. *Environmental Economics and Policy Studies.* 18(1), 63-92.

Ma, Y., Zhou, W. & Han, Q. (2017) Research of multi-point infill criteria based on multi-objective optimisation front and its application on aerodynamic shape optimisation. *AIAA Journal*. 48(5), 995-1006.

Mascia, M., Hu, S., Han, K., North, R., Van Poppel, M., Theunis, J., Beckx, C. & Litzenberger, M. (2016) Impact of Trffic Management on Black Carbon Emissions: a Microscopic Study. *Newtworks and Spatial Economics.* 17(1), 269-291.

Mirchandani, P. & Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis, *Transportation Research Part C: Emerging Technologies*. 9(6), 415–432.

Murat, Y., & Gedizlioglu, E. (2005). A fuzzy logic multi-phased signal control model for isolated junctions. *Transportation Research Part C,* 13(1), 19-36.

Nahi, N. (1973) Freeway-traffic data processing. Proceedings of the IEEE. 61(5), 537-541.

Osorio, C. & Bierlaire, M. (2009) A surrogate model for traffic optimisation of congested networks: an analytical queueing network approach. Transport and Mobility Laboratory. Report number: TRANSP-OR 090825.

Osorio, C. & Bierlaire, M. (2013) A Simulation-Based Optimisation Framework for Urban Transportation Problems. *Operations Research.* 61(6), 1333-1345.

Osorio, C. & Selvam, K. (2015) Solving large-scale urban transportation problems by combining the use of multiple traffic simulation models. In: *4th International Symposium of Transport Simulation-ISTS'14, 1-4 June 2014, Corsica, France.* Transportation Research Procedia 6 (2015), pp 272 – 284.

Osorio, C. & Selvam, K. (2017) Simulation-based optimisation: achieving computational efficiency through the use of multiple simulators. *Transportation Science.* 51(2), 395-789

Osorio, C. (2010) *Mitigating network congestion: Analytical models, optimisation methods and their applications.* PhD thesis. Ecole Polytechnique Federale de Lausanne, Switzerland.

Prashanth, L. & Bhatnagar, S. (2011) Reinforcement learning with function approximation for traffic signal control. *IEEE transactions on intelligent transportation systems.* 12(2), 412-421.

Rakha, H., Van Aerde, M., Ahn, K. & Trani, A. (2000) Requirements for evaluating traffic signal control impacts on energy and emissions based on instantaneous speed and acceleration measurements. *Transportation Research Record.* 1738, 56-67.

Regis, R. & Shoemaker, C. (2007) A Stochastic Radial Basis Function Method for the Global Optimisation of Expensive Functions. *INFORMS Journal on Computing*, 19(4), 497-509.

Richards, P. (1956) Shock waves on the Highway. *Operations Research*. 4(1), 42-51.

Robertson, D. & Bretherton, R. (1991). Optimizing networks of traffic signals in real time-the scoot method. *Vehicular Technology IEEE Transactions on, 40*(1), 11-15.

Robertson, D. (1969) TRANSYT: a traffic network study tool. *Road Research Laboratory, Ministry of Transport, Crowthorne, Berkshire.* Report number: Report LR253.

Robertson, D. (1986). Research on the transyt and scoot methods of signal coordination. *Ite Journal, 56*(1), 36-40.

Rodriguez-Roman, D. (2018) A surrogate-assisted genetic algorithm for the selection and design of highway safety and travel time improvement projects. *Safety Science*. 103, 305-315.

Sacks, J., Welch, W., Mitchell, T., & Wynn, H. (1989) Design and Analysis of Computer Experiments. *Statistical Science.* 4(4), 409-423.

Salkham, A., Cunningham, R., Garg, A. & Cahill, V. (2008) A collaborative reinforcement learning approach to urban traffic control optimization. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia, pp. 560-566.

Sasena, M., Papalambros, P. & Goovaerts, P. (2010) Exploration of metamodeling sampling criteria for constrained global optimisation. *Engineering Optimisation.* 34(3), 263-278.

Sayyadi, R. & Awasthi, A. (2016) A simulation-based optimisation approach for identifying key determinants for sustainable transportation planning. *International journal of systems science. Operations & logistics.* 5(2), 161-174.

Schneider, J. & Kirkpatrick, S. (2006) *Stochastic optimisation*. Berlin Heidelberg, Springer.

Schrank, D., Eisele, B. & Lomax, T. (2019) *2019 Urban Mobility Report.* Texas A&M Transportation Institute with cooperation from INRIX.

Shabestary, S. & Abdulhai, B. (2018) Deep Learning vs. Discrete Reinforcement Learning for Adaptive Traffic Signal Control. *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC).* pp. 286–293.

Shelby, S., Gettman, D., Ghaman, R., Sabra, Z. & Soyka, N. (2008) An overview and performance evaluation of ACS-Lite: A low cost adaptive signal control system. *Transportation research board annual meeting*. Transportation Research Board, Washington DC.

Shi, R., Steenkiste & Veloso (2018) Generating Synthetic Passenger Data through Joint Traffic-Passenger Modeling and Simulation. In: *21st International Conference on Intelligent*

Transportation Systems (ITSC). pp. 3391-3402.

Simpson, T., Peplinski, J., Koch, P. & Allen, J.. (2001) Metamodels for Computer-based Engineering Design: Survey and recommendations. *Engineering with Computers.* 17(2), 129-150.

Simpson, T., Toropov, V., Balabanov, V. & Viana, F. (2008) Design and analysis of computer experiments in multidisciplinary design optimisation: a review of how far we have come – or not. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimisation Conference, Victoria, British Colombia, 10-12 September 2008.*

Sims, A . (1980). The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on Vehicular Technology, 29*(2), 130-137.

Smith, M. & Mounce, R. (2011) A splitting rate model of traffic re-routeing and traffic control. *Transportation Research B.* 45(9), 1389-1409.

Sobester, A., Leary, S., & Keane, A. (2005). On the design of optimisation strategies based on global response surface approximation models. *Journal of Global Optimisation, 33*(1), 31-59.

Søndergarrd, J. (2003) *Optimisation using surrogate models – by the space mapping techniques*. PhD thesis. Technical University of Denmark.

Song, J., Hu, S., Han, K. & Jiang, C. (2010) Nonlinear decision rule approach for real-time traffic control for congestion and emission mitigation. *Networks and Spatial Economics*. 20(3), 675-702.

Stevanovic, A. (2006) *Assessing deterioration of pretimed, actuated-coordinated, and scoot control regimes in simulation environment*. PhD thesis. The University of Utah.

Sun, D., Benekohal, R. & Waller, T. (2006) Bi-level programming formulation and heuristic solution approach for dynamic traffic signal optimisation. *Computer-Aided Civil and Infrastructure Engineering.* 21(5), 321-333.

Sunkari, S. (2004) The Benefits of Retiming Traffic Signals. *ITE journal* 74(4), 26-30.

Templ, M. (2017) Synthetic Data. In: Statistical Disclosure Control for Microdata. Springer,

Cham.

Thøgersen, J. (2009) Promoting public transport as a subscription service: Effects of a free month travel card. *Transport Policy*. 16(6), 335-343.

Venter, G. & Sobieski, J. (2002) Particle swarm optimisation. *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference April 22-25, 2002, Denver, Colorado.* American Institute of Aeronautics and Astronautics, Inc.

Wang, C., Quddus, M. & Ison, S. (2013) A spatio-temporal analysis of the impact of congestion on traffic safety on major roads in the UK. *Tranportmetrica A: Transport Science.* 9(2), 124-148.

Wang, G. & Shan, S. (2007) Review of metamodeling techniques in support of engineering design optimisation. *Journal of Mechanical Design*. 129(4), 370-380.

Wang, Y., Yang, X., Liang, H. & Liu, Y. (2018) A review of the self-adaptive traffic signal control system based on future traffic environment. *Journal of advanced transportation.* 2018, 1-12.

Webster, F. (1958) *Traffic Signal Settings.* London, H. M. Stationary Office. Road research technical paper report number: 39.

Wei, W., Zhang, Y., Mbede, J. B., Zhang, Z. & Song, J. (2001) Traffic signal control using fuzzy logic and MOGA. *2001 IEEE International Conference on Systems, Man and Cybernetics. E-Systems and e_Man for Cybernetics in Cyberspace*. pp. 1335-1340.

Wiering, M. (2000) Multi-agent reinforcement learning for traffic light control. *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1151-1158.

Wiering, M., Vreeken, J., van Veenen, J. & Koopman, A. (2004) Simulation and optimisation of traffic in a city. *IEEE Intelligent Vehicles Symposium, 2004, Parma, Italy,* pp 453-458.

Wild, SM. & Shoemaker, C. (2011) Global Convergence of Radial Basis Function Trust Region Derivative-Free Algorithms. *SIAM Journal on Optimisation.* 21(3), 761-781.

Wu, D., Yin, Y., Lawphongpanich, S. & Yang, H. (2012) Design of more equitable congestion pricing and tradable credit schemes for multimodal transportation networks. *Transportation Research Part B Methodological.* 46(9),1273–1287 25.

Yin, B., Dridi, M. & EI Moudni, A. (2015) Adaptive traffic signal control for multi-intersection based on microscopic model. *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. pp. 49-55.

Zhang, L., Chen, X., He, X. & Xiong, C. (2014) Bayesian stochastic Kriging metamodel for active traffic management of corridors. *IIE Annual Conference, Montreal, Canada, May31-June 3, 2014*.

Zhang, X., Fu, Y., Jiang, S. Xue, X., Jiang, Y. & Agam, G. (2018) Stacked multichannel autoencoder-an efficient way of learning from synthetic data. *Multimedia Tools and Applications.* 77, 26563-26580.

Zheng, L., Xue, X., Xu, C. & Ran, B. (2019) A stochastic simulation-based optimization for equitable and efficient network-wide signal timing under uncertainties. *Transportation Research Part B: Methodological.* 122, 287-308.

Appendix A

Summary of real time signal controls in Chapter 2

	Optimisation method	Structure	Traffic model	Objective
Cai, Wong & Heydecker (2009)	ADP	isolated	Vertical queuing theory	Average delay
Cai (2009)	ADP	Isolated/distributed	Vertical queuing theory/CTM	Average delay
Yin, Dridi & EI Moundni (2015)	ADP	distributed	Vertical queuing theory & car following & lane choice	Average waiting time
Lin & Wang (2004)	MILP	centralised	CTM	Total delay & number of stops
Han et al. (2016a)	MILP	centralised	Link-based simulation model	Weighted throughput and emission (constraint)
Christofa, Papamichail & Skabardonis (2013)	MINLP	isolated	Queuing theory	Total person delay
Girianna & Benekohal (2004)	GA	centralised	Link-based simulation model	Net effect of released vehicles and disutility function
Lee et al.(2005)	GA	centralised	Link-based simulation model	Vehicle delay
Sun, Benekohal & Waller (2006)	GA	centralised	CTM	Travel time
Wei et al. (2001)	Fuzzy logic	isolated	×	Vehicle delay and percentage of stopped vehicles
Chiu & Chand (1993)	Fuzzy logic	distributed	×	
Kosonen (2003)	Fuzzy logic	distributed	×	
Bingham (2001)	Neurofuzzy	isolated	×	
Murat & Gedizlioglu (2005)	Fuzzy logic	isolated	×	

Dion & Hellinga (2002)	Rule-based approach	isolated	×	Delay
Wiering (2000) and Wiering et al. (2004)	Critic-only model-based RL	distributed	×	Waiting time
Khamis & Gomaa (2014)	Critic-only model-based RL	distributed	×	Trip waiting time, total trip time and junction waiting time
Li, Lv & Wang (2016)	Critic-only model-free RL	isolated	×	
Shabestary & Abdulhai (2018)	Critic-only model-free RL	isolated	×	
Chu et al. (2020)	Actor-critic RL	distributed	×	
Aslani, Mesgari & Wiering (2017)	Actor-critic RL	distributed	×	
Liu et al. (2015)	LDR	centralised	Link-based simulation model	Weighted throughput
Song et al. (2019)	NLD	centralised	Microscopic simulation model (S-Paramics)	Delay and emission

Table A. An overview of main characteristics of real time signal controls reviewed in Chapter 2

Appendix B

Numerical settings of the Glasgow network

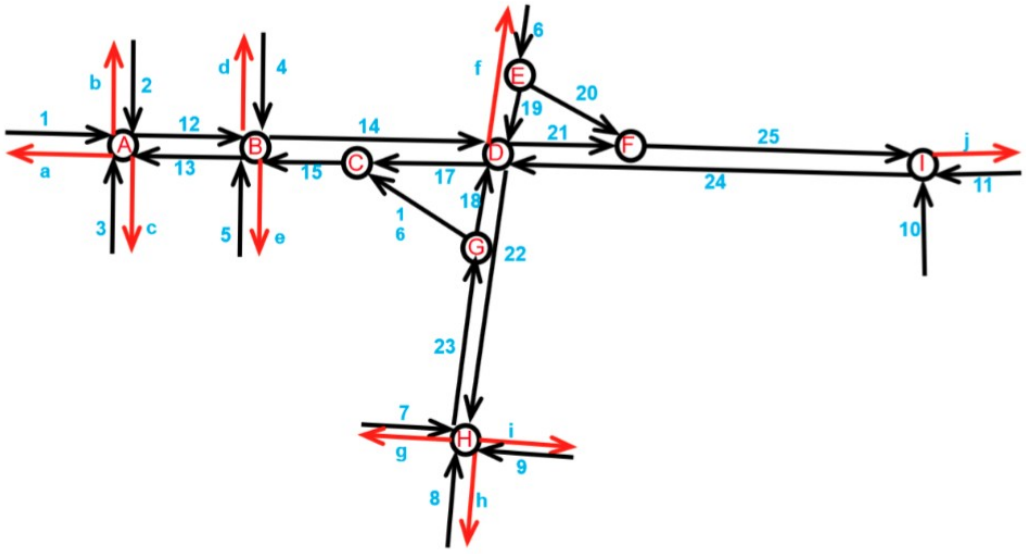


Figure B The abstract layout of Glasgow Network

	0-15 min	15-30min	30-45min	45-60min
Link 1	0.2732	0.2764	0.2175	0.2207
Link 2	0.1789	0.1950	0.1479	0.1704
Link 3	0.1307	0.1629	0.1307	0.1136
Link 4	0.1829	0.1476	0.1391	0.1048
Link 5	0.0654	0.0793	0.0686	0.0386
Link 6	0.0903	0.0827	0.0801	0.0643
Link 7	0.1400	0.1456	0.1444	0.1133
Link 8	0.0689	0.0800	0.0800	0.0678
Link 9	0.0767	0.0824	0.0989	0.0933
Link 10	0.0908	0.0999	0.0848	0.0575
Link 11	0.1917	0.1998	0.1927	0.1766

Table B.1. Average flow of eleven entrance links of Glasgow network (Veh/s)

Link → Link	Turning Ratio	Link → Link	Turning Ratio
1 → 12	0.6670	**9 → h**	0.3623
1 → b	0.0902	**10 → 24**	0.1667
1 → c	0.2428	**10 → j**	0.8334
2 → 12	0.1440	**12 → 14**	0.8911
2 → a	0.1130	**12 → d**	0.0495
2 → c	0.7430	**12 → e**	0.0594
3 → 12	0.0492	**13 → a**	0.8212
3 → a	0.5408	**13 → b**	0.0927
3 → b	0.4100	**13 → c**	0.0861
4 → 13	0.1403	**14 → 21**	0.8241
4 → 14	0.3743	**14 → 22**	0.0854
4 → e	0.4854	**14 → f**	0.0905
5 → 13	0.1967	**15 → 13**	0.8032
5 → 14	0.2131	**15 → d**	0.1181
5 → d	0.5902	**15 → e**	0.0787
6 → 19	0.8136	**19 → 17**	0.4097
6 → 20	0.1864	**19 → 22**	0.5903
7 → 23	0.1190	**22 → g**	0.0818
7 → h	0.1985	**22 → h**	0.4818
7 → i	0.6825	**22 → i**	0.4364
8 → 23	0.7419	**23 → 16**	0.1837
8 → g	0.0484	**23 → 18**	0.8163
8 → i	0.2097	**24 → 17**	0.7466
9 → 23	0.2319	**24 → 22**	0.1301
9 → g	0.4058	**24 → f**	0.1233

Table B.2. Turning ratio of Glasgow Network

Appendix C

Webster method

Webster was first proposed by Webster (1958). It aims to reduce the delay at the intersection. Smith & Mounce (2011) defined the red time cost of equi-saturation policy as Equation (C.1).

$$\textit{Red time cost: } \frac{f_i}{s_i g_i} \tag{C.1}$$

Where f_i is the flow of link *i*, which is an incoming link, s_i is the saturation flow of link *i* and g_i is the proportion of green time.

The red time cost changes until the red time costs of all the stages become the same. For an intersection with *k* stages, the green split should satisfy the following equation:

$$\frac{f_1}{s_1 g_1} = \frac{f_2}{s_2 g_2} = \cdots = \frac{f_k}{s_k g_k} \tag{C.2}$$

Let the red time costs of the links to be τ, then Equation (C.2) is converted to Equation (C.3):

$$\frac{f_1}{s_1 g_1} = \frac{f_2}{s_2 g_2} = \cdots = \frac{f_k}{s_k g_k} = \tau \tag{C.3}$$

Define b_i as the ratio of the link flow and saturation flow of link *i*, therefore

$$\frac{x_1}{s_1} = b_1, \frac{x_2}{s_2} = b_2, \cdots, \frac{x_k}{s_k} = b_k \tag{C.4}$$

Combine the Equation (C.3) and Equation (C.4) and the green split of each stage can be represented as a function of b and τ:

$$\frac{\mathrm{b}_1}{\mathrm{g}_1} = \frac{\mathrm{b}}{\mathrm{g}_2} = \cdots = \frac{\mathrm{b}_\mathrm{k}}{\mathrm{g}_\mathrm{k}} = \tau \tag{C.5}$$

$$g_1 = \frac{b_1}{\tau}, g_2 = \frac{b_2}{\tau}, \cdots, g_k = \frac{b_k}{\tau} \tag{C.6}$$

Since g_i is the proportion of the green time of link *i*, the sum of g_i of all the incoming links should be 1.

$$g_1 + g_2 + \cdots + g_k = 1 \tag{C.7}$$

$$\frac{b_1}{\tau} + \frac{b_2}{\tau} + \cdots + \frac{b_k}{\tau} = 1 \tag{C.8}$$

$$\frac{1}{\tau}(b_1 + b_2 + \cdots + b_k) = 1 \tag{C.9}$$

If all the link flows and saturation flows are known, then τ can be calculated according to the values of b and thus the green time of each stage can be calculated, the equations of green time is shown below.

$$g_i = \frac{b_i}{\sum_{n=1}^{k} b_n} \tag{C.10}$$