ABSTRACT

Title of dissertation: SIMULATION OPTIMIZATION
OF TRAFFIC LIGHT SIGNAL TIMINGS
VIA PERTURBATION ANALYSIS

William Casey Howell, Doctor of Philosophy, 2006

Dissertation directed by: Professor Michael C. Fu
Applied Mathematics
and Scientific Computation Program

We develop simulation optimization algorithms for determining the traffic light signal timings for an isolated intersection and a network of two-signalized intersections modeled as single-server queues. Both problem settings consider traffic flowing in one direction. The system performance is estimated via stochastic discrete-event simulation. In the first problem setting, we examine an isolated intersection. We use smoothed perturbation analysis to derive both left-hand and right-hand gradient estimators of the queue lengths with respect to the green/red light lengths within a signal cycle. Using these estimators, we are able to apply stochastic approximation, which is a gradient-based search algorithm. Next we extend the problem to the case of a two-light intersection, where there are two additional parameters that we must estimate the gradient with respect to: the green/red light lengths within a signal cycle at the second light and the offset between the two light signals. Also, the number of queues increases from two to five. We again derive both left-hand and right-hand gradient estimators of the all queue lengths with respect to the three aforementioned parameters. As before, we are able to apply gradient-based search based on stochastic approximation using these estimators. Next we reexamine the two aforementioned problem settings. However, this time we are solely concerned with optimization; thus, we model the intersections using three different stochastic fluid models, each incorporating different degrees of detail. From these new models, we derive infinitesimal perturbation analysis gradient estimators. We then implement these estimators on the underlying discrete-event simulation and are able to apply gradient-based search based on stochastic approximation using these estimators. We perform numerical experiments to test the performance of the three gradient estimators and also compare these results with finite-difference estimators. Optimization for both the one-light and two-light settings is carried out using the gradient estimation approaches.

SIMULATION OPTIMIZATION OF TRAFFIC LIGHT SIGNAL TIMINGS VIA PERTURBATION ANALYSIS

by

William Casey Howell

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:

Professor Michael C. Fu, Chairman/Advisor
Professor Raymond Johnson
Professor Paul Schonfeld
Professor Eric Slud
Professor Konstantina Trivisa

© Copyright by

William Casey Howell

2006

DEDICATION

To my family: Beatrice, William, Melanie and Casey with love and appreciation.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Michael Fu, for all of his help and guidance. I also would like to thank all the members of my committee for their time and help throughout this entire process.

I would like to thank my family for their unyielding love and support. I would like to thank all my friends for asking because they cared and also the ones that didn't ask because they understood.

I would like to thank Monica, Angela, Narryn, Kim, Tasha, Duane, Gikiri and Ahmed for their help, support, guidance and inspiration throughout this process. Thanks to Narryn for helping to inspire the research upon which this dissertation is based. Special thanks to Angela for her continued support and inspiration. Extra special thanks to Monica for keeping me going when I thought I couldn't.

ACRONYMS

AIMSUM Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks

CORSIM Corridor Simulation

CRONOS Control of Networks by Optimization of Switchovers

DES Discrete-event simulation

DNP Degenerate nominal path

FD Finite-difference

FDSA Finite difference stochastic approximation

FHWA Federal Highway Administration

GSMP Generalized semi-Markov process

IBP Initial busy period

IPA Infinitesimal perturbation analysis

MAXBAND arterial timing program that optimizes band-width based on a mixed-linear integer programming technique

NETSIM Network simulation program developed by FHWA

NP Nominal path

OPAC Optimized Policies for adaptive control

PA Perturbation analysis

PP Perturbed path

PRODYN Real-time traffic control system

RHODES Real Time Hierarchical Optimized Distributed Effective System

SA Stochastic approximation

SCOOT Split cycle offset optimisation technique

SD Symmetric-difference

SFM Stochastic fluid model

SIGCAP Signalized intersection capacity analysis program

SIGSET Signalized intersection traffic signal settings

SimTraffic micro simulation and animation of vehicular traffic software

SMARTEST Simulation modeling applied to road transport European scheme test

SPA Smoothed perturbation analysis

SPSA Simultaneous perturbation stochastic analysis

TRANSYT Traffic Network Study Tool

VISSIM Visual traffic simulation tool

SELECTED NOTATION

$L_q(t)$	$=$	# cars in queue q at time t
$\bar{L}_q(t)$	$=$	average queue length for queue q up to time t
	$=$	$\frac{1}{t}\int_0^t L_q(x)dx$
N	$=$	# red-green cycles simulated
$\bar{L}$	$=$	average number in traffic system
T_q	$=$	length of green light period for queue q
F_q	$=$	service time distribution for queue q with mean $1/\mu_q$, and p.d.f. f_q
G_q	$=$	interarrival time distribution for queue q with mean $1/\lambda_q$, and p.d.f. g_q
θ	$=$	vector of controllable variables
$\beta(\Delta\theta)$	$=$	critical event change due to a perturbation $\Delta\theta$
$\mathcal{Z}$	$=$	characterization, which is the set of conditioning quantities on the sample path on which the conditional contribution is estimated

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Problem Statement

Traffic congestion poses an ever increasing problem for major metropolitan areas across the United States and abroad. As the population continues to grow and the number of licensed drivers increases, the problem will inevitably worsen. The cost of traffic congestion goes far beyond the numerous lost hours by vehicle drivers due to congestion delays. Traffic congestion has negative environmental and health consequences as well. Environmentally, traffic congestion results in increased fuel consumption and air pollution. In terms of health consequences, traffic congestion leads to increased stress and mental and physical discomfort, which may contribute to a lower quality of life. Additionally, congestion slows the transportation of goods and services, which results in higher prices for consumers.

There are many causes of traffic congestion, such as weather, vehicular accidents, reckless driving, poor road design and road construction work zones. Although there are strategies employed to minimize these contributing factors, they rely heavily on the compliance of individual drivers; therefore, they are not always effective. As such, there is a need to focus on factors that can be more effectively controlled. Some of these factors include improved road infrastructure, access limitations and traffic controls. While all of these factors can be altered in order to reduce traffic congestion, some of these factors are more difficult to alter than others. One of the most obvious means of reducing traffic is the construction of more roadways; however, that approach has serious limitations. Traffic congestion and its associated problems will continue to increase, because roads and highways are unlikely to expand enough to alleviate the problems due to the cost and limited land supply. Therefore, other more cost efficient and feasible strategies are needed.

One of the most promising ways to reduce traffic congestion is better utilization and control of the existing infrastructure through efficient management of traffic systems. A traffic system is defined as the passage of vehicles through a road infrastructure. Roadways, controls (e.g., traffic signals, stop signs), drivers and vehicles are the four principal elements of these systems [38]. Traffic control is the process by which the passage of vehicles through a road infrastructure is governed. It is quite evident that the efficiency of traffic control directly depends on the efficiency and relevance of the control methodologies. Poor traffic control can lead to traffic congestion, whereas well-designed traffic control plans, such as efficient traffic signal timings, can significantly reduce traffic congestion. In fact, in traffic systems that contain traffic signals, control of traffic light signal timings is one of the least expensive and most effective means of reducing vehicular congestion in metropolitan road networks [43]. This is especially true in times of peak traffic flow, such as during morning and evening rush hours. Traffic signal management is one of the fastest methods to achieve traffic congestion improvements. While building new roads can takes years, a new traffic signal plan can be implemented in a matter of weeks.

Traffic signals are controlled by a plan that controls when and how to change phases. The plans used to guide these changes, known as traffic signal management (control), vary in complexity. The four control parameters for traffic signal management are:

- *Offset*: time between cycle starts of different signals;
- *Stage specification*: outlines what options (go straight, turn left, etc.) a vehicle has at any given time at each intersection;
- *Cycle time*: time it take a signal to progress through all stages;
- *Split*: portion of a cycle that the signal is green for each direction.

1.2 Literature Review

There are three main types of plan for traffic signal control: fixed-time (or pretimed), semi-actuated, and fully-actuated. Fixed-time plans are the simplest type of control plans. Each phase of the light lasts for a specific duration of time before the next phase begins. This type of plan uses historical data to determine and preset the signal timings. This type of system is used at many traffic lights. These settings are independent of the current traffic situation; however, some fixed-time control plans use multiple signal setting plans based upon the time of day. The advantage of fixed-time plans is that they require no extra hardware. In many cases, fixed-time plans are the only reasonable type of plan that can be implemented. The cost of vehicles sensors and detectors prohibit their installation at many intersections. Actuated plans are traffic responsive, which are more advanced but require vehicle sensors and detectors to acquire real-time data. For the most part, there are no preset signal timings in these plans; they set the timings based upon current and/or predicted traffic demand at the intersection. There are some actuated plans that choose a fixed-time plan based on current traffic situation. Even for a fully actuated plan, fixed-time plans are still employed as a back-up in case of sensor malfunction. Most fixed-time optimization approaches at signals without sensors are brute force methods, which require traffic data to be collected and analyzed.

In the research literature, the problem of efficient traffic flow via traffic signal management has been studied and reviewed via many different optimization methods.

1.2.1 Single Intersection

When considering a single intersection, there are two main fixed-time strategies, stage-based and phase-based. Stage-based strategies determine the optimal split and cycle times. Phase-based strategies take it one step further and also determine the optimal stage specifications.

Two of the well-known stage-based fixed-time strategies for a single intersection, SIGSET [1] and SIGCAP [2], were proposed by Allsop in 1971 and 1976, respectively. When given m specified stage specifications, SIGSET and SIGCAP will determine the optimal split and cycle times. SIGSET performs this optimization by deriving capacity constraints and an objective function. The objective function is a nonlinear delay function derived by Webster [47]. SIGSET seeks to minimize the total delay. As a result, the optimization problem becomes a linearly constrained nonlinear programming problem. SIGCAP seeks to maximize the intersection's capacity. Slight changes are made to the capacity constraints via the demands. These changes lead to a linear programming problem.

Phase-based fixed-time strategies [27] solve a similar type of problem. The problem addressed in stage-based strategies is extended to consider different staging specifications. These approaches determine split and cycle times, as well as stage specification in order to optimize the total delay or system capacity. The extension of determining stage specifications adds binary variables into the optimization problem, which leads to a binary mixed integer linear programming problem. This type of problem requires an application of a branch-and-bound method that acquire an exact solution. This causes the computation time to be greater than that of the stage-based fixed-time strategies.

As stated before, actuated plans use real-time data. The data is usually provided by detectors that are 40 meters upstream from the intersection. The plan basically employs some form of logic to make decisions on the traffic signal based upon the collected data at these detectors. One of the simplest actuated plans can be explained as follows. The light is kept green for some G_{min} (minimum green light length); if a vehicle is detected, then the light is allowed to stay green for an additional G_{extra} (small increment of time) time. The green light gets G_{extra} additional time until G_{max} (maximum allowable green light length) is reached.

There are more advanced versions of this type of plan, such as the one proposed by Miller [34], where the decision to change stage is made every T seconds. That is, the plan determines what the time gains and loses, M_k, would be in all directions if the decision to change stage is postponed by $k * T$ seconds, for $k = 1, 2, ...$ seconds. If $M_k < 0$ (i.e., no increase in performance) for all k, then the change is made immediately; otherwise the change is postponed until the next evaluation period.

1.2.2 Network of Intersections

As with the single intersection case, there are both fixed-time and actuated plans for networks of intersections. Two of the most popular fixed-time plans are MAXBAND [31, 32] and TRANSYT [36].

The earliest version of MAXBAND optimizes with respect to the offsets for n intersections of a two-way arterial. The splits and cycles are assumed given. MAXBAND determines the optimal offsets based on maximizing the number of vehicles that can travel within a given range without stopping. This leads to a binary mixed integer linear programming problem. In later versions [7], some clever techniques are employed in order to reduce the computational demands of the branch and bound solution method. Stamatiadas and Gartner [44] have made extensions to MAXBAND that make it applicable to networks of arterials. Gartner and others continue to improve and extend MAXBAND for wider application and better performance in [19, 20]. MAXBAND is in use worldwide.

Some of the most widely used traffic signal control strategies are based on TRANSYT [36]. Since Robertson first introduced it in 1969, numerous improvements and extensions have been made to it. TRANSYT-7F is the most current

U.S. version of TRANSYT. This strategy is so widely accepted that it is often the standard to which other strategies are compared. The basic idea of this method is a hill-climbing algorithm. TRANSYT starts off with given values for split and cycle times and offsets (note that the same cycle time is used for all intersections in the network). The performance measures of interest are estimated using simulation based upon the current system parameters. Next, the algorithm makes a slight perturbation to the system parameters and calculates the performance measures under these changes. This process continues until (local) optimization of the performance measures is achieved.

Robertson [37] made a natural progression from TRANSYT to SCOOT, which is basically a traffic-responsive version of TRANSYT. SCOOT has been put into use in over 150 cities in the United Kingdom. In SCOOT, real-time data is fed into a network model. The model is run in real time to see what effects small perturbations in cycle and split times, and offsets at individual intersections, will have on the network. As with TRANSYT, if the perturbed control parameters turn out to improve the performance measure of interest, then the changes are implemented on the true network.

When it comes to actuated control strategies for networks of intersections, some other methods employ a model-based optimization approach. The model-based control strategy is a fairly new approach. The previously mentioned methods also use models; however, this method makes use of very detailed models. Examples of this type of strategy include OPAC [18], PRODYN [10], CRONOS [4] and RHODES [39]. This approach uses a more detailed network model. It is based on pre-specified staging specifications. Based on the current stage and real-time data, it considers the optimal set of switching times over some horizon. Several constraints are included (e.g., minimum green light period) in this problem, allowing this strategy to attack the problem as a dynamic optimization problem. For efficient real-time application, a rolling-horizon procedure is employed. That is, the optimization time horizon is H seconds; however, only the changes for the next H seconds are implemented. Then the optimization problem is solved for an H second long horizon, and the process repeats.

According to the FHWA [11], there is a significant void in signal timing plans, because there is no model designed to provide signal settings for an isolated, actuated intersection. It is possible to use some of the aforementioned plans to deal with this problem; however, the FHWA views this approach as a work-around solution.

Gradient information can be valuable when it comes to system analysis, control and optimization. Of all the approaches reviewed, none make use of gradient information, with the exception of TRANSYT which uses estimated "brute-force" via actual perturbations. In this dissertation we use gradient information of a more direct nature, namely via perturbation analysis. In fact, to the best of our knowledge, this is the first successful application of perturbation analysis in the traffic signal setting.

1.3 Stochastic Optimization Tools

Stochastic optimization deals with finding the maximum or minimum of the expected value of a specific performance measure of interest when only a noisy estimation of the performance measure is available. In this dissertation we use Monte Carlo simulation to analyze the traffic systems, which is modeled with random interarrival and service times for the individual vehicles. While this is not an extremely detailed model, it behaves well when verified against real data.

1.3.1 Simulation

Simulation is defined as the technique of imitating the behavior of a real-world situation or system over time by means of an analogous model to gain information more conveniently [3]. It is a powerful tool for analyzing dynamic stochastic systems, and often the only means to model large and complex systems that arise in real-world applications. Some of the advantages of simulation include:

- Ability to compress and expand time;
- Ability to control sources of variation;
- Ability to restore system state;
- Allows for alterations to system parameters without affecting the real system;
- Facilitates replication.

Traffic analysis is a problem well suited for the use of simulation. Tartaro, Toress and Wainer [45] state that urban traffic analysis and control is a problem whose complexity makes difficult the analysis with traditional analytical methods. Traffic simulation has been in use as early as the 1950s. In fact, Webster [47] stated in 1958 that, “Since a theoretical calculation of delay is very complex and direct observation of delay on the road is complicated by uncontrollable variations, it was decided to use a method whereby the events on the road are reproduced in the laboratory by means of some machine which simulates behavior of traffic...”. This was one of the earliest suggestions that simulation could and should be used in traffic control. With the recent development in computer technology and programming tools, numerous traffic simulations have been created, and the concept is still gaining popularity.

When it comes specifically to traffic signal control, Hewage and Ruwanpura [26] state that computer simulation can be used effectively to analyze traffic flow patterns and signal light timings. In the 1970s, the Federal Highway Administration began using traffic simulation to evaluate traffic signal control systems. The Urban Traffic Control System was the first traffic simulation software developed. The resulting software was named Network Simulation (NETSIM) and was implemented in Washington, D.C. When developing a traffic signal control plan, interaction with a traffic simulation can be invaluable. Real traffic data is used to determine the needed distributions that govern the simulation. Using the traffic simulation in the developmental stages of a traffic signal control system provides a safe virtual environment and significantly reduces the software development time [21]. The safety comes from the fact that testing a signal control plan on a real system can have quite undesirable effects. The reduction in development time is achieved through the simulation via quicker run times of the traffic scenarios and also examination of rare cases. Also, the simulation can be used to gather crucial system data, whereas in the real system such data would be difficult, timely and expensive to gather. The system data collected from the simulation can be used in lieu of the data that would otherwise need to be collected from the real system. For these reasons, Ghaman et al. [21] state that traffic simulation plays a key role in the development of signal control systems.

Some of the traffic simulation systems in use today include CORSIM, VISSIM, AIMSUM, Paramics and SIMTRAFFIC. The SMARTEST project [40] reviews over fifty traffic simulation software packages.

1.3.2 Modeling

When it comes to traffic modeling, the two main approaches are microscopic and macroscopic. In microscopic models, each car's individual path through the system can be traced. Macroscopic models take a more aggregate approach. These modeled are often driven by flow rates.

We employ two forms of modeling in this dissertation. Neither model is intended to be an exact replica of the traffic system, but each model captures important qualities of the real traffic system. The underlying system is modeled using discrete-event simulation (DES). DES models are stochastic, because they contain some state variables that are random. In this model, significant changes (events) to the system occur at a finite or countable set of times. This model would fall under the microscopic approach. Lindgren and Tantiyanugulchai [30] state that there is a growing recognition that stochastic microscopic simulation models can be very useful in operational analysis.

We also model the traffic problem using stochastic fluid models (SFM). The SFM paradigm adopts a fluid flow view, as opposed to the transaction-flow view of traditional queueing models. The efficiency of a fluid model lies in its ability to aggregate multiple events. This model falls under the macroscopic approach.

1.3.3 Gradient Estimation

This dissertation applies gradient estimation techniques for stochastic simulation models. Indirect gradient estimation approaches estimate an approximation of the true gradient value. These approaches offer the greatest generality and flexibility. The most basic and straightforward forms of (indirect) gradient estimation use brute-force finite-differences (FD), which entails perturbing each component of the gradient separately while holding the other fixed. FD estimators can be computationally impractical for higher-dimensional problems, because each parameter that is perturbed requires two simulation runs. There are also difficulties in deciding the correct size of the perturbation. In an attempt to get a more accurate gradient estimator, small perturbations are often attempted; however, small perturbations can lead to other problems. One such problem is extremely noisy gradients due to the stochastic nature of the output. Another problem is due to the numerical complications of dividing two small numbers. For these reason, indirect gradient estimation, namely FD, is not always ideal. In this dissertation we use the symmetric-difference (SD) form of FD, where the system is evaluated under a positive and negative perturbation. The extra computation cost can be balanced out by the gain in accuracy.

Direct gradient estimation attempts to estimate the true gradient by analyzing the underlying stochastic system, requiring further analysis of the model. Direct gradient estimators usually provide an unbiased estimate and are more computationally efficient. Thus, for optimization, faster convergence can be achieved with direct gradient estimators. One method of direct gradient estimation is perturbation analysis (PA).

1.3.4 Perturbation Analysis

PA attempts to derive an unbiased gradient estimation from a single sample path of the system [29, 25]. PA does not require actual perturbation of the parameters; thus, it can be implemented online. In this dissertation we apply two different forms of PA.

The first and most basic form is infinitesimal perturbation analysis (IPA), which is simple to implement but often limited in applicability. A necessary assumption for IPA to work is that small perturbations in the parameters will in turn cause small perturbations in the performance measures of interest [15, 24]. When IPA is not applicable, there are other forms of PA available.

One such alternative approach, introduced by Gong and Ho [22], is smoothed perturbation analysis (SPA), which is a gradient estimation technique based on conditional Monte Carlo. The success of this approach relies on the smoothing property of the conditional expectation. SPA is a rather general technique; however, implementation usually depends heavily on the problem. Fu and Hu [15] have developed a formal generalized semi-Markov process (GSMP) implementation, which leads to an estimator with an IPA component and a conditional component.

1.4 Research Contributions

In this dissertation we develop algorithms, using gradient estimation and stochastic optimization, that determine the optimal traffic signal split settings for signalized traffic systems. Although optimization of a network of signalized intersections is the ultimate goal, we examine both a single intersection and a network of two intersections, as single intersections serve as the building block of all signalized intersection networks. We also note that the interaction between intersections is a critical element in network optimization. This interaction can be captured in the gradient estimation process.

Removing the one-way assumption of traffic flow should be relatively straightforward to handle. Allowing turns would require further analysis for the model; however, the derivation of the gradients estimators should follow a similar analysis the current model.

We apply both IPA and SPA as appropriate. The underlying DES traffic model requires an SPA estimator. When we model the traffic system using SFM, we are able to derive IPA estimators; however, both estimators are implemented on the underlying stochastic DES model for system analysis and optimization.

Our work contributes to the methodological foundations of metropolitan traffic flow management by introducing new simulation-based optimization algorithms for determining traffic light signal timings. We derive gradient estimators to carry out this optimization. Each of the estimators has at least one version (left-hand or right-hand) that requires no off-line simulation. The resulting estimators are computationally efficient and can be implemented in fixed-time plans, as well as on line with actuated plans, because they do not require altering the parameter values and are computationally inexpensive.

We model the traffic systems using two different methods and implement multiple gradient estimation techniques in conjunction with these models. Specifically, we provide the following research contributions:

- We develop unbiased and efficient gradient estimators in the single intersection traffic setting.

- We develop simpler PA gradient estimators based on SFM in the single intersection traffic setting.

- We apply these gradient estimators, both DES and SFM, in stochastic approximation (SA) for system optimization in the single intersection traffic setting.

- We extend these gradient estimation algorithms to a network of two signalized intersections.

- We compare these algorithms with known methods in numerical experiments.

The remainder of this dissertation is organized as follows. In Chapter 2, we develop an optimization algorithm for a single intersection. We develop unbiased SPA gradient estimators and employ them in SA for system optimization. We carry out numerical experiments, including comparisons with other known methods. In Chapter 3, we make the traffic system of interest a network of two signalized intersections. Once again, unbiased gradient estimators derived via SPA are used with SA for network control and optimization. We compare the performance of our estimators with those of other well-known approaches via numerical experiments. In Chapter 4, we look at both the single intersection and the network of two intersections problems. We derive IPA gradient estimators derived from SFM, but implemented on the underlying DES. We carry out numerical experiments to compare these results with those from the Chapter 2 and 3. In Chapter 5, we give conclusions and future work.

Chapter 2

Single Intersection

2.1 Introduction

Our proposed approach to the single intersection optimization problem is gradient based, and we derive simulation-based gradient estimators that are more efficient than brute-force finite differences; furthermore, they can be implemented online, which also differentiates the algorithm from that of Spall and Chin [43]. Due to the difficulty of the problem, we apply an approach called smoothed perturbation analysis (SPA), introduced by Gong and Ho [22]. Another simpler technique called infinitesimal perturbation analysis (IPA) is not applicable in the setting [23]. IPA is not applicable, because the sample performance measure is discontinuous in the parameter space. Because of this discontinuity, SPA, which uses conditional expectation, is required. For the single intersection of two one-way streets, we use the framework of Fu and Hu [14] to derive unbiased left-hand and right-hand gradient estimators for the queue lengths at each of the streets. We then employ these gradient estimators in a stochastic approximation algorithm to optimize the signal light timings. Numerical comparisons with optimization using finite difference estimators illustrate the promise of the proposed approach.

The rest of the chapter is organized as follows. In Section 2, we lay out the problem setting, including the queueing model and assumptions. In Section 3, we provide the detailed derivations of the various SPA estimators, including implementation details and proofs of unbiasedness. In Section 4, we report illustrative numerical results on the efficiency of the estimators and their effectiveness in optimizing traffic signal light timings. And finally in Section 5, we discuss some conclusions of the work.

2.2 Problem Setting

The system of interest consists of two one-way streets – labeled 1 and 2 in Figure 2.1 – intersecting at a traffic light that has two states:

A_1 The light is green for street 1. This state allows both departures and arrivals at street 1, but only arrivals at street 2.

A_2 The light is green for street 2. This state allows both departures and arrivals at street 2, but only arrivals at street 1.

In this model, we ignore yellow light lengths and assume the time to change from one state to another is negligible. The length of a green cycle in states A_1 and A_2 are denoted by T_1 and T_2, respectively. When the light is green for one street, the light is red for the other street. A complete signal cycle is defined by a green-red sequence, and the time to complete such a cycle is denoted by $T = T_1 + T_2$. We assume that the green-red cycle repeats identically and indefinitely, and without loss of generality assume that the sequence begins with a green for street 1. In state $A_j, j = 1, 2$, cars in street j's queue are served one at a time, according to i.i.d. "service times" with mean $1/\mu_j$, c.d.f. F_j and p.d.f. f_j, whereas in the other

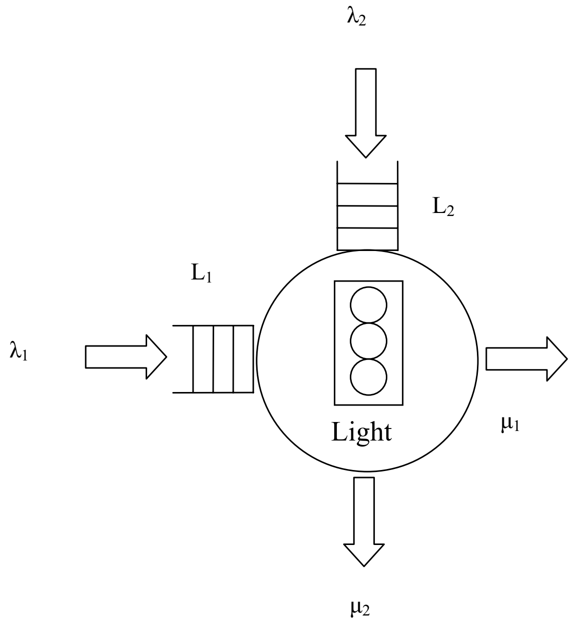


Figure 2.1: Isolated intersection traffic system visual depiction.

queue no cars are “served.” All service time and interarrival time distributions are continuous. If a car does not make it through the intersection during a cycle, it must “start over” with a fresh service time during the subsequent green cycle, i.e., the departure process must start over from scratch once it is realized that the car will not exit the queue during the current cycle. Arrivals to each street follow a renewal process with interarrival c.d.f. G_j, assumed to have finite rate λ_j. Unlike the departure process, the arrival processes to both intersection are “on” in both states. The performance measure of interest is the average number of cars waiting at the traffic light for a particular street. By Little’s Law, this is essentially equivalent to the average waiting time. We define:

$$
\begin{aligned}
L_j(t) &= \#\text{ cars waiting on street } j \text{ at time } t\ (j = 1, 2); \\
\bar{L}_j(t) &= \text{average queue length for street } j \text{ up to time } t \quad = \quad \frac{1}{t}\int_0^t L_j(x)dx; \\
N &= \#\text{ red-green cycles simulated}; \\
\bar{L} &= \bar{L}_1(NT) + \bar{L}_2(NT).
\end{aligned}
$$

In other words, the average total queue length performance measure $\bar{L}$ is taken over N green-red cycles. Note that "queue length" throughout includes all cars waiting at the street, even the one currently "in service."

The optimization problem is then given by

$$\min_{T_1,T_2} E[\bar{L}] \tag{2.1}$$

$$\text{subject to} \quad T_1 + T_2 = T,$$

which we propose to solve by satisfying the first-order condition

$$\nabla_\theta E[\bar{L}] = 0, \tag{2.2}$$

where θ is the vector of controllable variables (parameters), e.g., T_1 and T_2. T is fixed; therefore, we can remove the dependence on T_2 and the optimization problem described by equation 2.1 can be rewritten as

$$\min_{T_1} E[\bar{L}] \tag{2.3}$$

To find the value of θ satisfying equation (2.2), we use gradient-based simulation optimization via a stochastic approximation recursion of the following form:

$$\theta_{n+1} = \Pi_\Theta \left(\theta_n - a_n \widehat{\nabla} E[\bar{L}(\theta_n)]\right), \tag{2.4}$$

where a_n is a positive sequence of step sizes, $\widehat{\nabla}$ represents a gradient estimate, and Π_Θ is a projection onto the feasible region Θ.

The gradient estimate in (2.4) requires estimators for

$$\frac{dE[\bar{L}_j]}{d\theta}, \quad j = 1, 2. \tag{2.5}$$

We assume T is given, so the constraint essentially reduces (2.1) to a single-variable optimization problem. With T fixed, a positive perturbation in T_1 results in a negative perturbation in T_2 and vice versa. Although we take T_1 and T_2 as deterministic, a more general formulation could have T_1 and T_2 as random variables, with θ as a parameter in the distribution of T_1 or T_2.

2.3 Derivation of Estimators

Now we examine the concept of SPA as given by Gong and Ho [22]. By defining

$$\begin{aligned} \theta &= \text{controllable parameter,} \\ \xi &= \text{random sequence of numbers defined on a probability space } (\Omega, F, P), \end{aligned}$$

we have that a stochastic DES can be represented by the pair (θ, ξ). Then the performance measure of interest $L(\theta, \xi)$ is a random variable on (Ω, F, P).

Next, let $\mathcal{F}_t$ be an increasing family of σ-algebras on (Ω, F, P), that is generated by the simulation model up to time t. Now let N be the number of cycles simulated and let T be the length of a cycle, then it follows that NT is the duration of a simulation run. The characterization, $\mathcal{Z}$, is a $\mathcal{F}_{NT}$ measurable random vector. $\mathcal{Z}$ is a set of data obtainable from the data generated from a simulation such as queue content, services times, interarrival times, etc. The choice of which data makes up the characterization is a problem dependent decision. The SPA estimator then can stated as

$$\left(\frac{dE[\bar{L}_j]}{d\theta}\right)_{SPA} = \frac{1}{N}\sum_{k=1}^{N}\frac{\delta}{\delta\theta}L(\theta,\mathcal{Z}_k) \tag{2.6}$$

where

$$\frac{\delta}{\delta\theta}L(\theta,\mathcal{Z}) = \lim_{\Delta\theta\to 0}\frac{E[\Delta L(\theta,\xi)|\mathcal{Z}]}{\Delta\theta}. \tag{2.7}$$

It is the case that the estimator represented in equation (2.6) can be implemented provided that $\frac{\delta}{\delta\theta}L(\theta,\mathcal{Z}_k)$ can be calculated from the simulation run generated by (θ,ξ_k).

Now we consider the conditions that will ensure the consistency of the estimator represented in equation (2.6). By the strong law of law numbers we have

$$\begin{aligned}\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\frac{\delta}{\delta\theta}L(\theta,\mathcal{Z}_k) &= \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\frac{\delta}{\delta\theta}L(\theta,\mathcal{Z}(\theta,\xi_k))\\ &= E\frac{\delta}{\delta\theta}L(\theta,\mathcal{Z})\\ &= E\lim_{\Delta\theta\to 0}\frac{E[\delta L(\theta,\xi)|\mathcal{Z}]}{\Delta\theta}. \qquad (2.8)\end{aligned}$$

So the question is under what conditions

$$\begin{aligned}\frac{\delta}{\delta\theta}L(\theta,\xi) &= \lim_{\Delta\theta\to 0}E\frac{E[\delta L(\theta,\xi)|\mathcal{Z}]}{\Delta\theta}\\ &= E\lim_{\Delta\theta\to 0}\frac{E[\delta L(\theta,\xi)|\mathcal{Z}]}{\Delta\theta} \qquad (2.9)\end{aligned}$$

holds. So basically we are left with requiring an interchange of limit and expectation. Conditions for such an interchange involve applying the dominated convergence theorem.

Following the framework of Fu and Hu [14], the general SPA estimator consists of an infinitesimal perturbation analysis (IPA) term and a conditional term, the latter due to possible critical event order changes, which intuitively are changes in the order of events in a sample path that drastically alter the performance measure of interest. For instance, in our traffic light setting, a perturbation might lead to one less or one more departure in a given green cycle. How to estimate the probability (rate) of such a change and the subsequent expected effect on the performance measure is the key to deriving the SPA estimator.

The general form of the SPA estimator is

$$\left(\frac{dE[\bar{L}_j]}{d\theta}\right)_{SPA} = \frac{d\bar{L}_j}{d\theta} + \lim_{\Delta\theta\to 0}\frac{P_{\mathcal{Z}}(\beta(\Delta\theta))}{\Delta\theta}\lim_{\Delta\theta\to 0}\delta E_{\mathcal{Z}}[\bar{L}_j(\beta(\Delta\theta))], \tag{2.10}$$

where $\beta(\Delta\theta)$ denotes a critical event change due to a perturbation of $\Delta\theta$, and $\delta E_{\mathcal{Z}}[\bar{L}_j(\beta(\Delta\theta))]$ denotes the corresponding expected change in the performance measure $E_{\mathcal{Z}}[\bar{L}_j]$. The subscript $\mathcal{Z}$ denotes a conditioning on the characterization, which is the set of conditioning quantities on the sample path on which the conditional contribution is estimated, and it will differ for each of the four estimators we derive. In addition to choosing $\mathcal{Z}$, the chief difficulty in implementing an SPA estimator is the estimation of the expected change, $\lim_{\Delta\theta\to 0}\delta E_{\mathcal{Z}}[\bar{L}_j(\beta(\Delta\theta))]$. Ideally, this quantity would be able to be estimated from the original sample path, which we call the nominal path (NP), but its general form is given as $\lim_{\Delta\theta\to 0}\delta E_{\mathcal{Z}}[\bar{L}_j(\beta(\Delta\theta))] = E_{\mathcal{Z}}\left[\bar{L}_j^{PP} - \bar{L}_j^{DNP}\right]$, which is defined by three other sample paths:

- **NP**: nominal path, the original sample path;

- **PP**: perturbed path, limiting version of nominal path on which the critical event change occurs, that is a version of the NP on which the parameter that causes the event change is just big enough to cause the event change;

- **DNP**: degenerate nominal path, limiting version of the nominal path on which no critical event change occurs, that is a version of the NP on which the parameter that causes the event change is just small enough to not cause the event change;

where the superscripts denote the performance measures on the corresponding sample paths. Over N cycles, the estimator (2.10) becomes

$$\left(\frac{dE[\bar{L}_j]}{d\theta}\right)_{SPA} = \frac{d\bar{L}_j}{d\theta} + \sum_{i=0}^{N} \lim_{\Delta\theta\to 0} \frac{P_{\mathcal{Z}_i}(\beta_i(\Delta\theta))}{\Delta\theta} \lim_{\Delta\theta\to 0} E_{\mathcal{Z}_i}\left[\bar{L}_j^{PP_i} - \bar{L}_j^{DNP_i}\right]. \quad (2.11)$$

Since the optimization is with respect to T_1, we will take $\theta = T_1$ throughout. We derive four estimators: left-hand ($\Delta\theta \uparrow 0$) and right-hand ($\Delta\theta \downarrow 0$) estimators for each of the two streets, with l and r subscripts denoting left-hand and right-hand estimators, respectively. The critical event changes, $\beta_i(\Delta\theta)$, are quite intuitive: a shortening of a green cycle could cause a departure to be lost during the cycle, whereas a lengthening could allow an additional departure.

2.3.1 Right-Hand Estimator for Queue 1

We first consider queue 1 with $\Delta\theta > 0$, corresponding to the right-hand estimator for $dE[\bar{L}_1]/d\theta$. In this case ($\Delta\theta = \Delta T_1 > 0$), there is a positive perturbation in the green signal length of street 1 while keeping the total signal cycle length, T, unchanged. Since small perturbations at the end of T_1 do not affect the departure

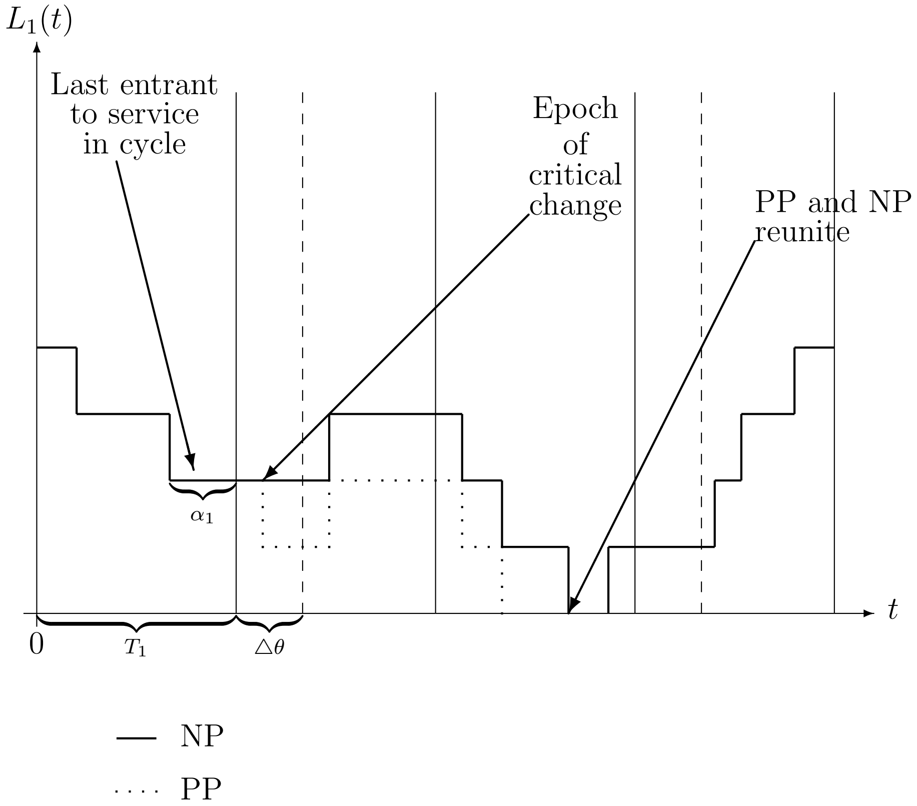


Figure 2.2: Example of $L_1(t)$ sample path for a single intersection traffic system with positive perturbation of ($\triangle\theta > 0$) T_1 cycle.

times of cars from street 1, the IPA contribution is zero. A small enough increase in the green signal length would not cause any change in the queue length for street 1; however, a large enough increase would lead to an additional departure; in other words, the performance measure is piecewise constant. The critical change in this case is this additional departure. An additional departure is possible if and only if the queue is nonempty at the light change. When nonempty, the last car to enter service is the only candidate for a critical change because the probability of more than one critical change is of higher order and thus can be ignored. Thus, we only consider the last car as a possible extra departure. To calculate the probability rate and expected effect of this critical change, we condition on all arrivals and service times except for the last entry to service during the current A_1 state. Since the only critical event change in a cycle is a function of the last entry to service, we can index by cycles, and we define:

$$
\begin{aligned}
\alpha_i &= \text{time until light change from last entry of service during } i\text{th cycle}, \\
\mathcal{S} &= \text{set of all service times}, \\
\mathcal{S}_i^* &= \text{last service time of } i\text{th cycle}, \\
\Lambda &= \text{set of all arrival times}, \\
\mathcal{Z}_i &= \mathcal{S} \backslash \{\mathcal{S}_i^*\} \ \cup \ \Lambda,
\end{aligned}
$$

where the service time for the last car to enter service (and not depart) in cycle i is greater than α_i. NOTE: If the service time was less than α_i, then that car would have exited the system. DNP and PP are then defined by the critical change occurring precisely at the green/red light change, with the service times of the last car to enter service being α_i^+ and α_i^-, respectively, where

$$
\begin{aligned}
\alpha_i^+ &= \alpha + \epsilon, \\
\alpha_i^- &= \alpha - \epsilon,
\end{aligned}
$$

for $\epsilon > 0$ infinitesimally small. If X denotes a random variable with service time distribution F_1, then the probability of a critical change is given by

$$P(\beta_i(\Delta\theta)) = P(X \leq \alpha_i + \Delta\theta \mid X \geq \alpha_i), \tag{2.12}$$

and hence

$$\lim_{\Delta\theta \to 0} \frac{P(\beta_i(\Delta\theta))}{\Delta\theta} = \frac{f_1(\alpha_i)}{1 - F_1(\alpha_i)}. \tag{2.13}$$

Thus, the estimator given by (2.11) becomes

$$\left(\frac{dE[\bar{L}_1]}{d\theta}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N}\frac{f_1(\alpha_i)}{1-F_1(\alpha_i)}E_{\mathcal{Z}_i}[\bar{L}_1^{PP_i} - \bar{L}_1^{DNP_i}]. \tag{2.14}$$

We note that NT is the length of each simulation run. To calculate the resulting expected effect, $E_{\mathcal{Z}}\left[\bar{L}_1^{PP_i} - \bar{L}_1^{DNP_i}\right]$, we observe that starting at the critical change, $L_1^{DNP_i}(t)$ will be identical to $L_1(t)$, whereas $L_1^{PP_i}(t)$ will be one lower than $L_1(t)$ until $L_1(t)$ empties. Thus, we have that $L_1^{DNP_i}(t) = L_1^{PP_i}(t) + 1$ for all t from the epoch of the first light change after the critical change to the time when the system first empties after the critical change (see Figure 2.2 for an example). Figure 2.2 shows one possible sample path, we note that the last entrant to service that does not exit the system could have also been an arrival to an empty queue. Thus,

$$E_{\mathcal{Z}_i}[\bar{L}_1^{PP_i} - \bar{L}_1^{DNP_i}] = -E[\min(NT, \inf\{t > \tau \mid L_1(t) = 0\})] - \tau, \tag{2.15}$$

where $\tau = iT - T_2$ corresponds to the epoch of the ith light change from green to red (for street 1). We subtract τ because the critical change only effects the sample path after its occurrence. Estimation of (2.15) can be done offline as follows. We define

γ_i = residual interarrival time at the epoch of the ith light change,

$R_N^{(1)}(\gamma_i)$ = expected time to empty queue 1, given N cars in the queue and an initial interarrival time of γ_i,

Q_i = number in queue at the epoch of the ith light change from green to red.

Thus, (2.15) can be rewritten as $R^{(1)}_{Q_i}(\gamma_i)$ and subsequently the estimator given by (2.14) becomes

$$\left(\frac{dE[\bar{L}_1]}{d\theta}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N}\frac{f_1(\alpha_i)}{1-F_1(\alpha_i)}\left[-R^{(1)}_{Q_i}(\gamma_i)\right]. \tag{2.16}$$

Note that the sign of the estimator will be negative, which makes intuitive sense, because an increase in the green signal length should decrease the average queue length.

2.3.2 Right-Hand Estimator for Queue 2

We now consider the case of queue 2 with $\Delta\theta > 0$ to derive the right-hand estimator for $dE[\bar{L}_2]/d\theta$. An increase in T_1 affects the entrance to service of cars in queue 2 and hence the departure times of cars in queue 2, because it results in a decrease in T_2, delaying the transition from state A_1 to A_2 and leading to an IPA perturbation in the departure times of every car in the initiating busy period (IBP) of the cycle. If the queue was empty at the beginning of the green period, we say that the particular cycle has no initiating busy period and hence there will be no IPA contribution for that cycle. Also any car that arrives after an idle period will not be affected by a perturbation in T_1, i.e., once the system empties, the perturbation is lost. The critical change for this case is a loss of a departure. A departure by a car that is in the initiating busy period may be eliminated by the perturbation and hence represents a potential critical change. To calculate the probability rate and expected effect of each of these possible critical changes, we condition on all arrival times and all service times except that of the kth initiating busy period departure of the ith period. We define:

$$\begin{aligned}
\alpha_i^k &= \text{time until light change from the entry to service of the} \\
&\quad\ \text{kth IBP departure,} \\
\mathcal{S}_i^{*k} &= \text{set of all service times of ith cycle prior to kth IBP departure,} \\
\mathcal{Z}_i^k &= \mathcal{S}\backslash\{\mathcal{S}_i^{*k}\} \ \cup \ \Lambda, \\
H_i &= \text{number of IBP departures during ith cycle,} \\
\beta_i^k &= \text{critical change cause by kth IBP departure during ith cycle,} \\
PP_i^k &= \text{perturbed path caused by kth IBP departure during ith cycle,} \\
DNP_i^k &= \text{degenerate nominal path caused by kth IBP departure} \\
&\quad\ \text{during ith cycle.}
\end{aligned}$$

If X denotes a random variable with service time distribution F_2, then the probability rate of a critical change is given by

$$\lim_{\Delta\theta\to 0}\frac{P(\beta_i^k(\Delta\theta))}{\Delta\theta} = \lim_{\Delta\theta\to 0}\frac{P(X \geq \alpha_i^k - \Delta\theta \mid X \leq \alpha_i^k)}{\Delta\theta} = \frac{f_2(\alpha_i^k)}{F_2(\alpha_i^k)}. \tag{2.17}$$

Thus, the estimator given by (2.10) becomes

$$\left(\frac{dE[\bar{L}_2]}{d\theta}\right)_{SPA,r} = \frac{1}{NT}\left(\sum_{i=1}^{N} H_i + \sum_{i=1}^{N}\sum_{k=1}^{H_i}\frac{f_2(\alpha_i^k)}{F_2(\alpha_i^k)}E_{\mathcal{Z}_i^k}[\bar{L}_2^{PP_i^k} - \bar{L}_2^{DNP_i^k}]\right). \tag{2.18}$$

Estimation of the expected difference between $\bar{L}_2^{PP_i^k}$ and $\bar{L}_2^{DNP_i^k}$ is similar to the previous estimator. The difference in these two performance measures is the difference in the time it takes for the two paths to empty, which can be estimated by simulating the expected time to empty the system, given that the initial queue length is equal to the queue length of the PP path at the time if the light change. Because arrivals are not affected by the perturbation of T_1, we must consider additional arrivals in the expected difference calculation. Defining

$$\begin{aligned}
R_N^{(2)}(\gamma_i) &= \text{expected time to empty queue 2, given } N \text{ cars in the queue} \\
&\quad \text{and an initial interarrival time of } \gamma_i, \\
Y_i^k &= \text{number in queue immediately after the epoch of the} \\
&\quad k\text{th IBP departure during the } i\text{th cycle}, \\
A_i^k &= \text{number of arrivals between } k\text{th entry to service and} \\
&\quad \text{next light change during } i\text{th cycle},
\end{aligned}$$

the final estimator becomes

$$\left(\frac{dE[\bar{L}_2]}{d\theta}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{H_i}\frac{f_2(\alpha_i^k)}{F_2(\alpha_i^k)}R^{(2)}_{Y_i^k+A_i^k}(\gamma_i). \tag{2.19}$$

The sign of the estimator will be positive, which makes intuitive sense, because a decrease in the green signal length should increase the average queue length.

2.3.3 Left-Hand Estimator for Queue 1

We now consider the case of queue 1 with $\Delta\theta < 0$ to derive the left-hand estimator for $dE[\bar{L}_1]/d\theta$. Decreasing T_1 affects the departures of cars in queue 1 by advancing the transition from state A_1 to state A_2. Because the perturbation occurs at the end of the cycle, a small enough perturbation will have no effect on the departure times of cars. Therefore, there is no IPA contribution for the estimator. The critical change for this case is again a loss of a departure. Every car that successfully completes service represents a potential critical change. To calculate the probability rate and expected effect of these possible critical changes, we condition on all arrival times and all service times except that of the kth departure. If X denotes a random variable with service time distribution F_1, then the probability rate of a critical change is given by

$$\lim_{\Delta\theta \to 0} \frac{P(\beta_i^k(\Delta\theta))}{\Delta\theta} = \lim_{\Delta\theta \to 0} \frac{P(X \geq \alpha_i^k - \Delta\theta \mid X \leq \alpha_i^k)}{\Delta\theta} = \frac{f_1(\alpha_i^k)}{F_1(\alpha_i^k)}. \tag{2.20}$$

Defining

$$D_i \quad = \quad \text{number of departures during the } i\text{th cycle,}$$

we get that the estimator given by (2.10) becomes

$$\left(\frac{dE[\bar{L}_1]}{d\theta}\right)_{SPA,l} = -\frac{1}{NT} \sum_{i=1}^{N} \sum_{k=1}^{D_i} \frac{f_1(\alpha_i^k)}{F_1(\alpha_i^k)} E_{\mathcal{Z}_i^k}[\bar{L}_1^{PP_i^k} - \bar{L}_1^{DNP_i^k}]. \tag{2.21}$$

Estimation of the difference between $\bar{L}_1^{PP_i^k}$ and $\bar{L}_1^{DNP_i^k}$ is identical to that in the previous estimator, so the final estimator is given by

$$\left(\frac{dE[\bar{L}_1]}{d\theta}\right)_{SPA,l} = -\frac{1}{NT} \sum_{i=1}^{N} \sum_{k=1}^{D_i} \frac{f_1(\alpha_i^k)}{F_1(\alpha_i^k)} R^{(1)}_{Y_i^k + A_i^k}(\gamma_i). \tag{2.22}$$

The sign of the estimator will be negative, which makes intuitive sense, because an increase in the green signal length should decrease the average queue length.

2.3.4 Left-Hand Estimator for Queue 2

The case of queue 2 with $\Delta\theta < 0$ for the left-hand estimator for $dE[\bar{L}_2]/d\theta$ is similar to the right-hand estimator for queue 1, except that there is an IPA component. A decrease in T_1 affects the entrance to service of cars in queue 2 and hence the departure times of cars in queue 2, because it results in an increase in T_2,

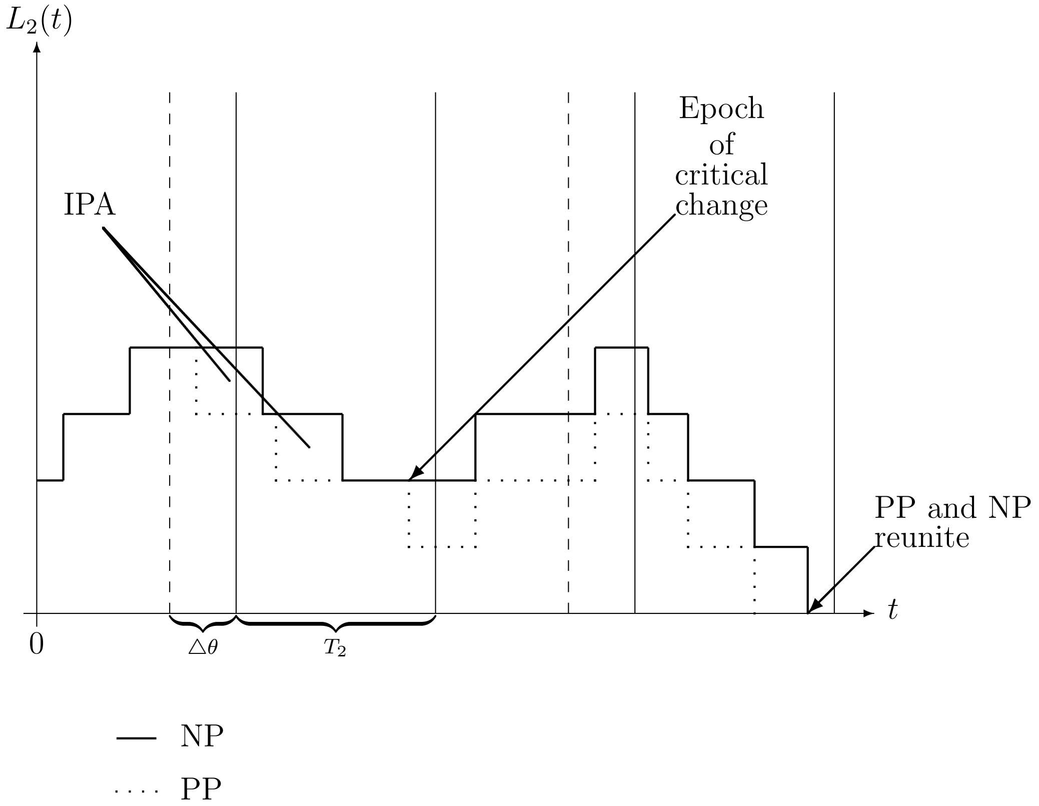


Figure 2.3: Example of $L_2(t)$ sample path for a single intersection traffic system with negative perturbation of ($\triangle\theta < 0$) T_1 cycle.

advancing the transition from state A_1 to A_2 and leading to a perturbation in the departure times of every car in the initiating busy period of the cycle (see Figure 2.3 for an example). If the queue was empty at the beginning of the green period, there will be no IPA contribution for that cycle, and any car that arrives after an idle period will not be affected by a perturbation in T_1, i.e., once the system empties, the perturbation is lost. Except for a sign change and a process for checking if a car is a member of the IBP, the rest of the analysis proceeds analogously to that used to derive (2.16). Defining

$$\mathbf{1}(\alpha) = \begin{cases} 0 & \text{if } \alpha \text{ corresponds to a IBP car} \\ 1 & \text{otherwise,} \end{cases} \tag{2.23}$$

the estimator given by (2.10) becomes

$$\left(\frac{dE[\bar{L}_2]}{d\theta}\right)_{SPA,l} = -\frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{1}{NT}\sum_{i=1}^{N} \frac{f_2(\alpha_i)}{1-F_2(\alpha_i)} E_{\mathcal{Z}_i}[\bar{L}_2^{PP_i} - \bar{L}_2^{DNP_i}]\mathbf{1}(\alpha), \tag{2.24}$$

and the final estimator is given by

$$\left(\frac{dE[\bar{L}_2]}{d\theta}\right)_{SPA,l} = \frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{1}{NT}\sum_{i=1}^{N} \frac{f_2(\alpha_i)}{1-F_2(\alpha_i)} \left[R_{Q_i}^{(2)}(\gamma_i)\right]\mathbf{1}(\alpha). \tag{2.25}$$

Note that the sign of the estimator will be positive, which makes intuitive sense, because a decrease in the green signal length should increase the average queue length.

2.3.5 Special Cases

For the special case of exponential interarrival and service times, (2.16), (2.19), (2.22), and (2.25) respectively simplify to

$$\left(\frac{dE[\bar{L}_1]}{d\theta}\right)_{SPA,r} = -\frac{\mu_1}{NT}\sum_{i=1}^{N} R_{Q_i}^{(1)}, \tag{2.26}$$

$$\left(\frac{dE[\bar{L}_2]}{d\theta}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{\mu_2}{NT}\sum_{i=1}^{N}\sum_{k=1}^{V_i} \frac{R_{Y_i^k+A_i^k}^{(2)}}{e^{\mu_2 \alpha_i^k}}, \tag{2.27}$$

$$\left(\frac{dE[\bar{L}_1]}{d\theta}\right)_{SPA,l} = -\frac{\mu_1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{V_i} \frac{R_{Y_i^k+A_i^k}^{(1)}}{e^{\mu_1 \alpha_i^k}}, \tag{2.28}$$

$$\left(\frac{dE[\bar{L}_2]}{d\theta}\right)_{SPA,l} = \frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{\mu_2}{NT}\sum_{i=1}^{N} R_{Q_i}^{(2)}, \tag{2.29}$$

where the dependence of $R_N^{(j)}$ on the residual interarrival time has been removed due to the memoryless property of exponential distribution.

2.3.6 Unbiasedness of the Estimators

The estimators derived in the previous sections are unbiased if

$$E\left[\left(\frac{dE[\bar{L}_j]}{d\theta}\right)_{SPA}\right] = \frac{dE[\bar{L}_j]}{d\theta},\ j = 1, 2. \tag{2.30}$$

To establish equation (2.30), some additional conditions are required:

(A1) $F_1(\cdot)$ is Lipschitz continuous with Lipschitz constant K_1.

(A2) $F_2(\cdot)$ is Lipschitz continuous with Lipschitz constant K_2.

where for $I \subseteq \Re$, a function $f : I \to \Re$ is said to Lipschitz continuous if there exist a constant K such that $|f(x) - f(y) < K|x - y|$ for all $x, y \in I$ and the smallest such K for which this holds is called the Lipschitz constant.

We then have the following result.

Proposition 1.

(i) *Under condition (A1), (2.14) is an unbiased estimator for* $\frac{dE[\bar{L}_1]}{d\theta}$*,*

(ii) *Under condition (A1), (2.21) is an unbiased estimator for* $\frac{dE[\bar{L}_1]}{d\theta}$*,*

(iii) *Under condition (A2), (2.18) is an unbiased estimator for* $\frac{dE[\bar{L}_2]}{d\theta}$*,*

(iv) *Under condition (A2), (2.24) is an unbiased estimator for* $\frac{dE[\bar{L}_2]}{d\theta}$*.*

We establish (iv) of Proposition 1. The proofs of (i), (ii), and (iii) proceed similarly; thus, their details are omitted here. To proceed, we introduce the following additional notation:

$$\begin{aligned}
\Gamma(NT) &= \{i \leq NT : L_2(iT;\theta) > 0\}; \\
\mathcal{A}_k &= \{L_2(t;\theta) = L_2(t;\theta+\Delta\theta), t = T, 2T, \ldots, kT\}; \\
\mathcal{B}_k &= \{L_2(t;\theta) = L_2(t;\theta+\Delta\theta), t = T, 2T, \ldots, (k-1)T\} \\
&\quad \cup \{L_2(KT;\theta) = L_2(KT;\theta+\Delta\theta)\},
\end{aligned}$$

$k = 1, 2 \ldots, N$, where $\Delta\theta = \Delta T_1 > 0$. The set $\mathcal{Z}_k$ is the characterization for our estimator; it contains everything except the service time of the last entrant to service in period k. $\mathcal{A}_k$ and $\mathcal{B}_k$ are both functions of $\Delta\theta$, though we omit the explicit display of the argument. The event $\mathcal{B}_k$ indicates that a perturbation in the value of θ to $\theta+\Delta\theta$ first causes a change in the queue length in period k. The event $\mathcal{A}_N$ represents the case where the perturbation does not cause a change in the queue length over the entire sample path. Thus, $\mathcal{B}_1, \ldots, \mathcal{B}_N, \mathcal{A}_N$ partition our sample space and we can write

$$\begin{aligned}
\frac{dE[\bar{L}_2]}{d\theta} &= \lim_{\Delta\theta\to 0}\left\{\frac{E[(\bar{L}_2(\theta+\Delta\theta) - \bar{L}_2(\theta))\mathbf{1}(\mathcal{A}_n)]}{\Delta\theta}\right. \\
&\quad \left. + \sum_{k=1}^{N} \frac{E[(\bar{L}_2(\theta+\Delta\theta) - \bar{L}_2(\theta))\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta}\right\}. \qquad (2.31)
\end{aligned}$$

We first prove the following lemma.

Lemma 1. *Under condition (A2),*

(a) $E[\bar{L}_2(\theta+\Delta\theta) - \bar{L}_2(\theta))\mathbf{1}(\mathcal{A}_n)] = \Delta\theta \times E\left[\frac{1}{NT}\sum_{i=1}^{N} H_i\right]$.

(b) $\lim_{\Delta\theta\uparrow 0} \frac{E[\bar{L}_2(\theta+\Delta\theta) - \bar{L}_2(\theta))\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta} = E\left[\frac{1}{NT}\frac{f_2(\alpha_k)}{1-F_2(\alpha_k)} R^{(2)}_{Q_k}(\gamma^k)\right]$.

(c) $\lim_{\Delta\theta\downarrow 0} \frac{E[\bar{L}_2(\theta+\Delta\theta)-\bar{L}_2(\theta))\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta} = E\left[\frac{1}{NT}\sum_{i=1}^{V_k} \frac{f_2(\alpha_k^i)}{F_2(\alpha_k^i)} R^{(2)}_{Y_k^i+A_k^i}(\gamma_k)\right]$.

Proof. For part (a), recall that in Section 2.3.4 we showed

$$(L_2(\theta+\Delta\theta) - L_2(\theta))\mathbf{1}(\mathcal{A}_N) = -\sum_{i=1}^{N} H_i \Delta\theta, \tag{2.32}$$

which establishes (a).

For part (b), we consider $E[(L_2(\theta+\Delta\theta) - L_2(\theta))\mathbf{1}(\mathcal{B}_k)], k = T, 2T, \ldots, NT$. First we rewrite it as

$$E[E[(L_2(\theta+\Delta\theta) - L_2(\theta))\mathbf{1}(\mathcal{B}_k)|\mathcal{Z}_k]]. \tag{2.33}$$

We have

$$\begin{aligned}
&|E[L_2(\theta+\Delta\theta)\mathbf{1}(\mathcal{B}_k)|\mathcal{Z}_k]| \\
&\quad = |E[L_2(\theta+\Delta\theta)|\mathcal{Z}_k, \alpha_k(\theta) < \mathcal{S}_k^* < \alpha_k(\theta+\Delta\theta)]\mathbf{1}(\mathcal{A}_{k-1}) \\
&\qquad \times P(\alpha_k(\theta) < \mathcal{S}_k^* < \alpha_k(\theta+\Delta\theta))| \\
&\quad = |E[L_2(\theta+\Delta\theta)|\mathcal{Z}_k, \alpha_k(\theta) < \mathcal{S}_k^* < \alpha_k(\theta+\Delta\theta)]\mathbf{1}(\mathcal{A}_{k-1}) \\
&\qquad \times (F_2(\alpha_k(\theta+\Delta\theta)) - F_2(\alpha_k(\theta)))| \\
&\quad \le K_2\Delta\theta E[L_2(\theta+\Delta\theta)|\mathcal{Z}_k, \alpha_k(\theta) < \mathcal{S}_k^* < \alpha_k(\theta+\Delta\theta)],
\end{aligned}$$

where $\alpha_k(\theta+\Delta\theta) = \alpha_k(\theta)+\Delta\theta$ and the last inequality follows from assumption (A2). To bound the expectation, we introduce notation for a renewal counting process based on the arrivals (without service). Let $\{X_n, n = 1, 2, \ldots\}$ be a sequence of i.i.d. interarrival times with common distribution G_2, and denote the associated counting process $\{P(t), t \ge 0\}$. G_2 generates nonnegative interarrival times with a finite rate

(λ_j); thus, $G_2(0) < 1$. Noting that arrivals are independent of θ, from basic renewal theory, we have

$$E\left[\sup_{\Delta\theta} L_2(\theta + \Delta\theta)|\mathcal{Z}_k, \alpha_k(\theta) < \mathcal{S}_k^* < \alpha_k(\theta + \Delta\theta)\right] \;\; \leq \;\; E[P(NT)] < \infty,$$

so by invoking the dominated convergence theorem (DCT), we have

$$\begin{aligned}
\lim_{\Delta\theta \to 0} \frac{E[\bar{L}_2(\theta + \Delta\theta)\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta} &= E\left[\lim_{\Delta\theta \to 0} \frac{E[\bar{L}_2(\theta + \Delta\theta)\mathbf{1}(\mathcal{B}_k)|\mathcal{Z}_k]}{\Delta\theta}\right] \\
= \;\; & E\left[\lim_{\Delta\theta \to 0} \frac{(F_2(\alpha_k(\theta + \Delta\theta)) - F_2(\alpha_k(\theta)))}{\Delta\theta}\right. \\
& \times \lim_{\Delta\theta \to 0} E[\bar{L}_2(\theta + \Delta\theta)|\mathcal{Z}_k, \alpha_k(\theta) < \mathcal{S}_k^* < \alpha_k(\theta + \Delta\theta)] \\
& \times \mathbf{1}(\mathcal{A}_{k-1})] \\
= \;\; & E[f_2(\alpha_k)E[\bar{L}_2^{PP_k}(t)]] \\
= \;\; & E\left[\frac{f_2(\alpha_k)}{1 - f_2(\alpha_k)}\mathbf{1}\{\mathcal{S}_k^* > \alpha_k(\theta)\}E[\bar{L}_2^{PP_k}(t)]\right].
\end{aligned}$$

We can similarly show

$$\lim_{\Delta\theta \to 0} \frac{E[\bar{L}_2(\theta)\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta} = E\left[\frac{f_2(\alpha_k)}{1 - f_2(\alpha_k)}\mathbf{1}\{\mathcal{S}_k^* > \alpha_k(\theta)\}E[\bar{L}_2^{DNP_k}(t)]\right]. \qquad \square$$

By establishing a bound for each part of our estimator, we are able to use the DCT to make the necessary expectation and limit switch. Thus, combining Lemma 1 with (2.31) establishes (iv) of Proposition 1. The following lemma and parts (c) and (d) of Lemma 1, needed to establish (i), (ii), and (iii), can be proven analogously, where again $\theta = T_1$.

Lemma 2. *Under condition (A1),*

(a) $E[\bar{L}_1(\theta + \Delta\theta) - \bar{L}_1(\theta))\mathbf{1}(\mathcal{A}_n)] = 0.$

(b) $\lim_{\Delta\theta\downarrow 0} \frac{E[\bar{L}_1(\theta+\Delta\theta)-\bar{L}_1(\theta))\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta} = E\left[\frac{1}{NT}\frac{f_1(\alpha_i)}{1-F_1(\alpha_k)}R^{(1)}_{Q_k}(\gamma_k)\right].$

(c) $\lim_{\Delta\theta\uparrow 0} \frac{E[\bar{L}_1(\theta+\Delta\theta)-\bar{L}_1(\theta))\mathbf{1}(\mathcal{B}_k)]}{\Delta\theta} = E\left[\frac{1}{NT}\sum_{i=1}^{V_k}\frac{f_1(\alpha_k^i)}{F_1(\alpha_k^i)}R^{(1)}_{Y_k^i+A_k^i}(\gamma_k)\right].$

For the special case of F_j $(j = 1,2)$ exponentially distributed, (A1) and (A2) are automatically satisfied, so we have the following corollary.

Corollary 1. *If G_j and $F_j\,(i = 1,2)$ are exponential distributions, then (2.28) and (2.26) are unbiased estimators for $\frac{dE[\bar{L}_1]}{d\theta}$, and (2.29) and (2.27) are unbiased estimators for $\frac{dE[\bar{L}_2]}{d\theta}$.*

2.4 Numerical Results

We implemented all four SPA estimators and compared them with various symmetric finite difference (FD) estimates for two sets of parameters. We then tested their use in optimization. In all cases, we took the interarrival times and service times to be exponentially distributed, so estimators (2.26),(2.27),(2.28), and (2.29) were used. The same estimators could be used as an approximation for non-exponential times.

2.4.1 Gradient Estimation

The first case ("C1") corresponds to symmetric street flows and signal timings: $\mu_1 = \mu_2 = 2.0; \lambda_1 = \lambda_2 = 4.5; T = 60, T_1 = T_2 = 30$. The second case ("C2") is an asymmetric system: $\mu_1 = 1.5, \mu_2 = 0.75; \lambda_1 = \lambda_2 = 5.0; T = 110, T_1 = 35, T_2 = 75$. The simulations were carried out for $N = 10,000$ cycles over 10,000 replications.

Estimator	$dE[\bar{L}_1]/d\theta$ (std. err.)	$dE[\bar{L}_2]/d\theta$ (std. err.)
SPA (RH)	-2.465 (0.001)	2.463 (0.001)
SPA (LH)	-2.465 (0.001)	2.464 (0.001)
FD (.05)	-2.475 (0.024)	2.455 (0.021)

Table 2.1: SPA gradient estimate simulation results for the isolated intersection traffic setting for "C1" (standard errors in parentheses).

Estimator	$dE[\bar{L}_1]/d\theta$ (std. err.)	$dE[\bar{L}_2]/d\theta$ (std. err.)
SPA (RH)	-8.303 (0.006)	0.0687 (0.000003)
SPA (LH)	-8.295 (0.007)	0.0687 (0.000003)
FD (.05)	-8.169 (0.115)	0.0687 (0.000035)

Table 2.2: SPA gradient estimate simulation results for the isolated intersection traffic setting for "C2" (standard errors in parentheses).

The results are shown in Tables 2.1 and 2.2. The FD estimates are quite sensitive to the difference value chosen; the best results are reported here, where the number in parentheses following the heading "FD" in the tables indicates the specific difference value. Even so, the SPA estimator is more precise, with a standard error always at least an order of magnitude better, and it is also more stable and computationally efficient. In fact, when $\bar{L}$ is also desired, the FD estimators require (on average) nearly three times as much computation time. The confidence intervals for all estimators overlap for both cases.

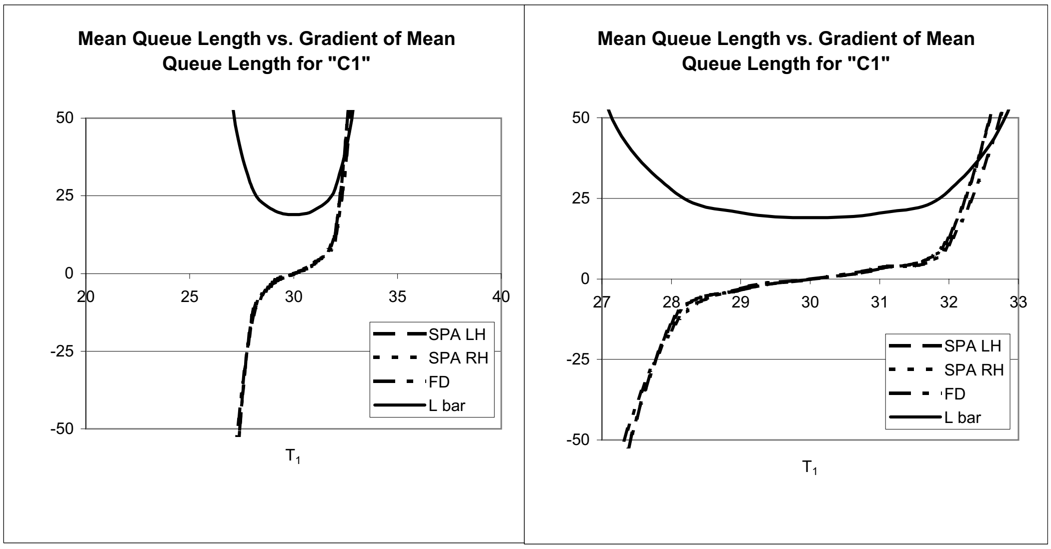


Figure 2.4: Plot and zoomed in plot of $E[\bar{L}]$ and $\frac{dE[\bar{L}]}{d\theta}$ simulation results for "C1".

2.4.2 Optimization

In the implementation of the SA algorithm (2.4), we take

$$\Theta = T\left(\frac{\lambda_1}{\mu_1}\right) < T_1 < T\left(1 - \frac{\lambda_2}{\mu_2}\right),$$

which represents the region of stability. We consider cases using the same values for μ_i, λ_i, and T as in section 2.4.1, where now T_1 will be optimized. Figures 2.4 and 2.5 depict the mean and gradient of the average queue length for both the SPA and FD estimators. The mean queue length was obtained via the discrete-event simulation model, in which 10,000 cycles were simulated over 10,000 replications. Because SA is an iterative algorithm, not only are we concerned with reaching the optimum, but we need subsequent updates to not cause deviation from optimum. To this end, we run simulations and count the number of times the average number in system was within p% (for p = 10,5,1) of the minimum average number in system based on the current T_1 from the SA algorithm. We label the three ranges as

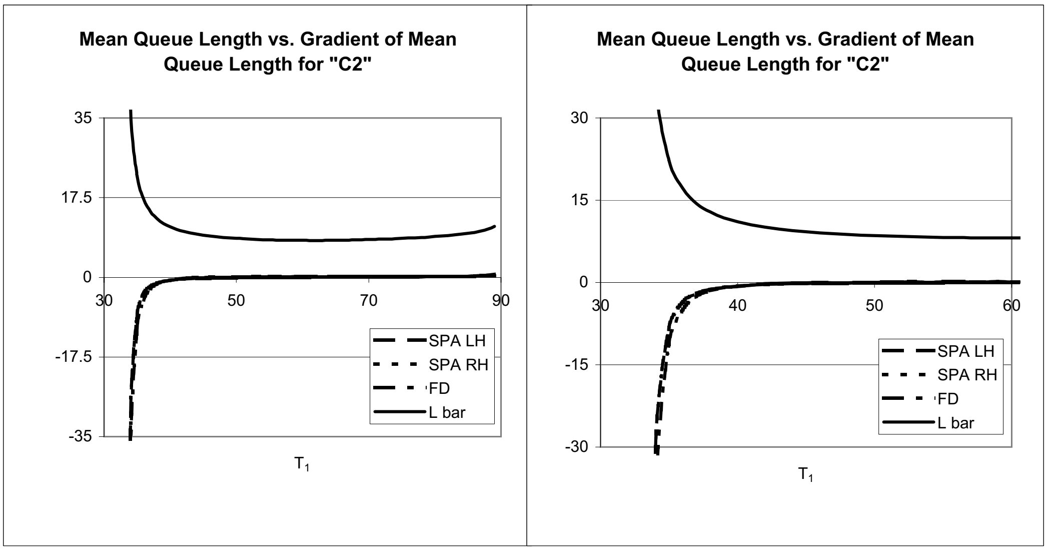


Figure 2.5: Plot and zoomed in plot of $E[\bar{L}]$ and $\frac{dE[\bar{L}]}{d\theta}$ simulation results for "C2".

- 10%-range : within 10% of the optimal $\bar{L}$;
- 5%-range : within 5% of the optimal $\bar{L}$;
- 1%-range : within 1% of the optimal $\bar{L}$.

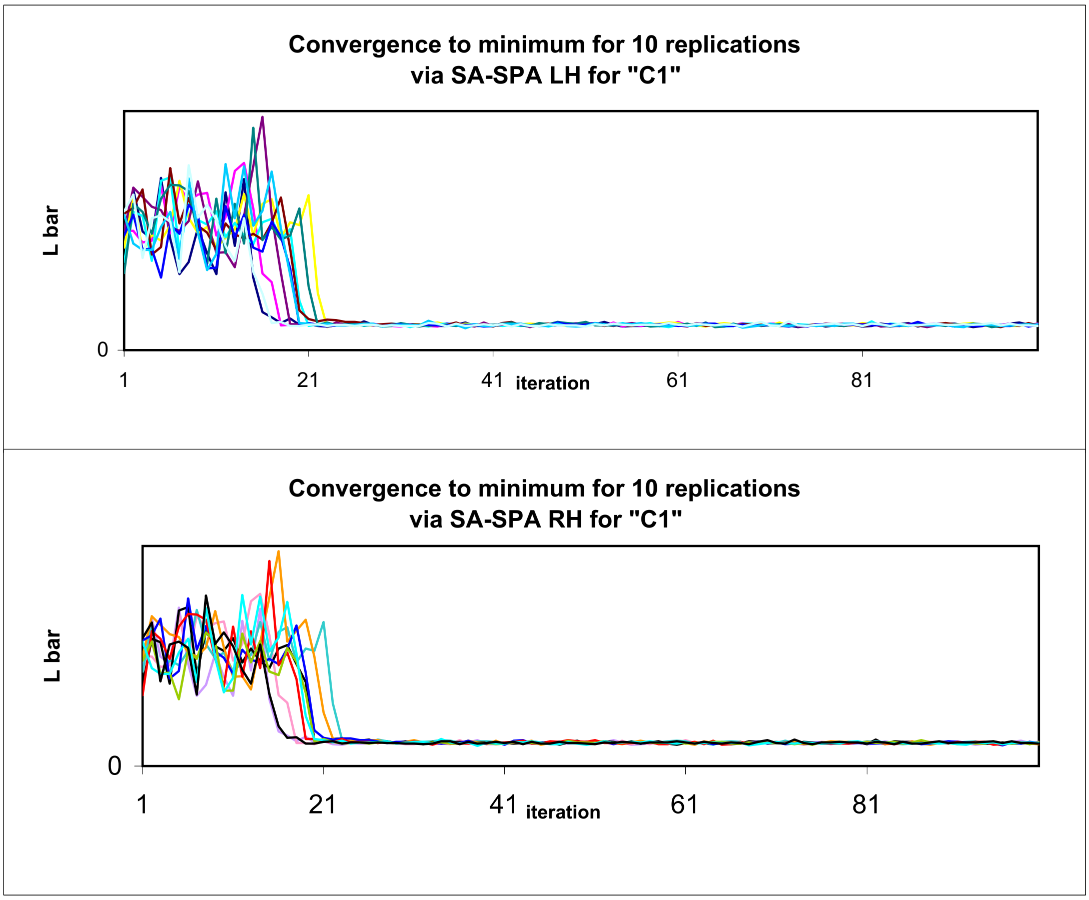
All three gradient estimators were implemented, in conjunction with a SA algorithm, for cases "C1" and "C2". Tables 2.3 and 2.4 show the number of times each iteration fell within the aforementioned optimum ranges. We also tested the SPA estimators with the SA algorithm to see if the optimal value was eventually reached. The SA algorithm was allowed to run for 100 iterations to see if the minimum $\bar{L}$ value was reached. This simulation was run for 10 different replications, and these results are shown in Figures 2.6 and 2.7. We can see that in each replication, the minimum $\bar{L}$ value was reached relatively quickly, and the estimator never caused a deviation from the minimum for subsequent iterations of the algorithm. These two tests show that the SPA estimators work just as well as FD estimators in iterative

Method	*10%-Range* (std. err.)	*5%* (std. err.)	*1%* (std. err.)
SPA (RH)	77.0 (0.004)	46.4 (0.005)	10.9 (0.003)
SPA (LH)	77.8 (0.004)	45.6 (0.005)	10.9 (0.003)
FD (.05)	76.5 (0.004)	45.0 (0.005)	10.1 (0.003)

Table 2.3: Mean number of times that the SA algorithm was in the optimal range for the isolated intersection traffic setting for “C1”.

Method	*10%-Range* (std. err.)	*5%* (std. err.)	*1%* (std. err.)
SPA (RH)	92.9 (0.003)	89.9 (0.003)	57.0 (0.005)
SPA (LH)	92.7 (0.003)	90.2 (0.003)	61.0 (0.005)
FD (.05)	94.0 (0.002)	91.8 (0.003)	60.1 (0.005)

Table 2.4: Mean number of times that the SA algorithm was in the optimal range for the isolated intersection traffic setting for “C2”.



Convergence to minimum for 10 replications via SA-SPA LH for "C1"

Convergence to minimum for 10 replications via SA-SPA RH for "C1"

Figure 2.6: Convergence to minimum for 10 replications of the SA algorithm for "C1" for two different gradient estimation methods: SPA LH SA and SPA RH SA.

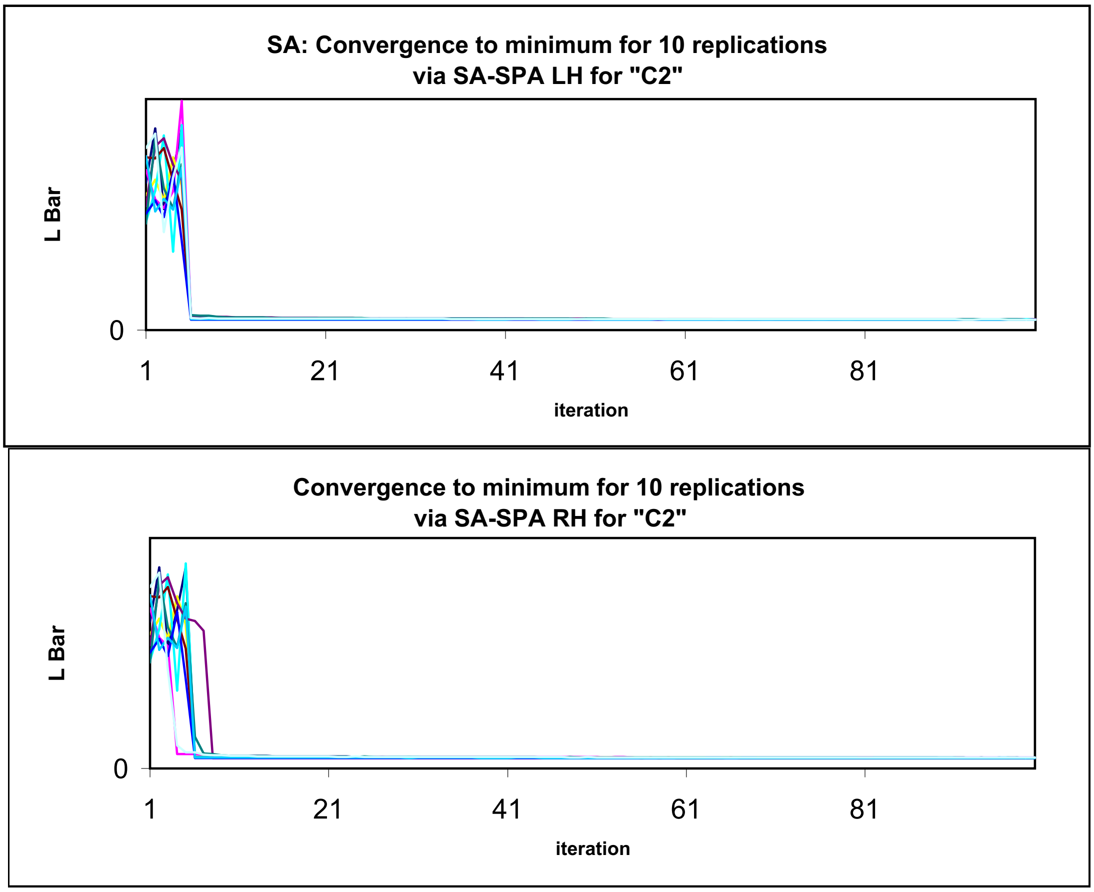


Figure 2.7: Convergence to minimum for 10 replications of the SA algorithm for "C2" for two different gradient estimation methods: SPA LH SA and SPA RH SA.

gradient descent algorithms such as SA; however, we again mention that FD requires more computational effort.

2.5 Conclusions

As far as we are aware, this is the first successful attempt to apply direct stochastic gradient estimation techniques to a traffic flow optimization setting. The resulting estimators demonstrated superior computational performance over FD estimators, and in addition can be used on line with real-time traffic updating systems, because unlike FD estimators, they do not require altering the parameter values. Thus, although we have considered only a single intersection, this work constitutes an important stepping stone in the foundation of simulation-based metropolitan traffic flow management. The next logical step is the analysis of two networked intersections. This step is addressed in the following chapter.

Chapter 3

Network of Two Signalized Intersections

3.1 Introduction

In this chapter, we again take a gradient-based approach. We derive simulation based gradient estimators for a network of two signalized intersections. SPA, which uses conditional expectation, is again required. We use the framework of Fu and Hu [14] to derive unbiased left-hand and right-hand gradient estimators for the queue lengths at each street of each light. We then incorporate these gradient estimators in a stochastic approximation algorithm to optimize the signal light timings. Numerical comparisons of gradient estimation and optimization are carried out with FD, FDSA and simultaneous perturbation stochastic approximation (SPSA). The results illustrate the promise of the proposed approach.

The rest of the chapter is organized as follows. In Section 2, we lay out the problem setting, including the queueing models and assumptions. In Section 3 we provide the detailed derivations of the various SPA estimators, including implementation details. In Section 4, we report illustrative numerical results on the efficiency of the estimators and their effectiveness in optimizing traffic signal timings. And finally, in Section 5, we discuss some conclusions of the work.

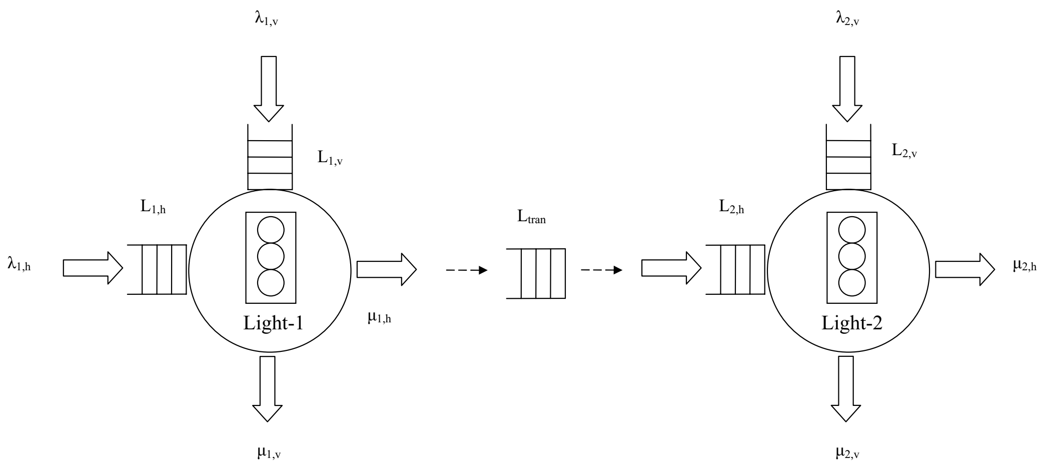


Figure 3.1: Network of two signalized intersections traffic system visual depiction.

3.2 Problem Setting

The system of interest consists of two signalized intersections, Light-1 and Light-2. Each of the two intersections is of the type analyzed in Chapter 2. In between the two intersections is a queue we call the transient queue. The transient queue represents the time it takes a vehicle to travel from Light-1 to Light-2. A visual representation of the system can be seen in Figure 3.1. Each parameter has subscripts that are used to identify which queue it references. Each of the two intersections, Light-1 and Light-2, has traffic flowing in from two directions, horizontal (h) and vertical (v). So we use subscripts of the form i, j, where $i \in \{1, 2\}$ and $j \in \{h, v\}$ representing the light number and direction respectively. The two lights are not required to mimic each other; thus, in this setting, we have four (not two as in the isolated intersection) light states:

$A_{h,h}$ Light-1 is green for the horizontal street. This state allows both departures and arrivals at the horizontal street of Light-1, but only arrivals at the vertical street of Light-1. Also, Light-2 is green for the horizontal street. Thus, both departures and arrivals are allowed at the horizontal street of Light-2, but only arrivals at the vertical street of Light-2.

$A_{h,v}$ The light is green for the horizontal street of light 1. This state allows both departures and arrivals at the horizontal street of light 1, but only arrivals at the vertical street of light 1. Also, the light is green for the vertical street of light 2. Thus, both departures and arrivals are allowed at the vertical street of light 2, but only arrivals at the horizontal street of light 2.

$A_{v,h}$ Light-1 is green for the vertical street. This state allows both departures and arrivals at the vertical street of Light-1, but only arrivals at the horizontal street of Light-1. Also, Light-2 is green for the horizontal street. Thus, both departures and arrivals are allowed at the horizontal street of Light-2, but only arrivals at the vertical street of Light-2.

$A_{v,v}$ Light-1 is green for the vertical street. This state allows both departures and arrivals at the vertical street of Light-1, but only arrivals at the horizontal street of Light-1. Also, Light-2 is green for the vertical street of Light-2. Thus, both departures and arrivals are allowed at the vertical street of Light-2, but only arrivals at the horizontal street of Light-2.

Each of the five queues is described below.

- $L_{1,h}$, represents the cars waiting at the horizontal street of Light-1.

- $L_{1,v}$, represents the cars waiting at the vertical street of Light-1.

- $L_{2,h}$, represents the cars waiting at the horizontal street of Light-2.
- $L_{2,v}$, represents the cars waiting at the vertical street of Light-2.
- L_{tran}, represents the cars traveling from Light-1 to Light-2.

Some of these queues have exogenous arrivals and some have endogenous arrivals. The arrival process for each queue is outlined below.

- $L_{1,h}$: exogenous arrivals with rate $\lambda_{1,h}$;
- $L_{1,v}$: exogenous arrivals with rate $\lambda_{1,v}$;
- $L_{2,h}$: endogenous arrivals originating from L_{tran};
- $L_{2,v}$: exogenous arrivals with rate $\lambda_{2,v}$;
- L_{tran} : endogenous arrivals originating from $L_{1,h}$;

If queue $L_{i,j}$ has exogenous arrivals, then it has associated c.d.f. $G_{i,j}$ and p.d.f. $g_{i,j}$ for the *i.i.d.* interarrival times.

When street (i, j), for $i \in \{1, 2\}$ $j \in \{h, v\}$, has a green light, cars at that street are served one at a time, according to i.i.d. "service times" with mean $1/\mu_{i,j}$, c.d.f. $F_{i,j}$ and p.d.f. $f_{i,j}$. During a red light phase no cars are "served." If a car does not make it through the intersection during a cycle, it must "start over" with a fresh service time during the subsequent green cycle. This is true for all queues except the transient queue. The transient queue behaves differently. This queue contains constantly moving vehicles so we model it as an infinite-server queue. Progression through the queue can not be interrupted, because there is no light governing flow through this queue; therefore servers are never turned off, and departures are always possible whenever the queue is nonempty.

The time to complete a red-green cycle for each intersection is denoted by T_i, where $T_i = T_{i,h} + T_{i,v}$. We assume that the green-red cycle repeats identically and indefinitely, and without loss of generality assume that the sequence begins with a green light for streets $(1, h)$ and $(2, v)$. Once Light-1 turns green for street $(1, h)$, Light-2 turns green for street $(1, h)$ after T_{offset} time has elapsed. As a simplifying assumption, we set $T_1 = T_2$.

The performance measure of interest is the average number of cars waiting in a queue. We define:

$$\begin{aligned}
L_{i,j}(t) &= \text{\# cars waiting on street } (i,j) \text{ at time } t \ (i \in \{1,2\} \text{ and } j \in \{h,v\}); \\
\bar{L}_{i,j}(t) &= \text{average queue length for street } (i,j) \text{ up to time } t \\
&= \frac{1}{t}\int_0^t L_{i,j}(x)dx; \\
L_{tran}(t) &= \text{\# cars in the transient queue at time } t; \\
\bar{L}_{tran}(t) &= \text{average queue length for transient queue up to time } t \\
&= \frac{1}{t}\int_0^t L_{tran}(x)dx; \\
N &= \text{\# red-green cycles simulated;} \\
\bar{L} &= \bar{L}_{1,h}(NT) + \bar{L}_{1,v}(NT) + \bar{L}_{tran}(NT) + \bar{L}_{2,h}(NT) + \bar{L}_{2,v}(NT). \\
&= \text{total average number of cars in the network}
\end{aligned}$$

Due to the fixed nature of T_i, setting $T_{i,h}$ in turn determines the value for $T_{i,v}$; thus, the optimization problem is given by

$$\min_{T_{1,h},T_{offset},T_{2,h}} E[\bar{L}] \tag{3.1}$$

$$\text{subject to} \quad T_{1,h} + T_{1,v} = T_1 \ , \ T_{2,h} + T_{2,v} = T_2 \ \text{ and } \ T_1 = T_2,$$

which we propose to solve by satisfying the first-order condition

$$\nabla_\theta E[\bar{L}] = 0, \tag{3.2}$$

where θ is the vector of controllable variables (parameters). To find the value of θ satisfying (3.2), we use gradient-based simulation optimization via a stochastic approximation recursion of the form (2.4).

The gradient estimate in (2.4) requires estimators for

$$\frac{dE[\bar{L}_{i,j}]}{d\theta}, \quad i = 1,2 \ \text{ and } \ j = h, v \ \text{ and } \ \frac{dE[\bar{L}_{tran}]}{d\theta}. \tag{3.3}$$

3.3 Derivation of Estimators

Following the framework of Fu and Hu [14], the general SPA estimator consists of an IPA term and a conditional term. The general form of the SPA estimator for this setting is given in following equation:

$$\left(\frac{dE[\bar{L}_{i,j}]}{d\theta}\right)_{SPA} = \frac{d\bar{L}_{i,j}}{d\theta} + \lim_{\Delta\theta\to 0} \frac{P_{\mathcal{Z}}(\beta(\Delta\theta))}{\Delta\theta} \lim_{\Delta\theta\to 0} \delta E_{\mathcal{Z}}[\bar{L}_{i,j}(\beta(\Delta\theta))]. \tag{3.4}$$

The subscript $\mathcal{Z}$ denotes a conditioning on the characterization, which is the set of conditioning quantities on the sample path on which the conditional contribution is estimated, and it will differ for each estimator.

In this setting the optimization is with respect to $T_{1,h}$, $T_{2,h}$ and T_{offset}; thus, we take

$$\theta = \begin{bmatrix} T_{1,h} \\ T_{2,h} \\ T_{offset} \end{bmatrix}$$

throughout.

There are 5 queues and 3 controllable parameters, yielding a total of 15 possible estimators. When we take into account direction (e.g., left-hand and right-hand), we are faced with the task of deriving 30 estimators for a network of two signalized intersections.

3.3.1 Estimators for Light-1

For Light-1, we are concerned with queues $L_{1,h}$ and $L_{1,v}$.

Light-1 has all exogenous arrivals. Its queues $L_{1,h}$ and $L_{1,v}$ are not affected by any other queues in the network. For this reason, the corresponding estimators can be extrapolated from the gradient derivation process in Chapter 2 to obtain:

$$\left(\frac{dE[\bar{L}_{1,h}]}{d\theta_1}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N}\frac{f_{1,h}(\alpha_i)}{1-F_{1,h}(\alpha_i)}\left[-R_{Q_i}^{(1,h)}(\gamma_i)\right], \tag{3.5}$$

$$\left(\frac{dE[\bar{L}_{1,v}]}{d\theta}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N}H_i + \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{H_i}\frac{f_{1,v}(\alpha_i^k)}{F_{1,v}(\alpha_i^k)}R_{Y_i^k+A_i^k}^{(1,v)}(\gamma_i), \tag{3.6}$$

$$\left(\frac{dE[\bar{L}_{1,h}]}{d\theta}\right)_{SPA,l} = -\frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i}\frac{f_{1,h}(\alpha_i^k)}{F_{1,h}(\alpha_i^k)}R_{Y_i^k+A_i^k}^{(1,h)}(\gamma_i), \tag{3.7}$$

$$\left(\frac{dE[\bar{L}_{1,v}]}{d\theta}\right)_{SPA,l} = \frac{1}{NT}\sum_{i=1}^{N}H_i + \frac{1}{NT}\sum_{i=1}^{N}\frac{f_{1,v}(\alpha_i)}{1-F_{1,v}(\alpha_i)}\left[R_{Q_i}^{(1,v)}(\gamma_i)\right], \tag{3.8}$$

which correspond to equations (2.16), (2.19), (2.22) and (2.25), respectively.

Light-2 is downstream of Light-1; therefore, changes to Light-2 will not affect the queues of Light-1. Hence, we can conclude that

$$\left(\frac{dE[\bar{L}_{1,h}]}{d\theta_2}\right)_{SPA,r} = \left(\frac{dE[\bar{L}_{1,h}]}{d\theta_2}\right)_{SPA,l} = 0 \tag{3.9}$$

and

$$\left(\frac{dE[\bar{L}_{1,v}]}{d\theta_2}\right)_{SPA,r} = \left(\frac{dE[\bar{L}_{1,v}]}{d\theta_2}\right)_{SPA,l} = 0. \tag{3.10}$$

Similarly, we observe that any changes to T_{offset} will affect Light-2 but not Light-1 so we have

$$\left(\frac{dE[\bar{L}_{1,h}]}{d\theta_3}\right)_{SPA,r} = \left(\frac{dE[\bar{L}_{1,h}]}{d\theta_3}\right)_{SPA,l} = 0 \tag{3.11}$$

and

$$\left(\frac{dE[\bar{L}_{1,v}]}{d\theta_3}\right)_{SPA,r} = \left(\frac{dE[\bar{L}_{1,v}]}{d\theta_3}\right)_{SPA,l} = 0. \tag{3.12}$$

3.3.2 Estimators for Light-2

A change in $T_{1,h}$ will affect Light-2 through the arrival process for queue $L_{2,h}$. While the arrivals to queue $L_{2,h}$ will be changed, there is no effect on queue $L_{2,v}$, so we have

$$\left(\frac{dE[\bar{L}_{2,v}]}{d\theta_3}\right)_{SPA,r} = \left(\frac{dE[\bar{L}_{2,v}]}{d\theta_3}\right)_{SPA,l} = 0. \tag{3.13}$$

Next we consider the two estimators for queue $L_{2,h}$ w.r.t. $T_{1,h}$. We have observed that perturbing $T_{1,h}$, will affect the departures of queue $L_{1,h}$. It is also the case that the arrivals to queue $L_{2,h}$ are the departures from queue $L_{1,h}$; thus, we have that a perturbation of $T_{1,h}$ will affect the arrivals of queue $L_{2,h}$. This is a new type of critical change, significantly different from the critical changes in any of the other estimators we have seen thus far, because previously, only departures were affected by a perturbation.

We now consider street $(2, h)$ with $\Delta\theta_1 > 0$, corresponding to the right-hand estimator for $dE[\bar{L}_{2,h}]/d\theta_1$. In this case ($\Delta\theta_1 = \Delta T_{1,h} > 0$), there is a positive perturbation in the green signal length of street $(1, h)$ while keeping the total signal cycle length, T_1, unchanged. From the Chapter 2 analysis, we have that this perturbation can lead to a possible extra departure. Because the departures from street $(1, h)$ are the arrivals to street $(2, h)$, we have an extra arrival as the critical change. Not only do we gain an extra arrival at the point of perturbation, we also will lose an arrival sometime in the future. The lost arrival is due to the fact that the upstream queue providing arrivals to queue $L_{2,h}$ has been depleted; thus, the upstream queue cannot provide all future arrivals. Another way of looking at it is to notice that one of the arrivals to queue $L_{2,h}$ is occurring early. An example sample path for this perturbation can be seen in Figure 3.2.

To calculate the probability rate and expected effect of this critical change, we condition on all arrivals and service times (for queue $L_{1,h}$) except for the last entry to service during the current $A_{h,j}$ state. Since the critical event change in a cycle is a function of the last entry to service, we can index by cycles. The service time for the last car to enter service in queue $L_{1,h}$ during cycle i is greater than α_i. DNP and PP are then defined by the critical change occurring precisely at the green/red light change, with the service times of the last car to enter service being α_i^+ and α_i^-, respectively. If X denotes a random variable with service time distribution $F_{1,h}$, then the probability of a critical change is given by

$$P(\beta_i(\Delta\theta)) = P(X \leq \alpha_i + \Delta\theta \mid X \geq \alpha_i), \tag{3.14}$$

and hence

$$\lim_{\Delta\theta \to 0} \frac{P(\beta_i(\Delta\theta))}{\Delta\theta} = \frac{f_{1,h}(\alpha_i)}{1 - F_{1,h}(\alpha_i)}. \tag{3.15}$$

Thus, the estimator becomes

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_1}\right)_{SPA,r} = \frac{1}{NT} \sum_{i=1}^{N} \frac{f_{1,h}(\alpha_i)}{1 - F_{1,h}(\alpha_i)} E_{\mathcal{Z}_i}[\bar{L}_{2,h}^{PP_i} - \bar{L}_{2,h}^{DNP_i}]. \tag{3.16}$$

To calculate the resulting expected effect, $E_{\mathcal{Z}}\left[\bar{L}_{2,h}^{PP_i} - \bar{L}_{2,h}^{DNP_i}\right]$, we note that there are three possible sample path outcomes. The three case can be distinguished as follows:

- **Case 1**: the early arrival departs in the same (green) phase in which it arrives;
- **Case 2**: the early arrival does not depart during same phase and queue $L_{1,h}$ empties before queue $L_{2,h}$;
- **Case 3**: the early arrival does not depart during same phase and queue $L_{2,h}$ empties before queue $L_{1,h}$.

For Case 1, we observe that starting at the critical change, $L_{2,h}^{DNP_i}(t)$ will be identical to $L_{2,h}(t)$, whereas $L_{2,h}^{PP_i}(t)$ will be one higher than $L_{2,h}(t)$ until $L_{2,h}(t)$ empties, where we have that $L_{2,h}(t)$ empties during the current green phase. We define

$$\begin{aligned}
Te^i_{2,h} &= \text{epoch of } L_{2,h}(t) \text{ emptying during } i\text{th cycle,}\\
Y^i_{2,h} &= \text{system time of early arrival vehicle during the } i\text{th cycle,}\\
\alpha^i_{2,h} &= \text{duration of time until light change from last entry of service}\\
&\ \text{to queue } L_{2,h} \text{ during } i\text{th cycle,}\\
A^i_{2,h} &= \text{epoch of early arrival to queue } L_{2,h} \text{ during the } i\text{th cycle,}\\
D^i_{2,h} &= \text{epoch at which the early arrival vehicle from the } i\text{th cycle}\\
&\ \text{departs from queue } L_{2,h},\\
e^i_{1,h} &= \text{epoch of first } L_{1,h}(t) \text{ emptying after } i\text{th cycle,}\\
e^i_{2,h} &= \text{epoch of first } L_{2,h}(t) \text{ emptying after } i\text{th cycle,}\\
e^{*i}_{2,h} &= \text{epoch of first } L_{2,h}(t) \text{ emptying after } e^i_{1,h}.
\end{aligned}$$

Thus, we have that $L^{DNP_i}_{2,h}(t) + 1 = L^{PP_i}_{2,h}(t)$ for all t such that $EA^i_{2,h} < t < ED^i_{2,h}$, which is $Y^i_{2,h}$ amount of time. This critical change also results in $L^{DNP_i}_{2,h}(t) = L^{PP_i}_{2,h}(t) + 1$ for all t such that $e^i_{1,h} < t < e^{*i}_{1,h}$. We can now state the expected difference in the two paths as

$$E_{\mathcal{Z}_i}[\bar{L}^{PP_i}_{2,h} - \bar{L}^{DNP_i}_{2,h}] = E[Y^i_{2,h}] - E[e^{*i}_{2,h} - e^i_{1,h}]. \tag{3.17}$$

Calculation of (3.17) can be done online as follows. By letting S be a generic random variable generated from the service time distribution for queue $L_{2,h}$, we have that

$$E[Y^i_{2,h}] \;=\; Te^i_{2,h} - A^i_{2,h} + E[S|S < \alpha^i_{2,h}],$$

where $e^{*i}_{2,h}$ and $A^i_{2,h}$ are both observable from the sample path. Now we have that $A^i_{2,h}, e^i_{1,h}, e^{*i}_{2,h}$ and $e^i_{2,h}$ are all observable from the sample.

For Case 2, we observe that starting at the critical change, $L_{2,h}^{DNP_i}(t)$ will be identical to $L_{2,h}(t)$, whereas $L_{2,h}^{PP_i}(t)$ will be one higher than $L_{2,h}(t)$ until $L_{1,h}(t)$ empties. Thus for Case 2, the expected difference is

$$E_{\mathcal{Z}_i}[\bar{L}_{2,h}^{PP_i} - \bar{L}_{2,h}^{DNP_i}] = E[\min(NT, \inf\{t > \tau \mid L_{1,h}(t) = 0\})] - \tau. \qquad (3.18)$$

For Case 3, we observe that starting at the critical change, $L_{2,h}^{DNP_i}(t)$ will be identical to $L_{2,h}(t)$, whereas $L_{2,h}^{PP_i}(t)$ will be one higher than $L_{2,h}(t)$ until $L_{2,h}(t)$ empties. Then after that, there is an m amount of time that $L_{2,h}^{PP_i}(t)$ could remain one higher than $L_{2,h}(t)$. In order to determine how much of this m_i the two path remain separated, we determine the likelihood of a departure from $L_{2,h}^{PP_i}(t)$ before the time m_i expires. Again, for S a generic service time,we have

$$\begin{aligned} E[\text{time that } L_{2,h}^{PP_i}(t) \text{ remains above } L_{2,h}^{i}(t) \text{ during } m_i] &= E[\min(S, m_i)] \\ &= \sigma_i. \qquad (3.19) \end{aligned}$$

Thus for Case 3, the expected difference is

$$E_{\mathcal{Z}_i}[\bar{L}_{2,h}^{PP_i} - \bar{L}_{2,h}^{DNP_i}] = E[\min(NT, \inf\{t > \tau \mid L_{2,h}(t) = 0\})] - \tau + \sigma_i. \qquad (3.20)$$

The final estimator incorporating all three cases becomes

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_1}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N}\frac{f_{1,h}(\alpha_i)}{1-F_{1,h}(\alpha_i)} \times \begin{cases} \big([e^i_{2,h} - EA^i_{2,h} + E[S|S < \alpha^i_{2,h}]] - [E[e^{*i}_{2,h} - e^i_{1,h}]]\big) \\ \qquad \text{if } (e^i_{2,h} < i * T + T_{offset} + T_{2,h}) \text{ and } (S < \alpha^i_{2,h}) \\ (E[\min(NT, \inf\{t > \tau \mid L_{1,h}(t) = 0\})] - \tau) \\ \qquad \text{if } e^i_{1,h} < e^i_{2,h} \\ (E[\min(NT, \inf\{t > \tau \mid L_{2,h}(t) = 0\})] - \tau + \sigma_i) \\ \qquad \text{otherwise.} \end{cases} \tag{3.21}$$

Note that again the sign of the estimator is unknown, which makes intuitive sense, because it cannot be determined a priori if the change from the original arrival time to the early arrival will be beneficial or detrimental to the system.

For the last estimator for Light-2, we consider queue $L_{2,h}$ with $\Delta\theta_1 < 0$, corresponding to the left-hand estimator for $dE[\bar{L}_{2,h}]/d\theta_1$. In this case ($\Delta\theta_1 = \Delta T_{1,h} < 0$), there is a negative perturbation in the green signal length of street $(1, h)$ while keeping the total signal cycle length, T_1, unchanged. From the Chapter 2 analysis that perturbation can lead to a possible loss of departure. And because the departures from street $(1, h)$ are the arrivals to street $(2, h)$, we have a lost arrival as the critical change. Because all other factors remain unchanged, not only do we gain an arrival at the point of perturbation, we also will gain an arrival sometime in the future. The gained arrival is due to the fact that the upstream queue providing arrivals to queue $L_{2,h}$ still has the car in it and thus must provide that arrival sometime in the future. Another way of looking at it is by noticing that one of the arrivals to queue $L_{2,h}$ is delayed.

To calculate the probability rate and expected effect of these possible critical changes, we condition on all arrival times and all service times except that of the kth departure (for queue $L_{1,h}$). If X denotes a random variable with service time distribution $F_{1,h}$, then the probability rate of a critical change is given by

$$\lim_{\Delta\theta\to 0} \frac{P(\beta_i^k(\Delta\theta))}{\Delta\theta} = \lim_{\Delta\theta\to 0} \frac{P(X \geq \alpha_i^k - \Delta\theta \mid X \leq \alpha_i^k)}{\Delta\theta} = \frac{f_{1,h}(\alpha_i^k)}{F_{1,h}(\alpha_i^k)}. \tag{3.22}$$

Then the estimator becomes

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_1}\right)_{SPA,l} = -\frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i} \frac{f_{1,h}(\alpha_i^k)}{F_{1,h}(\alpha_i^k)} E_{\mathcal{Z}_i^k}[\bar{L}_{2,h}^{PP_i^k} - \bar{L}_{2,h}^{DNP_i^k}]. \tag{3.23}$$

To calculate the resulting expected effect, $E_{\mathcal{Z}}\left[\bar{L}_{2,h}^{PP_i} - \bar{L}_{2,h}^{DNP_i}\right]$, we observe that starting at the critical change, $L_{2,h}^{DNP_i}(t)$ will be one above $L_{2,h}^{PP_i}(t)$ until either the time of the delayed arrival occurs or $L_{2,h}^{DNP_i}(t)$ empties. Thus, we have that $L_{2,h}^{DNP_i}(t) = L_{2,h}^{PP_i}(t) + 1$ for all t such that $p_1^i < t < p_2^i$, where we define

$$\begin{aligned} p_1^i &= \text{epoch of lost arrival during } i\text{th cycle} \\ &= i * T_{1,h}, \\ p_2^i &= \text{first occurrence of time when delayed arrival occurs or queue } L_{2,h} \text{ empties} \\ &= \min\{t_1, t_2 | L_{1,h}(t_1) = 0 \quad \text{and} \quad L_{2,h}(t_2) = 0\}. \end{aligned}$$

After time p_2^i, we observe that, $L_{2,h}^{DNP_i}(t)$ will be one below $L_{2,h}^{PP_i}(t)$ until $L_{2,h}^{PP_i}(t)$ empties. Thus, we have that $L_{2,h}^{DNP_i}(t) = L_{2,h}^{PP_i}(t) - 1$ for all $p_2^i < t < p_3^i$, where we define

$$p_3^i = \min\{NT, t | L_{2,h}(t) = 0 \quad \text{and} \quad t > T_{1,h}\}.$$

Thus,

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_1}\right)_{SPA,l} = \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i}\frac{f_{1,h}(\alpha_i^k)}{F_{1,h}(\alpha_i^k)}\left([p_2^i - p_1^i] - [p_3^i - p_2^i]\right). \qquad (3.24)$$

The times p_2^i and p_3^i are not observable from the sample path; however, they can be estimated offline. Namely, $[p_2^i - p_1^i]$ is the time it takes for queue $L_{1,h}$ to empty, given the system condition at time $i * T_{1,h}$. Similarly, $[p_3^i - p_2^i]$ is the time it takes for queue $L_{2,h}$ to empty, given the system condition at time p_2^i. By defining

$$\begin{aligned} \zeta(t) &= \text{system conditions at time } t, \\ \kappa^{(i,j)}(c) &= \text{time to empty queue } L_{i,j} \text{ given system conditions, c} \end{aligned}$$

our final estimator becomes

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_1}\right)_{SPA,l} = \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i}\frac{f_{1,h}(\alpha_i^k)}{F_{1,h}(\alpha_i^k)}\left[\kappa^{(1,h)}(\zeta(i * T_{1,h})) - \kappa^{(2,h)}(\zeta(p_2^i))\right]. \quad (3.25)$$

Note that again the sign of the estimator is unknown, which makes intuitive sense, because it cannot be determined ahead of time if the change from the original arrival time to the delayed arrival will be beneficial or detrimental to the system. For Light 2, we can derive the gradient estimators w.r.t $\theta_2 = T_{2,h}$ by following the gradient derivation process described in Chapter 2. Therefore, corresponding to equations 2.16, 2.19, 2.22 and 2.25, we have

$$\begin{aligned} \left(\frac{dE[\bar{L}_{2,h}]}{d\theta_2}\right)_{SPA,r} &= \frac{1}{NT}\sum_{i=1}^{N}\frac{f_{2,h}(\alpha_i)}{1 - F_{2,h}(\alpha_i)}\left[-R_{Q_i}^{(2,h)}(\gamma_i)\right], & (3.26) \\ \left(\frac{dE[\bar{L}_{2,v}]}{d\theta_2}\right)_{SPA,r} &= \frac{1}{NT}\sum_{i=1}^{N}H_i + \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{H_i}\frac{f_{2,v}(\alpha_i^k)}{F_{2,v}(\alpha_i^k)}R_{Y_i^k + A_i^k}^{(2,v)}(\gamma_i), & (3.27) \\ \left(\frac{dE[\bar{L}_{2,h}]}{d\theta_2}\right)_{SPA,l} &= -\frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i}\frac{f_{2,h}(\alpha_i^k)}{F_{2,h}(\alpha_i^k)}R_{Y_i^k + A_i^k}^{(2,h)}(\gamma_i), & (3.28) \\ \left(\frac{dE[\bar{L}_{2,v}]}{d\theta_2}\right)_{SPA,l} &= \frac{1}{NT}\sum_{i=1}^{N}H_i + \frac{1}{NT}\sum_{i=1}^{N}\frac{f_{2,v}(\alpha_i)}{1 - F_{2,v}(\alpha_i)}\left[R_{Q_i}^{(2,v)}(\gamma_i)\right]. & (3.29) \end{aligned}$$

When considering street 2, it is important to note that the green phase is actually split into two periods during a T_2 long cycle. For example, Light-2 starts red for the horizontal street; then after T_{offset} time, it changes green and remains that way for $T_{2,h}$ seconds before it changes again to red, at which point it will remain that way for the remainder of the cycle ($T_{2,v} - T_{offset}$ more seconds). We now consider queue $(2, h)$ with $\Delta\theta_3 > 0$, corresponding to the right-hand estimator for $dE[\bar{L}_{2,h}]/d\theta_3$. In this case, there is a forward shift of the green signal phase for street $(2, h)$, while keeping the total signal cycle length, T_2, unchanged. In essence, the epoch of the light change from red to green for street $(2, h)$ is delayed as well as the epoch of the light change from green to red. This perturbation will cause a delay from state $A_{i,v}$ to $A_{i,h}$, where the value of i is inconsequential to this derivation. This delay will cause each car in the IBP to enter service later, and subsequently depart later. That is, there will be a positive IPA contribution from each car in the IBP. This perturbation also causes a delay from state $A_{i,h}$ to $A_{i,v}$. Small perturbations at the end of $T_{2,h}$ do not affect the departure times of cars from street $(2, h)$; thus, the IPA contribution from this aspect of the perturbation is zero. Because the light turns red late, some cars will have an opportunity to exit when they did not in the NP. It follows that the critical change in this case is this additional departure. The perturbation does not propagate through an idle period; thus, an additional departure is possible if and only if the queue is nonempty at the light change and the car in service is not a member of the IBP. When these two criteria hold, the last car to enter service is the only candidate for a critical change because the probability of more than one critical change is of higher order; therefore, it can be ignored. Thus, we only consider the last car to enter service as a possible extra departure, and then only when that car is not a member of the IBP. An example of this perturbation can be seen in Figure 3.3. We define the event:

$$\phi_i = \text{Event that the last car to enter service during the } i\text{th cycle is an IBP car} \tag{3.30}$$

and with an analysis similar to those presented in the Chapter 2, we get that the final estimator becomes

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_3}\right)_{SPA,r} = \frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{1}{NT}\sum_{i=1}^{N} \frac{f_{2,h}(\alpha_i)}{1 - F_{2,h}(\alpha_i)} \left[-R_{Q_i}^{(2,h)}(\gamma_i)\right] \mathbf{1}(\phi_i). \tag{3.31}$$

Note that we cannot tell the sign of the estimator. This makes intuitive sense, because a shift of the green signal phase could have positive or negative effects.

We now consider queue $(2, h)$ with $\Delta\theta_3 < 0$, corresponding to the left-hand estimator for $dE[\bar{L}_{2,h}]/d\theta_3$. In this case there is a backward shift of the green signal phase for street $(2, h)$ while keeping the total signal cycle length, T_2, unchanged. In essence, the epoch of the light change from red to green street $(2, h)$ is expedited as well as the epoch of the light change from green to red. This perturbation will cause hastened transition from state $A_{i,v}$ to $A_{i,h}$. This early transition will cause each car in the IBP to enter service early and thus depart early. That is, there will be a negative IPA contribution from each car in the IBP. This perturbation also causes an early transition from state $A_{i,h}$ to $A_{i,v}$. Small perturbations at the end of $T_{2,h}$ do not affect the departure times of cars from street $(2, h)$; thus, the IPA contribution from that aspect of the perturbation is zero. Because the light turns red early, some cars that were able to exit the queue in the NP will lose that opportunity. Thus, the critical change in this case is this lost departure. Every car that successfully completes service represents a potential critical change, provided that car was not a member of the IBP. This distinction is needed because a car in the IBP does not lose any green light time; thus, there is no chance of losing that departure. Defining

$$\bar{H}_i \;=\; \text{number of non-IBP departures during } i\text{th cycle},$$

and using an analysis similar to that in Chapter 2, the final estimator becomes

$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_3}\right)_{SPA,l} = -\frac{1}{NT}\sum_{i=1}^{N} H_i + \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{\bar{H}_i}\frac{f_{2,h}(\alpha_i^k)}{F_{2,h}(\alpha_i^k)}R^{(2,h)}_{Y_i^k+A_i^k}(\gamma_i). \qquad (3.32)$$

The sign of the estimator is not necessarily positive or negative, which makes intuitive sense, because a shift of the green signal phase could have a net positive or negative effect.

We now consider queue $(2, v)$ with $\Delta\theta_3 < 0$, corresponding to the left-hand estimator for $dE[\bar{L}_{2,v}]/d\theta_3$. In this case, there is a backward shift of the red signal phase for street $(2, v)$, while keeping the total signal cycle length, T_2, unchanged. In essence, the epoch of the light change from green to red for street $(2, v)$ is expedited, as is the epoch of the light change from red to green. This perturbation will cause a hastened transition from state $A_{i,v}$ to $A_{i,h}$. Each car that departs during the first portion of the green phase for street $(2, v)$ of the current cycle has a chance of no longer departing because the first portion of the green light phase has been shortened. Thus, the critical change from this aspect of the perturbation is the loss of a departure. This perturbation also causes an early transition from state $A_{i,h}$ to $A_{i,v}$. In the second portion of the green phase for street $(2, v)$, the early transition will cause all cars in the IBP to enter service early and thus depart the queue early. That is, each departing car that is a member of the IBP will have a negative IPA contribution. Because these IBP cars are departing early, there is chance for an extra departure. Thus, another critical change is possible in the form of an additional departure if the last car to enter service is a member of the IBP. Because the green phase for street $(2,v)$ is split in each cycle, the gradient estimator contains for the first time, three contributing factors, two of which are SPA contributions due to critical changes. We must differentiate between the two green phases, so we define

$$\begin{aligned} H_i^{1st} &= \text{number of IBP departures during the first portion of the} \\ & \text{green phase of } i\text{th cycle,} \\ H_i^{2nd} &= \text{number of IBP departures during the second portion of the} \\ & \text{green phase of } i\text{th cycle,} \end{aligned}$$

and the final estimator becomes

$$\begin{aligned} \left(\frac{dE[\bar{L}_{2,v}]}{d\theta_3}\right)_{SPA,l} &= -\frac{1}{NT}\sum_{i=1}^{N} H_i^{2nd} - \frac{1}{NT}\sum_{i=1}^{N}\frac{f_{2,v}(\alpha_i)}{1-F_{2,v}(\alpha_i)}\left[R_{Q_i}^{(2,v)}(\gamma_i)\right]\mathbf{1}(\alpha_i) \\ &\quad +\frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i}\frac{f_1(\alpha_i^k)}{F_1(\alpha_i^k)}R^{(2,v)}_{Y_i^k+A_i^k}(\gamma_i). \qquad (3.33) \end{aligned}$$

The sign of the estimator is not necessarily positive or negative, which makes intuitive sense, because a shift of the green signal phase could have a net positive or negative effect.

We now consider queue $(2,v)$ with $\Delta\theta_3 > 0$, corresponding to the right-hand estimator for $dE[\bar{L}_{2,v}]/d\theta_3$. In this case, there is a forward shift of the red signal phase for street $(2,v)$, while keeping the total signal cycle length, T_2, unchanged. In essence, the epoch of the light change from the first portion of the green to red for street $(2,v)$ is delayed, as is the epoch of the light change from red to the second portion of the green. This perturbation will cause a delayed transition from state $A_{i,v}$ to $A_{i,h}$. Each car that departs during the first portion of the green phase for street $(2,v)$ of the current cycle has more time to depart, because the first portion of the green light phase has been lengthened. Thus, the critical change from this aspect of the perturbation is an additional departure. This perturbation also causes a delayed transition from state $A_{i,h}$ to $A_{i,v}$. In the second portion of the green phase for street $(2,v)$, the late transition will cause all cars in the IBP to enter service late and thus depart the queue late. That is, each departing car that is a member of the IBP will have a positive IPA contribution. Because these IBP cars are departing late, there is chance that each departure could be lost. Thus, another critical change is possible for each departing car that is a member of the IBP, in the form of a lost departure. Because the green phase for street $(2,v)$ is split into two parts for each cycle, we again have three contributing factors to the gradient estimator. Thus, the final estimator becomes

$$\begin{aligned}\left(\frac{dE[\bar{L}_{2,v}]}{d\theta_3}\right)_{SPA,r} &= \frac{1}{NT}\sum_{i=1}^{N} H_i^{2nd} + \frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{H_i^{2nd}} \frac{f_{2,v}(\alpha_i^k)}{F_{2,v}(\alpha_i^k)} R^{(2,v)}_{Y_i^k+A_i^k}(\gamma_i) \\ &+ \frac{1}{NT}\sum_{i=1}^{N} \frac{f_{2,v}(\alpha_i)}{1-F_{2,v}(\alpha_i)}\left[-R^{(2,v)}_{Q_i}(\gamma_i)\right]. \qquad (3.34)\end{aligned}$$

The sign of the estimator is not necessarily positive or negative, which makes intuitive sense, because a shift of the green signal phase could have a net positive or negative effect.

3.3.3 Estimators for Transient Queue

For $T_{1,h}$, we can see that it will affect queue L_{tran}. This is evident because the arrivals to queue L_{tran} are the departures from queue $L_{1,h}$, and by perturbing $T_{1,h}$, the departures of queue $L_{1,h}$ are affected. We now consider queue L_{tran} with $\Delta\theta_1 > 0$, corresponding to the right-hand estimator for $dE[\bar{L}_{tran}]/d\theta_1$. In this case ($\Delta\theta_1 = \Delta T_{1,h} > 0$), there is a positive perturbation in the green signal length of street $(1, h)$, while keeping the total signal cycle length, T_1, unchanged. In Chapter 2, we determined that this type of perturbation will cause a possible extra departure from queue $L_{1,h}$, and as with queue $L_{2,h}$, we have that an extra departure from queue $L_{1,h}$ results in an extra arrival to queue L_{tran}. Because queue $L_{1,h}$ is now depleted, there will be a lost arrival later. That is, this perturbation results in a early arrival. For cycle i, the extra (early) arrival occurs at time $i * T_{1,h}$, and the future lost arrival occurs when queue $L_{1,h}$ empties. The probability of this critical change, the early arrival, is the same as the probability of the extra departure from queue $L_{1,h}$, which we have calculated before. Thus, for completion of the estimator, we need to calculate the expected difference between $\bar{L}_{tran}^{PP_i^k}$ and $\bar{L}_{tran}^{DNP_i^k}$. We first observe that the $\bar{L}_{tran}^{DNP_i^k}$ will be identical to the $\bar{L}_{tran}$ throughput. Next we observe that the $\bar{L}_{tran}^{PP_i^k}$ will differ from the $\bar{L}_{tran}$ in two places. The first difference occurs at the early arrival point. This early arrival will cause the $\bar{L}_{tran}^{DNP_i^k}$ to be one above the $\bar{L}_{tran}$ for as long as the arrival is in the system. The transient queue is an infinite-server queue, so the time in system of the arrival is exactly equal to its service time. The second difference between the two paths occurs at the original arrival time, that is, the arrival time in the NP. This difference will cause $\bar{L}_{tran}^{PP_i^k}$ to be one below the $\bar{L}_{tran}$ until the original departure time, that is, the departure time in the NP. By using the property of an infinite-server queue, we have that this time, which is the time in system, is exactly equal to the service time. An example sample path for this perturbation can be seen in Figure 3.4.

By defining

$$\begin{aligned} S^i_{early} &= \text{service time for car when it arrives early} \\ &\quad\ \text{service time for car during PP,} \\ S^i_{original} &= \text{service time for car when it arrives at it's original time} \\ &\quad\ \text{service time for car during NP,} \end{aligned}$$

the estimator is given by

$$\left(\frac{dE[\bar{L}_{tran}]}{d\theta_1}\right)_{SPA,r} = -\frac{1}{NT}\sum_{i=1}^{N}\sum_{k=1}^{D_i}\frac{f_{1,h}(\alpha_i^k)}{F_{1,h}(\alpha_i^k)}E[S^i_{early} - S^i_{original}]. \tag{3.35}$$

The service times are all independent, so we have

$$\begin{aligned} E[S^i_{early} - S^i_{original}] &= E[S^i_{early}] - E[S^i_{original}] \\ &= \frac{1}{\mu_{2,h}} - \frac{1}{\mu_{2,h}} \\ &= 0, \end{aligned}$$

and it follows that our final estimator becomes

$$\left(\frac{dE[\bar{L}_{tran}]}{d\theta_1}\right)_{SPA,r} = 0. \tag{3.36}$$

Our final estimator is 0, which makes intuitive sense, because for an infinite server queue, a mere shifting of arrival times will have no effect on the average number in system.

We now consider queue L_{tran} with $\Delta\theta_1 < 0$, corresponding to the left-hand estimator for $dE[\bar{L}_{tran}]/d\theta_1$. With no further analysis, we have that

$$\left(\frac{dE[\bar{L}_{tran}]}{d\theta_1}\right)_{SPA,l} = 0. \tag{3.37}$$

Because the transient queue is upstream of Light-2, it is not affected by parameter $T_{2,h}$; thus, the corresponding estimators are 0, that is

$$\begin{aligned}\left(\frac{dE[\bar{L}_{tran}]}{d\theta_2}\right)_{SPA,r} &= 0, & (3.38)\\ \left(\frac{dE[\bar{L}_{tran}]}{d\theta_2}\right)_{SPA,l} &= 0, & (3.39)\end{aligned}$$

T_{offset} is a parameter that only affects Light-2; thus, its effect on the transient queue is nonexistent. We have

$$\begin{aligned}\left(\frac{dE[\bar{L}_{tran}]}{d\theta_3}\right)_{SPA,r} &= 0, & (3.40)\\ \left(\frac{dE[\bar{L}_{tran}]}{d\theta_3}\right)_{SPA,l} &= 0. & (3.41)\end{aligned}$$

3.4 Numerical Results

Table 3.1 provides a recap of the gradient estimator situation for a network of two signalized intersections. In Table 3.1, the parameters are across the top and the queues are down the side. Each contains one of the following marks signifying the type of estimator needed:

$\star$: Complete gradient estimator required and derived.

X : No gradient estimator required because no variable dependence.

0 : Gradient estimator exists but is equal to 0.

$Queue$	$w.r.t.\ T_{1,h}$	$w.r.t.\ T_{2,h}$	$w.r.t.\ T_{offset}$
$L_{1,h}$	$\star$	**X**	**X**
$L_{1,v}$	$\star$	**X**	**X**
$L_{2,h}$	$\star$	$\star$	$\star$
$L_{2,v}$	**X**	$\star$	$\star$
L_{tran}	0	**X**	**X**

Table 3.1: Table of gradient estimator requirements for each queue and parameter combination.

We can see that 8 estimators are required for a network of two signalized intersection traffic system. Including both left-hand and right-hand estimators brings the total to 16 estimators.

We chose to implement the most efficient estimators for the network of two signalized intersection setting, those that do not require any offline simulation. Thus, we have the following list of implemented gradient estimators:

$$\left(\frac{dE[\bar{L}_{1,h}]}{d\theta_1}\right)_{SPA,r}, \left(\frac{dE[\bar{L}_{1,v}]}{d\theta_1}\right)_{SPA,l}, \left(\frac{dE[\bar{L}_{tran}]}{d\theta_1}\right)_{SPA,r}, \left(\frac{dE[\bar{L}_{2,h}]}{d\theta_1}\right)_{SPA,r},$$
$$\left(\frac{dE[\bar{L}_{2,h}]}{d\theta_2}\right)_{SPA,r}, \left(\frac{dE[\bar{L}_{2,v}]}{d\theta_2}\right)_{SPA,r}, \left(\frac{dE[\bar{L}_{2,h}]}{d\theta_3}\right)_{SPA,r}, \left(\frac{dE[\bar{L}_{2,v}]}{d\theta_3}\right)_{SPA,r}.$$

Table 3.2 shows the settings for each case we used in this setting. For each case, $T_1 = T_2 = T = 60$.

Name	$\lambda_{1,h}$	$\lambda_{1,v}$	$\lambda_{2,v}$	$\mu_{1,h}$	$\mu_{1,v}$	$\mu_{2,h}$	$\mu_{2,v}$
Symmetric case (SC)	$\frac{1}{4.5}$	$\frac{1}{4.5}$	$\frac{1}{4.5}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
Near symmetric case (NC)	$\frac{1}{4.5}$	$\frac{1}{4.5}$	$\frac{1}{4.5}$	$\frac{1}{2}$	$\frac{1}{8.4}$	$\frac{1}{2}$	$\frac{1}{8.4}$
Asymmetric case (AC)	$\frac{1}{4.5}$	$\frac{1}{4.5}$	$\frac{1}{4.5}$	$\frac{1}{2}$	$\frac{1}{18}$	$\frac{1}{2}$	$\frac{1}{18}$

Table 3.2: System parameter specification for simulation cases.

Estimator	$dE[\bar{L}_{tran}]/d\theta_1$	$dE[\bar{L}_{2,h}]/d\theta_2$	$dE[\bar{L}_{2,v}]/d\theta_3$
SPA	0.0000 (0.00001)	-2.279 (0.0226)	0.0008 (0.0006)
FD (.05)	0.0002 (0.00009)	-2.297 (0.0938)	-0.01804 (0.0742)

Table 3.3: Select gradient estimator simulation results for the network of two signalized intersections traffic setting for "SC" (standard error of estimators in parentheses).

Gradient

In Tables 3.3 and 3.4, we show some of the results for the network of two signalized intersections traffic setting. The difference size used for the FD estimators is noted in parentheses on the data tables. We simulated 1000 cycles and performed 1000 replications.

The SPA estimators and FD estimators all have overlapping confidence intervals, but the SPA estimators have a smaller standard error, suggesting they are more accurate.

Estimator	$dE[\bar{L}_{2,v}]/d\theta_2$	$dE[\bar{L}_{2,h}]/d\theta_3$	$dE[\bar{L}_{2,v}]/d\theta_3$
SPA	0.034 (0.00003)	0.0135 (0.0007)	-0.00008 (0.00001)
FD (.05)	0.034 (0.00015)	0.0541 (0.0471)	-0.0039 (0.0016)

Table 3.4: Select gradient estimator simulation results for the network of two signalized intersections traffic setting for "AC" (standard error of estimators in parentheses).

3.5 Optimization

For the network of two signalized intersection traffic problem, we again set appropriate regions of stability in which the SA algorithm is used. We used the estimators derived in this chapter to optimize the system, and the results are compared to those using FDSA and simultaneous perturbation stochastic approximation. The results of the SA algorithms for the SC, NC and AC cases can be seen in Figures 3.5, 3.7 and 3.6, respectively. We observe that all the estimators reach the desired minimum. Of note is the fact that the SPA driven optimization reaches the optimal parameter setting as quickly as the FDSA but is more computationally efficient. And while the SPSA method is comparable to the SPA driven optimization as far as computational efficiency is concerned, the SPA driven optimization reaches the optimal setting in fewer SA iterations.

3.6 Conclusions

Again we see that these derived estimators fair favorably when compared to FD estimates. The SPA-driven optimization had the best overall performance of all the methods. The estimators for the network of two signalized intersections are more complicated than those for the isolated intersection; however, we also note that we were able to use some of the analysis from the isolated intersection case for the network. We conjecture that this would be similar for larger networks.

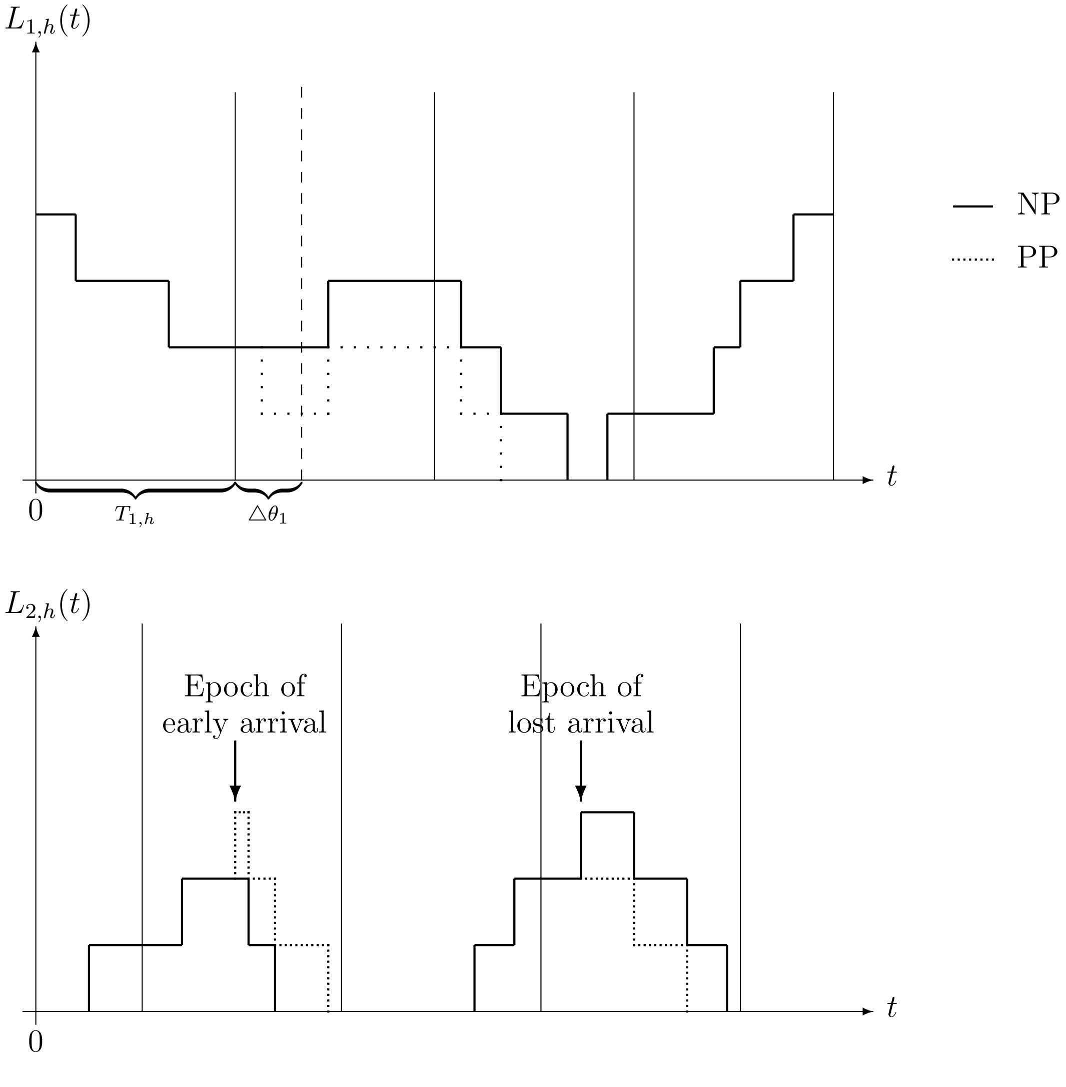


Figure 3.2: Example of $L_{2,h}(t)$ sample path for a network of two-signalized intersections traffic system with positive perturbation of $T_{1,h}$ ($\triangle\theta_1 > 0$).

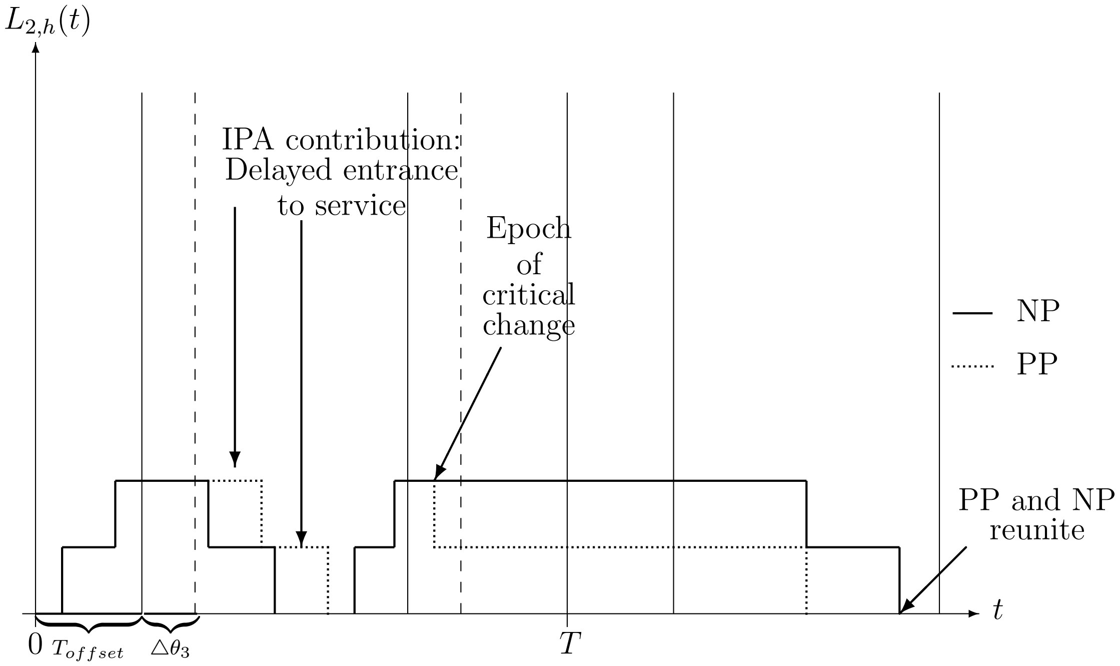


Figure 3.3: Example of $L_{2,h}(t)$ sample path for a network of two signalized intersections traffic system with positive perturbation of T_{offset} ($\triangle\theta_3 > 0$).

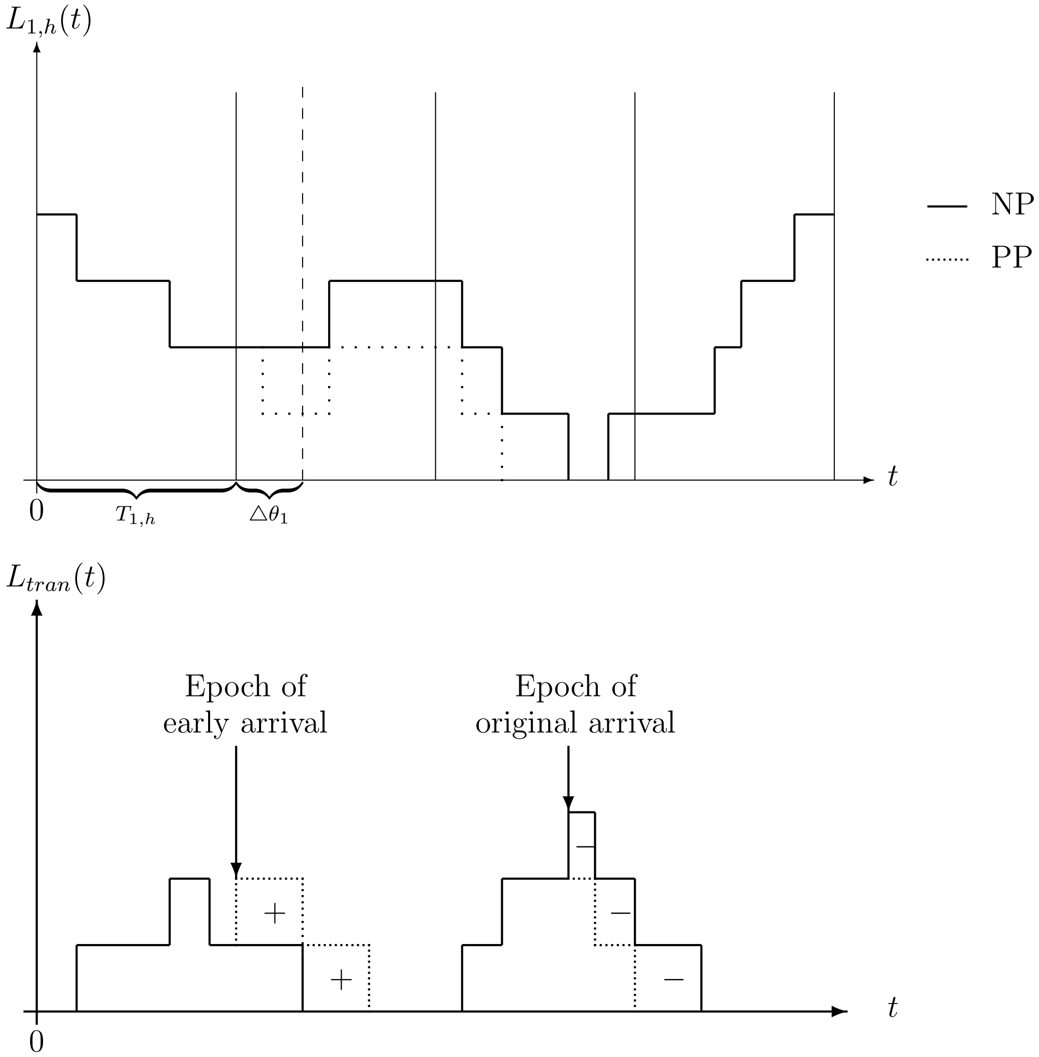
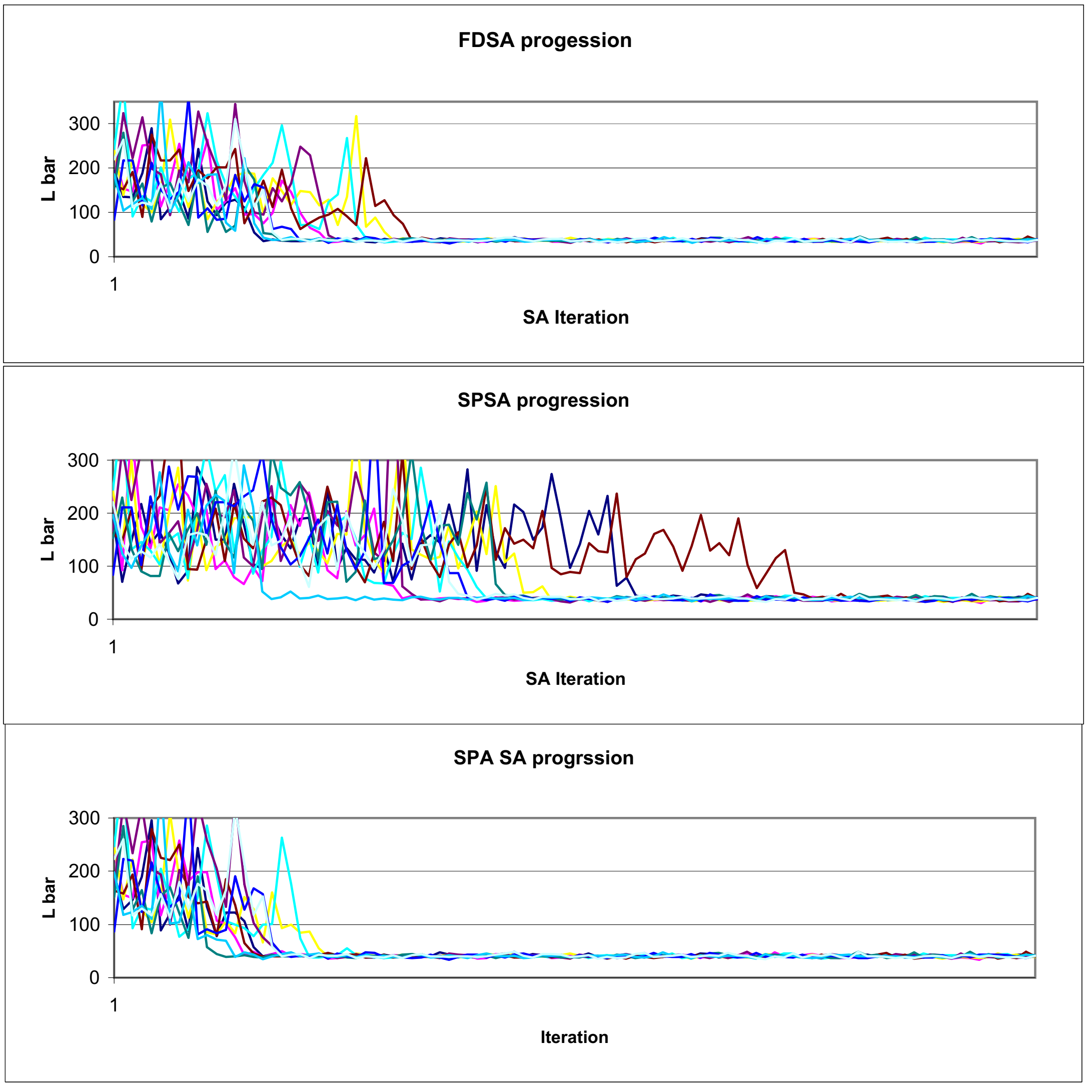


Figure 3.4: Example of $L_{tran}(t)$ sample path for a network of two signalized intersections traffic system with positive perturbation of $T_{1,h}$ ($\triangle\theta_1 > 0$), where "+" represents positive contribution and "-" represents negative contribution.



FDSA progession

SPSA progression

SPA SA progrssion

Figure 3.5: Convergence to minimum for 10 replications of the SA algorithm for "SC" for three different gradient estimation methods: FDSA, SPSA and SPA SA.

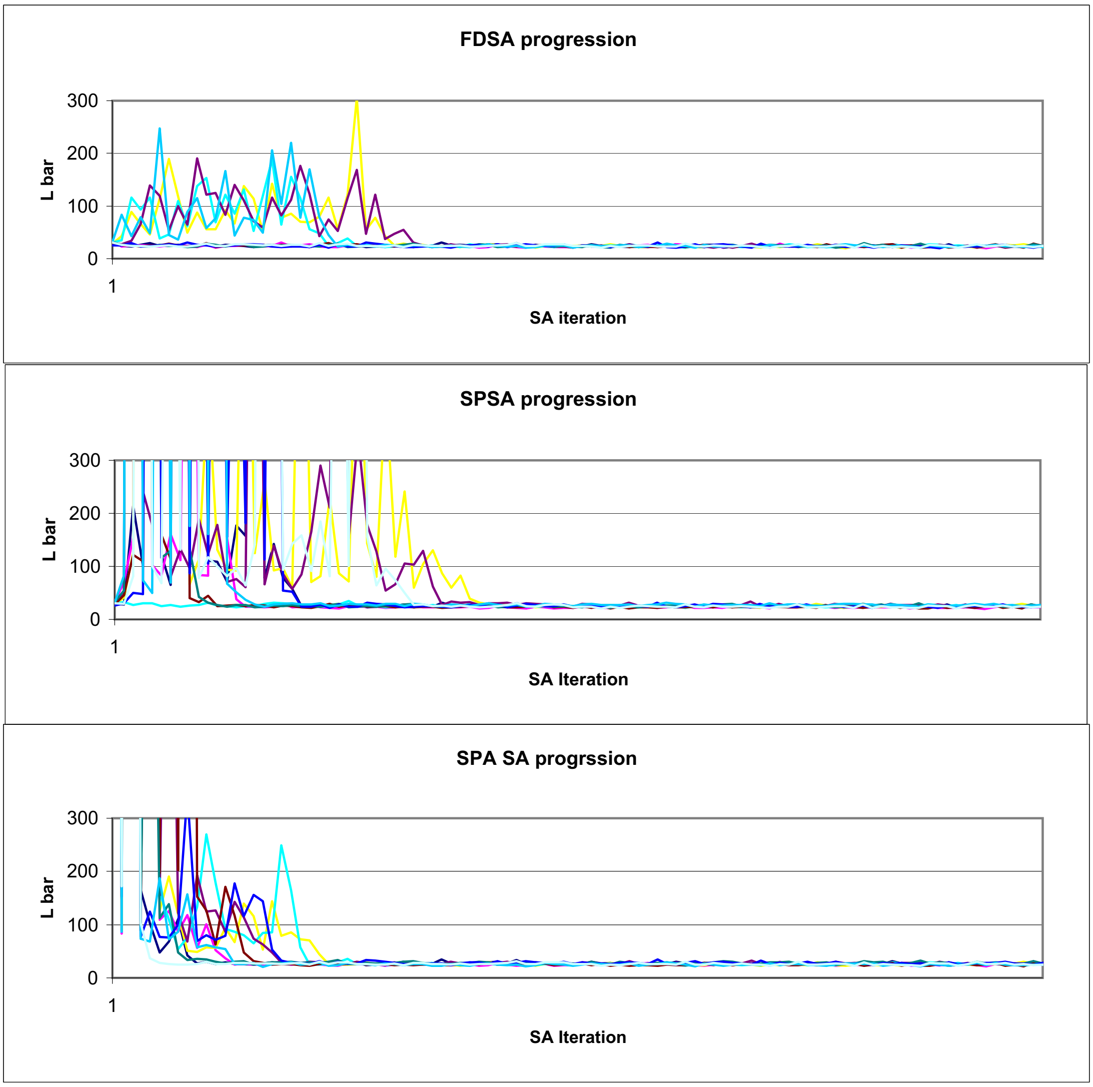


Figure 3.6: Convergence to minimum for 10 replications of the SA algorithm for "AC" for three different gradient estimation methods: FDSA, SPSA and SPA SA.

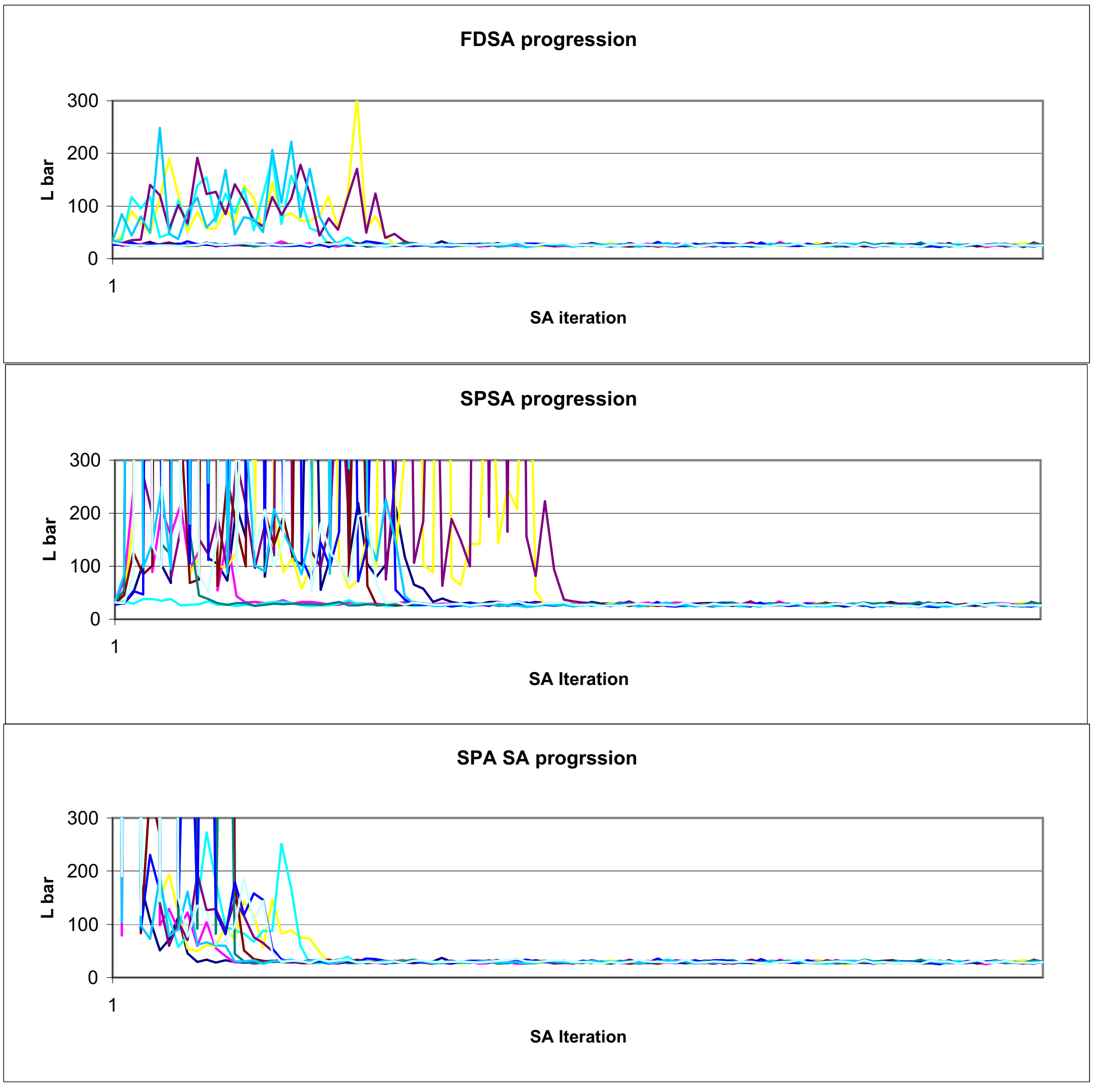


Figure 3.7: Convergence to minimum for 10 replications of the SA algorithm for "NC" for three different gradient estimation methods: FDSA, SPSA and SPA SA.

Chapter 4

Stochastic Fluid Models

An alternative modeling paradigm to a discrete-event queueing formulation is the use of fluid models. One justification for a fluid model is the differing roles that random phenomena may play on different time scales. When the variations on the faster time scale have less impact than those on the slower time scale, the use of fluid models is justified [5]. Though stochastic fluid models (SFMs) might not be very accurate for *performance evaluation*, they have proven to be very robust with respect to *optimization*, because they seem to capture the salient features of the problem. Several authors have reported that use of SFM efficiently lead to optimal or near-optimal solutions [5, 49, 50]. In this approach, we derive a gradient estimator for the performance measure of interest with respect to the control parameters of interest using the SFM, but implement them in the discrete-event simulation using standard stochastic approximation algorithms to determine the optimal parameter setting. This approach has some very important advantages.

- The gradient estimation is done *on-line*; thus, the approach can be implemented on the traffic light controller, and as operating conditions change, it will aim at *continuously* seeking to optimize a generally time-varying performance metric (this holds for both SPA and SFM-based estimators).
- Unlike the SPA estimators, SFM-based estimators do *not* require any knowl-

edge of the system's underlying stochastic processes.

- SFM-based IPA estimators are generally simpler to implement than SPA.
- SPA estimators are generally more accurate than the SFM-based IPA estimators, but simulation results indicated that in optimization problems, they sometimes perform equally well.

4.1 Isolated Intersection

In this chapter, we let $x_q(t;\theta)$, $q = \{(1,h),(1,v),(2,h),(2,v)\}$ denote the fluid buffer content of each queue in the interval $t \in [0,S]$, and we define the sample functions

$$Q_q(\theta) = \frac{1}{S}\int_0^S x_q(t;\theta)dt. \tag{4.1}$$

We then derive sample derivatives of $Q_q(\theta)$ with respect to θ using two different SFMs. $\bar{L}_q$ and Q_q correspond to the queue levels at street q of the stochastic discrete-event and stochastic fluid models, respectively. We take Q_q as an approximation of $\bar{L}_q$. Using infinitesimal perturbation analysis, $\frac{dQ_q}{d\theta}$ is derived. These IPA estimators of $\frac{dQ_q}{d\theta}$ are used to approximate $\frac{dE[\bar{L}_q(t)]}{d\theta}$.

4.1.1 Continuous Model

Figure 4.1 shows the equivalent queueing model, where the processing capacity of the server ($\beta_1(t)$) is divided between the two queues with proportions θ_1 and $(1-\theta_1)$. Figure 4.2 shows a typical sample path of the system. Clearly, this model

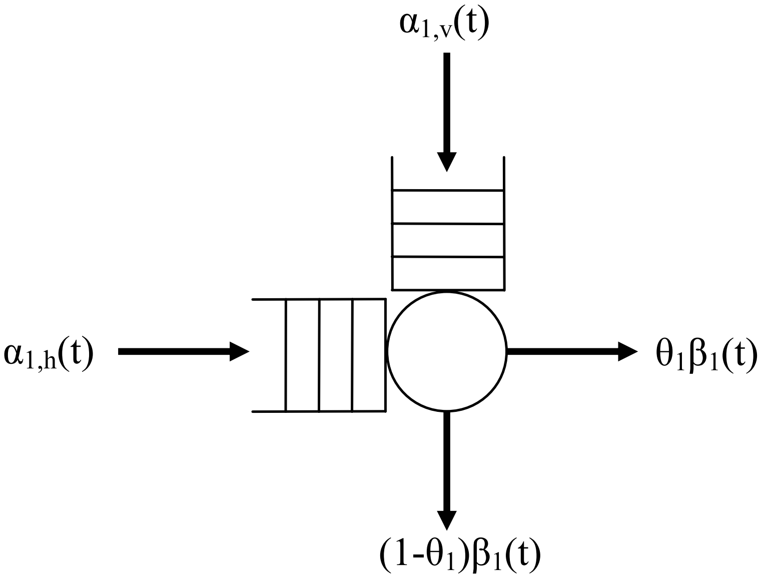


Figure 4.1: Isolated intersection: SFM Continuous model visual depiction.

is not truly representative of the traffic flow at an intersection. Using this modeling framework, we don't consider the specific scheduling policy used by the traffic light server (this will be done in the next section). Here we assume a fluid model where fluid is processed from both queues simultaneously at proportions θ_1 and $(1 - \theta_1)$. However, we decided to evaluate it just to see whether it also captures the salient features of the problem and is effective in optimization (as we will see in Section 4.1.3, this is not the case).

Sample Path Partition

For this model we partition the sample path into empty and non-empty periods. Empty periods are maximal intervals where $x_q(t;\theta_1) = 0$, while non-empty intervals indicate the intervals such that $x_q(t;\theta_1) > 0$, $q = \{(1,h),(1,v)\}$. Let

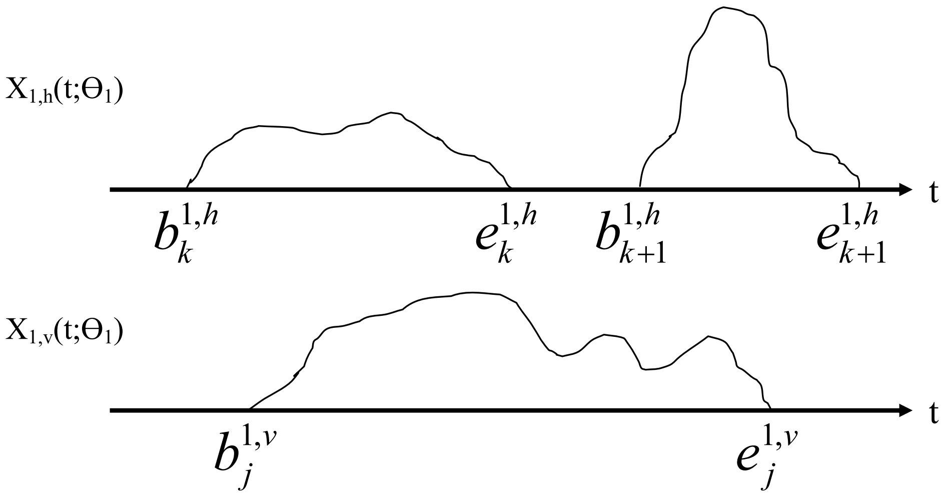


Figure 4.2: Example of sample path for SFM isolated intersection continuous model.

$\bar{E}_i^q = (b_i^q, e_i^q)$ indicate the ith non-empty period, where b_i^q indicates the beginning and e_i^q the end of the ith non-empty period at queue $q = \{(1,h),(1,v)\}$. That is, we define

$$\begin{aligned} b_i^q &= \text{epoch of the beginning of the } i\text{th non-empty period,} \\ e_i^q &= \text{epoch of the ending of the } i\text{th non-empty period.} \end{aligned}$$

Using this notation, the sample functions (4.1) can be written as

$$Q_q(\theta_1) = \frac{1}{S} \sum_{j=1}^{N_q} \int_{b_j^q}^{e_j^q} x_q(t;\theta_1)dt, \tag{4.2}$$

where N_q denotes the random number of non-empty periods in the interval $[0, S]$. Differentiating with respect to θ_1, we get

$$\frac{dQ_q(\theta_1)}{d\theta_1} = \frac{1}{S} \sum_{j=1}^{N_q} \int_{b_j^q}^{e_j^q} \frac{dx_q(t;\theta_1)}{d\theta_1} dt. \tag{4.3}$$

Because $x_q(b_j^q, \theta_1) = x_q(e_j^q, \theta_1) = 0$ for $q = \{(1,h),(1,v)\}$ and $j = 1, 2, ...$, in any interval $\bar{E}_j^{1,h} = (b_j^{1,h}, e_j^{1,h})$, the buffer content $x_{1,h}(t;\theta_1)$ is given by

$$x_{1,h}(t;\theta_1) = \int_{b_j^{1,h}}^{t} [\alpha_{1,h}(\tau) - \theta_1\beta_1(\tau)]d\tau. \tag{4.4}$$

Lemma 3. *The derivative of $x_{1,h}(t;\theta)$ with respect to θ_1 is given by*

$$\frac{dx_{1,h}(t;\theta_1)}{d\theta_1} = -\int_{b_j^{1,h}}^{t} \beta_1(\tau)d\tau. \tag{4.5}$$

Proof. The result of the differentiation is

$$\frac{dx_1(t;\theta_1)}{d\theta_1} = -\left(\alpha_1(b_j^{1,h}) - \theta\beta_1(b_j^{1,h})\right)\frac{db_j^{1,h}}{d\theta_1} - \int_{b_j^{1,h}}^{t} \beta_1(\tau)d\tau,$$

because $\frac{dt}{d\theta_1} = 0$. However, we point out that the first term vanishes. This is shown as follows. Any point $b_j^{1,h}$ is such that $\alpha_{1,h}(b_j^{1,h-}) - \theta_1\beta_1(b_j^{1,h-}) \leq 0$ and $\alpha_{1,h}(b_j^{1,h+}) - \theta_1\beta_1(b_j^{1,h+}) \geq 0$. The sign switch can occur in one of two ways, either continuously or discontinuously. If it occurs in a continuous fashion, then the above relations imply that $\alpha_{1,h}(b_j^{1,h}) - \theta_1\beta_1(b_j^{1,h}) = 0$. On the other hand, if it occurs in a discontinuous fashion, then it implies that the sign switch is due to a discontinuity either in $\alpha_{1,h}(b_j^{1,h})$ or $\beta_1(b_j^{1,h})$, which are independent of θ_1. Thus, $\frac{db_j^{1,h}}{d\theta_1} = 0$, so the term again vanishes. □

Similarly, for $q = (1, v)$, we have

$$x_{1,v}(t;\theta_1) = \int_{b_j^{1,v}}^{t} [\alpha_{1,v}(\tau) - (1-\theta)\beta_1(\tau)]d\tau, \tag{4.6}$$

and by differentiation we obtain the following result.

Lemma 4. *The derivative of $x_{1,v}(t;\theta_1)$ with respect to θ_1 is given by*

$$\frac{dx_{1,v}(t;\theta_1)}{d\theta_1} = +\int_{b_j^{1,v}}^{t} \beta_1(\tau)d\tau. \tag{4.7}$$

The proof is analogous to that of **Lemma** 3 and is therefore omitted.

Next, substituting (4.5) and (4.7) back into (4.3), we get the following result.

Theorem 4.1.1. *The sample derivatives for the workload are given by*

$$\frac{dQ_{1,h}(\theta)}{d\theta} = -\frac{1}{S}\sum_{j=1}^{N_{1,h}} \int_{b_j^{1,h}}^{e_j^{1,h}} \int_{b_j^{1,h}}^{t} \beta_1(\tau) d\tau dt, \tag{4.8}$$

$$\frac{dQ_{1,v}(\theta)}{d\theta} = \frac{1}{S}\sum_{j=1}^{N_{1,v}} \int_{b_j^{1,v}}^{e_j^{1,v}} \int_{b_j^{1,v}}^{t} \beta_1(\tau) d\tau dt. \tag{4.9}$$

Example: Next, let us consider a simple example where $\beta_1(t) = \beta_1$ (constant). In this case,

$$\begin{aligned}
\frac{dQ_{1,h}(\theta_1)}{d\theta_1} &= -\frac{1}{S}\sum_{j=1}^{N_{1,h}} \int_{b_j^{1,h}}^{e_j^{1,h}} \int_{b_j^{1,h}}^{t} \beta_1 d\tau dt \\
&= -\frac{1}{S}\sum_{j=1}^{N_{1,h}} \int_{b_j^{1,h}}^{e_j^{1,h}} \beta_1 (t - b_j^{1,h}) dt \\
&= -\frac{\beta_1}{S}\sum_{j=1}^{N_1} \frac{(e_j^{1,h})^2}{2} - b_j^{1,h} e_j^{1,h} - \left(\frac{(b_j^{1,h})^2}{2} - (b_j^{1,h})^2 \right) \\
&= -\frac{\beta_1}{2S}\sum_{j=1}^{N_{1,h}} (e_j^{1,h})^2 - 2b_j^{1,h} e_j^{1,h} + (b_j^{1,h})^2 \\
&= -\frac{\beta_1}{2S}\sum_{j=1}^{N_{1,h}} \left(e_j^{1,h} - b_j^{1,h} \right)^2 .
\end{aligned} \tag{4.10}$$

Similarly,

$$\frac{dQ_{1,v}(\theta_1)}{d\theta_1} = \frac{\beta_1}{2S}\sum_{j=1}^{N_{1,v}} \left(e_j^{1,v} - b_j^{1,v} \right)^2 . \tag{4.11}$$

Note that in order to implement these estimators, (4.10) and (4.11), we just accumulate the squares of the duration of each non-empty period.

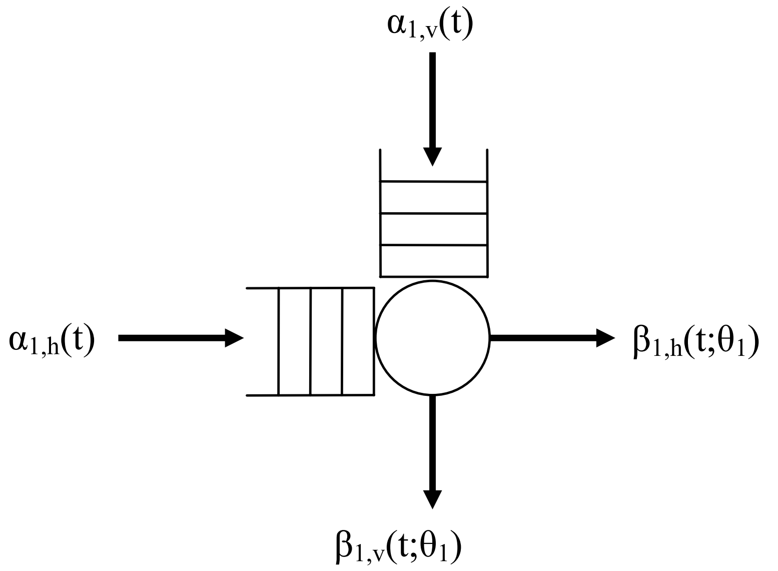


Figure 4.3: Isolated intersection: SFM Periodic model visual depiction.

4.1.2 Periodic Model

We now consider a periodic model. In this model, the server's behavior more closely resembles that of a traffic signal. Figure 4.3 shows this periodic fluid model.

Behaving similar to a traffic signal, the *entire* server capacity is allocated to the horizontal queue for a period $0 < \theta_1 < T$ and to the vertical queue for a period $0 < T - \theta_1 < T$. In this model, T indicates the length of one cycle. Figure 4.4 shows an example sample path from this model.

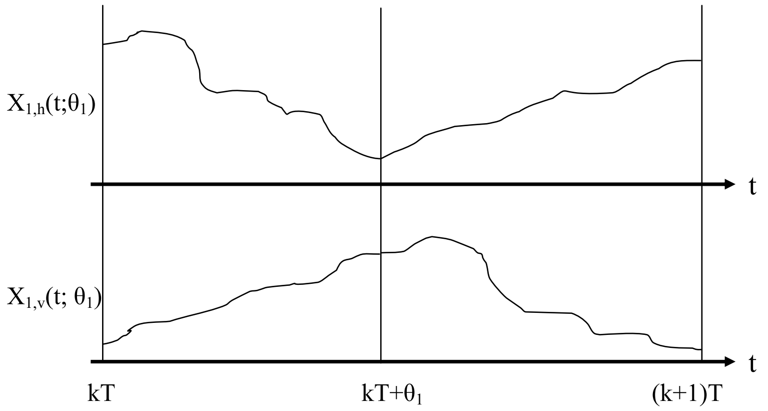


Figure 4.4: Example of sample path for SFM isolated intersection periodic model.

Sample Path Partition

In this modeling approach, the sample path is divided into intervals of length T, and the dynamics of the two queues are described as follows:

$$\frac{dx_{1,h}(t;\theta_1)}{dt} = \begin{cases} \alpha_{1,h}(t) - \beta_{1,h}(t;\theta_1) & \text{if } kT \leq t < (kT+\theta_1) \\ \alpha_{1,h}(t) & \text{if } (kT+\theta_1) \leq t < (k+1)T, \end{cases} \tag{4.12}$$

$$\frac{dx_{1,v}(t;\theta_1)}{dt} = \begin{cases} \alpha_{1,v}(t) & \text{if } kT \leq t < (kT+\theta_1) \\ \alpha_{1,v}(t) - \beta_{1,v}(t;\theta) & \text{if } (kT+\theta_1) \leq t < (k+1)T, \end{cases} \tag{4.13}$$

where $k = 1, 2, \cdots$. In addition, the service rates are defined as

$$\beta_{1,h}(t;\theta_1) \;=\; \begin{cases} \rho_{1,h}(t) & \text{if } kT \leq t < kT+\theta_1 \text{ and } x_{1,h}(t;\theta_1) > 0 \\ \alpha_{1,h}(t) & \text{if } kT \leq t < kT+\theta_1 \text{ and } x_{1,h}(t;\theta_1) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

$$\beta_{1,v}(t;\theta_1) \;=\; \begin{cases} \rho_{1,v}(t) & \text{if } kT+\theta_1 \leq t < (k+1)T \text{ and } x_{1,v}(t;\theta_1) > 0 \\ \alpha_{1,v}(t) & \text{if } kT+\theta_1 \leq t < (k+1)T \text{ and } x_{1,v}(t;\theta_1) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

where $\rho_{1,h}(t)$ and $\rho_{1,v}(t)$ are the maximum possible outflows from queues $Q_{1,h}$ and $Q_{1,v}$, respectively. The sample functions of (4.1) can be written as

$$Q_q(\theta_1) = \frac{1}{S}\sum_{k=0}^{K-1}\int_{kT}^{(k+1)T} x_q(t;\theta_1)dt, \tag{4.16}$$

where $K = \frac{S}{T}$ is the number of periods included in the interval $[0, S]$ and the index $q \in \{(1,h),(1,v)\}$. Differentiating with respect to θ_1 we get

$$\frac{dQ_q(\theta_1)}{d\theta} = \frac{1}{S}\sum_{k=1}^{K}\int_{kT}^{(k+1)T} \frac{dx_q(t;\theta_1)}{d\theta_1}dt. \tag{4.17}$$

We first evaluate a single term from the summation for $q = (1,h)$, i.e., we begin by examining

$$\int_{kT}^{(k+1)T} \frac{dx_{1,h}(t;\theta_1)}{d\theta_1}dt.$$

Given the queue dynamics stated in equation (4.12), we determine the queue content based upon which interval that t falls in:

Case A: $(kT \leq t < kT + \theta)$ This interval is further divided into two subcases depending on the observation of an empty period. Here we also make an assumption that during a period of 'green' light, if the queue becomes empty, then it will not become non-empty before the next 'red' light.

A1: When the queue does *not* become empty during this period, we have

$$x_{1,h}(t;\theta_1) = x_{1,h}(kT;\theta) + \int_{kT}^{t}[\alpha_{1,h}(\tau) - \rho_{1,h}(\tau)]d\tau. \tag{4.18}$$

A2: When the queue does become empty during the period, we have

$$x_{1,h}(t;\theta_1) = \begin{cases} x_{1,h}(kT;\theta_1) & \\ + \int_{kT}^{t} [\alpha_{1,h}(\tau) - \rho_{1,h}(\tau)] d\tau & \text{if } kT \leq t < e_k^{1,h} \\ 0 & \text{if } e_k^{1,h} \leq t < kT + \theta_1. \end{cases} \tag{4.19}$$

Case B: $(kT + \theta_1 \leq t < (k+1)T)$ When t falls in this interval, we have

$$x_{1,h}(t;\theta_1) = x_{1,h}(kT + \theta_1;\theta_1) + \int_{kT+\theta_1}^{t} \alpha_{1,h}(\tau) d\tau. \tag{4.20}$$

Summarizing, we get

$$x_{1,h}(t;\theta_1) = \begin{cases} x_{1,h}(kT;\theta_1) + \int_{kT}^{t} [\alpha_{1,h}(\tau) - \rho_{1,h}(\tau)] d\tau & \text{if } kT \leq t < e_k^{1,h} \\ 0 & \text{if } e_k^{1,h} \leq t < kT + \theta_1 \\ x_{1,h}(kT + \theta_1;\theta_1) + \int_{kT+\theta_1}^{t} \alpha_{1,h}(\tau) d\tau & \text{if } kT + \theta \leq t < (k+1)T \end{cases} \tag{4.21}$$

where $e_k^{1,h}$ indicates the time when the buffer empties during the kth period. If no such event occurs, then we set $e_k^{1,h} = (kT + \theta_1)$; thus, the second case does not occur. Next, differentiating equation (4.21) we get

$$\frac{dx_{1,h}(t;\theta_1)}{d\theta_1} = \begin{cases} \frac{dx_{1,h}(kT;\theta_1)}{d\theta_1} & \text{if } kT \leq t < e_{j_k}^{1} \\ 0 & \text{if } e_k^{1,h} \leq t < kT + \theta_1 \\ \frac{dx_{1,h}(kT+\theta_1;\theta_1)}{d\theta_1} - \alpha_{1,h}(kT + \theta_1) & \text{if } kT + \theta_1 \leq t < (k+1)T \end{cases}$$

In other words, the derivative $\frac{dx_{1,h}(t;\theta_1)}{d\theta_1}$ is a piecewise constant function. This function can be implemented iteratively using a single accumulator. That is, the value of the derivative at any given time point is a function of the previous values. As a result, the derivative $\frac{dQ_{1,h}(\theta_1)}{d\theta_1}$ can also be evaluated via accumulators. We can rewrite it as a product of the derivative and the corresponding intervals.

$$\begin{aligned}\frac{dQ_{1,h}(\theta_1)}{d\theta_1} &= \frac{1}{S}\sum_{k=0}^{K-1}\left[(e_k^{1,h} - kT)\frac{dx_{1,h}(kT;\theta_1)}{d\theta_1}\right. \\ &\left. +(T-\theta_1)\left(\frac{dx_{1,h}(kT+\theta_1;\theta_1)}{d\theta_1} - \alpha_{1,h}(kT+\theta_1)\right)\right] \qquad (4.22)\end{aligned}$$

where as mentioned earlier, $e_k^{1,h}$ is the time that buffer $x_{1,h}$ empties during the interval $[kT, kT+\theta_1)$, and if no such event occurs, then $e_k^{1,h} = kT + \theta_1$.

Using a similar analysis for $\frac{dQ_{1,v}(\theta_1)}{d\theta_1}$, we get

$$\begin{aligned}\frac{dQ_{1,v}(\theta_1)}{d\theta_1} &= \frac{1}{S}\sum_{k=0}^{K-1}\left[(e_k^{1,v} - kT)\frac{dx_{1,v}(kT;\theta_1)}{d\theta_1}\right. \\ &\left. +(T-\theta_1)\left(\frac{dx_{1,v}(kT+\theta_1;\theta_1)}{d\theta_1} - \alpha_{1,v}(kT+\theta_1)\right)\right]. \qquad (4.23)\end{aligned}$$

4.1.3 Numerical Results

In this section, we simulate the SFM estimators derived in this chapter. These estimators, along with the SPA estimators, from Chapter 2, are compared to FD estimates.

The Continuous Model was implemented on the underlying stochastic model. The lengths of the busy periods were recorded as the simulation progressed. The lengths of the busy periods of the stochastic model were larger on average than those of the SFM, because if a busy period started during a green light cycle and did not end before the light changed red, then that busy period had no chance of ending until after the following red-light cycle. The perpetuation of busy periods by red-light cycles causes problems for the Continuous Model.

The Periodic model was also implemented on the underlying stochastic model. The assumption that the models stays empty once it empties causes the estimator to be low, because in general the queue can become non-empty again during the same green-light cycle. We made a slight modification to the estimator to allow for arrivals to the queue during a green light phase while the system is empty. Instead of resetting $\frac{dx(kT;\theta)}{d\theta_1}$ whenever the system empties, we only reset when the system is empty at the epoch of the light change. This make intuitive sense, because the perturbation only can propagate through the cycle if the system is nonempty. We refer to this as the modified periodic model (periodic mod).

We implemented estimators for the horizontal queue using four different sets of parameters. In all cases, the interarrival and service time distributions were exponentially distributed, the number of cycles (K) was 10,000, and the number of replications was also 10,000.

The first case (C1) had parameter values:

- respective mean interarrival and service times of 4.5 and 2.0;
- mean green length of 30.0 and mean total cycle length of 60.0.

The second case (C2) had parameter values:

- respective mean interarrival and service times of 5.0 and 1.5;
- mean green length of 35.0 and mean total cycle length of 110.0.

The third case (C3) had parameter values:

- respective mean interarrival and service times of 3.5 and 0.5;

estimator	$dE[\bar{L}_{1,h}]/d\theta$ (std. error)
SPA (RH)	-2.465 (0.001)
SPA (LH)	-2.465 (0.001)
FD (.05)	-2.475 (0.024)
SFM (Simple)	-147.82 (0.154)
SFM (Periodic)	-1.713 (0.002)
SFM (Periodic mod)	-2.188 (0.006)

Table 4.1: Gradient estimate simulation results for all estimators for "C1" (standard errors in parentheses).

estimator	$dE[\bar{L}_1]/d\theta$ (std. error)
SPA (RH)	-8.3904 (0.0061)
SPA (LH)	-8.3835 (0.0066)
FD (.05)	-8.2115 (0.0356)
SFM (Simple)	-915.4631 (1.4496)
SFM (Periodic)	-0.1683 (0.0000)
SFM (Periodic mod)	-0.2412 (0.0000)

Table 4.2: Gradient estimate simulation results for all estimators for "C2" (standard errors in parentheses).

estimator	$dE[\bar{L}_1]/d\theta$ (std. error)
SPA (RH)	-0.1717 (0.0000)
SPA (LH)	-0.1716 (0.0001)
FD (.05)	-0.1716 (0.0003)
SFM (Simple)	-11.1422 (0.0004)
SFM (Periodic)	-0.1674 (0.0000)
SFM (Periodic mod)	-0.1960 (0.0000)

Table 4.3: Gradient estimate simulation results for all estimators for "C3" (standard errors in parentheses).

- mean green length of 20.0 and mean total cycle length of 40.0.

The fourth case (C4) had parameter values:

- respective mean interarrival and service times of 10.5 and 5.0;
- mean green length of 20.0 and mean total cycle length of 40.0.

Estimators were simulated for all 4 cases; however, the optimization was carried out only for C1, C2 and C3.

The performance measure of interest, average number in system, has a mean settle down point of 2750 cycles. That is, steady-state is reached for each of the cases on average after 2750 cycles are simulated. This value was determined by observing multiple simulation runs. The average number in system was checked at the start of each cycle, when the average number in system was within and stayed

estimator	$dE[\bar{L}_1]/d\theta$ (std. error)
SPA (RH)	-20.8959 (0.0424)
SPA (LH)	-20.8848 (0.0417)
FD (.05)	-20.2334 (0.1146)
SFM (Simple)	-775.8161 (4.0491)
SFM (Periodic)	-19.0584 (0.0979)
SFM (Periodic mod)	-19.7437 (0.0989)

Table 4.4: Gradient estimate simulation results for all estimators for "C4" (standard errors in parentheses).

within 10% of the steady-state average number in system, that cycle was recorded as the settle down point.

Gradient Estimation

The simulation results for C1, C2, C3 and C4 are shown in Tables 4.1, 4.2, 4.3 and 4.4, respectively. A comparison of the results shows that the Continuous model estimator is large in magnitude compared to the other estimates. We take the FD estimate as the true value in Table 4.1.3 to compare the percent error of each estimator. The left and right hand SPA estimates are extremely accurate. Both the Periodic and the modified Periodic models provide fair estimates of the gradient. The standard error for the four aforementioned estimators is always less than that for the FD estimates. Even though the estimates do not match the FD estimates exactly, they still look rather promising for use in system optimization.

We also note that because the standard error is smaller for the SPA and SFM gradient estimators; therefore, their success in SA is enhanced. That is, when a certain level of accuracy is desired, fewer simulations are needed for the SPA and SFM gradient estimators, making them more computationally efficient.

Optimization

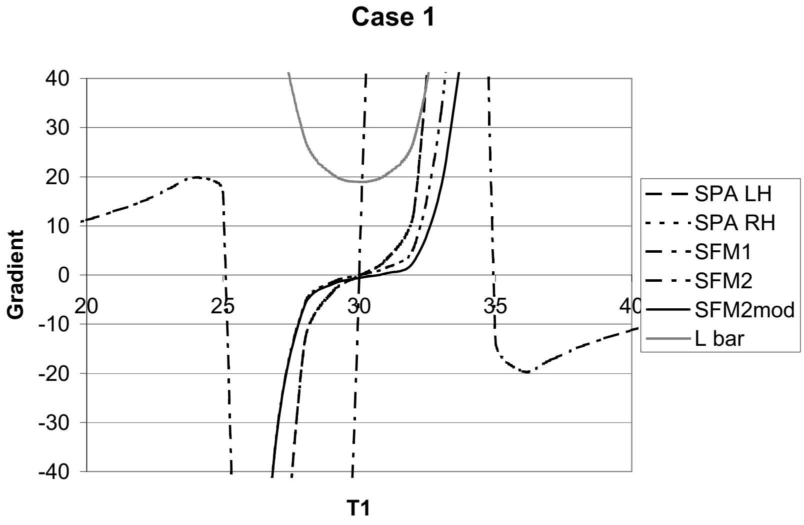


Figure 4.5: Five gradient estimation methods (SPA LH, SPA RH, SFM-IPA continuous, SFM-IPA periodic, SFM-IPA modified periodic) plotted vs $\bar{L}_{1,h}$.

We noticed that the SFM gradient estimated were not very accurate; however, from Figure 4.5, we can see that the estimates have a very important quality: they are close to 0 at the minimum of the function. This is a good indicator that the

SFM estimates can be used for optimization via a gradient descent algorithm such as SA. All six gradient estimation techniques cross 0 at the minimum of $\bar{L}_{1,h}$. Figure 4.5 shows this for C1; the same property of the estimators was exhibited for C2 and C3 as well.

Because SA is an iterative algorithm, not only are we concerned with reaching the optimum, but we would like to stay near the optimum for subsequent updates. Therefore, we ran simulations and counted the number of times the average number in system was within $p\%$ (for $p = 10,5,1$) of the minimum average number in system based on the current $T_{1,h}$ from the SA algorithm. We label the three ranges as

- 10%-range : within 10% of the optimal $\bar{L}_{1,h}$
- 5%-range : within 5% of the optimal $\bar{L}_{1,h}$
- 1%-range : within 1% of the optimal $\bar{L}_{1,h}$.

All six gradient estimation techniques were implemented in the SA algorithm for cases C1, C2 and C3. Tables 4.6, 4.7 and 4.8 show the percentage of time each estimation fell within the optimum range. We note that SFM1 performs very well for C3, relatively well for C1, and poorly for C2. We see that SFM2, SFM2mod, SPA RH, SPA LH and FD all have similar values. These five methods do a good job in the optimization of the isolated intersection traffic system. Also we have that these five methods do an equivalent job in the optimization process, which is noteworthy, because the fluid models gradient estimates were not accurate.

estimator	case 1	case 2	case 3	case 4	avg. abs. err.
SPA (RH)	-0.72%	2.18%	0.09%	3.27%	1.57%
SPA (LH)	-0.70%	2.10%	0.03%	3.22%	1.51%
FD (.05)	0.00%	0.00%	0.00%	0.00%	0.00%
SFM (Simp)	6170.05%	11048.61%	6393.84%	3734.33%	6836.71%
SFM (Per)	-30.86%	-97.95%	-2.45%	-5.81%	34.27%
SFM (Per mod)	-11.87%	-97.06%	14.21%	-2.42%	31.39%

Table 4.5: Percent error in gradient estimators when FD is used as the true gradient value.

estimator	10%-range	5%-range	1%-range
SPA (RH)	77.0	46.4	10.9
SPA (LH)	77.8	45.6	10.9
FD (.05)	76.5	45.0	10.1
SFM (Simple)	48.9	30.2	5.7
SFM (Periodic)	79.3	47.9	10.9
SFM (Periodic mod)	73.6	45.0	10.0

Table 4.6: Percentage of iterations that the SA algorithm is with p% of the optimum value for “C1”.

estimator	10%-range	5%-range	1%-range
SPA (RH)	92.9	89.9	60.1
SPA (LH)	92.7	90.2	61.0
FD (.05)	94.0	91.8	57.0
SFM (Simple)	3.0	2.6	1.2
SFM (Periodic)	95.9	94.3	66.3
SFM (Periodic mod)	95.1	91.3	10.9

Table 4.7: Percentage of iterations that the SA algorithm is with p% of the optimum value for "C2".

estimator	10%-range	5%-range	1%-range
SPA (RH)	43.0	21.8	4.6
SPA (LH)	43.1	22.4	4.2
FD (.05)	41.2	19.9	3.9
SFM (Simple)	43.0	21.7	4.7
SFM (Periodic)	48.3	26.4	4.7
SFM (Periodic mod)	42.9	21.7	4.6

Table 4.8: Percentage of iterations that the SA algorithm is with p% of the optimum value for "C3".

4.1.4 Conclusions

SFMs are promising for the purpose of *control and optimization*, rather than *performance analysis*. In this chapter we have shown that even if the exact gradient cannot be obtained by such "lower-resolution" models, one can still obtain near-optimal points that exhibit robustness with respect to certain aspects of the model on which they are based. For this reason, we believe that these SFM IPA gradient estimates can used to optimize the traffic model. Next we will consider the application of the SFM IPA estimator method to a network of two signalized intersections.

4.2 Network of Two Signalized Intersections

In this section, we analyze a network of two signalized intersections, identical to the system described in chapter 2.

4.2.1 Continuous Model

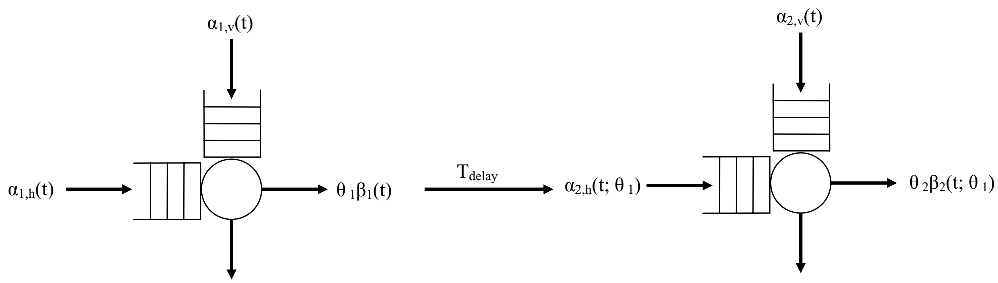


Figure 4.6: Network of two signalized intersections: Continuous model visual depiction.

The queueing model representing the SFM continuous model of a network of two signalized intersections is shown in Figure 4.6. Each server is always on for each street. The processing capacity for each Light-i (server) is split between the horizontal and vertical streets with proportions θ_i and $1-\theta_i$, respectively. The main addition in this model from the isolated intersection model is that the inflow to the horizontal queue of Light-2 comes from the outflow of the horizontal queue at Light-1. We also note there is no transient queue nor an offset between the intersections. In lieu of a transient queue, we enforce a delay on the fluid traveling from Light-1 to Light-2. No offset is applicable, because both lights are always green for each street.

Sample Path Partition

We start by determining which gradient estimators are required in this setting. In this setting, there are four queues $(x_{1,h}, x_{1,v}, x_{2,h}, x_{2,v})$ and two parameters (θ_1, θ_2). Queues $x_{1,h}$ and $x_{1,v}$ are affected by θ_1, but are independent of θ_2 because θ_2 only has an effect on the operation of Light-2. Queue $x_{2,h}$ has an exogenous inflow; thus, nothing in the system can affect that process. On the other hand, the outflow is dictated by the allowed processing capacity, i.e., the outflow is a function of θ_2. Queue $x_{2,h}$ is unique in this setting, because it has an endogenous inflow. This endogenous inflow comes from the outflow of queue $x_{1,h}$, and queue $x_{1,h}$'s outflow is a function of θ_1. Therefore, we have that the inflow of queue $x_{2,h}$ is a function of θ_1. And of course, queue $x_{2,h}$'s outflow is a function of θ_2. Hence, queue $x_{2,h}$ is a function of θ_1 and θ_2, that is $x_{2,h}(t) = F(t, \theta_1, \theta_2)$. It follows that for the network of two signalized intersections problem modeled using the continuous model SFM process, we need estimators for

$$\frac{dQ_{1,h}(\theta_1)}{d\theta_1}, \frac{dQ_{1,v}(\theta_1)}{d\theta_1}, \frac{dQ_{2,h}(\theta_1,\theta_2)}{d\theta_1}, \frac{dQ_{2,h}(\theta_1,\theta_2)}{d\theta_2}, \text{ and } \frac{dQ_{2,v}(\theta_2)}{d\theta_2}.$$

All estimators can be derived using the same analysis outlined in Section 4.1.1 except for $\frac{dQ_{2,h}(\theta_1,\theta_2)}{d\theta_1}$, which requires new analysis.

We start by again partitioning the sample path into empty and non-empty periods. For queue $x_{2,h}$, we have

$$Q_{2,h}(\theta_1, \theta_2) = \frac{1}{S} \sum_{j=1}^{N_{2,h}} \int_{b_j^{2,h}}^{e_j^{2,h}} x_{2,h}(t; \theta_1; \theta_2) dt, \tag{4.24}$$

where $N_{2,h}$ denotes the random number of non-empty periods in the interval $[0, S]$. Differentiating with respect to θ_1, we get

$$\frac{dQ_{2,h}(\theta_1, \theta_2)}{d\theta_1} = \frac{1}{S} \sum_{j=1}^{N_{2,h}} \int_{b_j^{2,h}}^{e_j^{2,h}} \frac{dx_{2,h}(t; \theta_1, \theta_2)}{d\theta_1} dt, \tag{4.25}$$

since $x_{2,h}(b_j^{2,h}, \theta_1) = x_{2,h}(e_j^{2,h}, \theta_1) = 0$ for $j = 1, 2, ...$ In any interval $\bar{E}_j^{2,h} = (b_j^{2,h}, e_j^{2,h})$, the buffer content $x_{2,h}(t; \theta_1, \theta_2)$ is given by

$$\begin{aligned} x_{2,h}(t; \theta_1, \theta_2) &= \int_{b_j^{2,h}}^{t} [\alpha_{2,h}(\tau) - \theta_2 \beta_2(\tau)] d\tau \\ &= \int_{b_j^{2,h}}^{t} [\theta_1 \beta_{1,h}(\tau - T_{delay}) - \theta_2 \beta_{2,h}(\tau)] d\tau \\ &= \theta_1 \int_{b_j^{2,h}}^{t} \beta_{1,h}(\tau - T_{delay}) d\tau - \theta_2 \int_{b_j^{2,h}}^{t} \beta_{2,h}(\tau) d\tau, \end{aligned} \tag{4.26}$$

where T_{delay} represents the amount of time the fluid is delayed when traveling from Light-1 down to Light-2. In order to get the desired estimator, we differentiate, and the derivative of $x_{2,h}(t;\theta_1;\theta_2)$ with respect to θ_1 is given by

$$\begin{aligned}
\frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_1} &= -\int_{b_j^{2,h}}^{t} \beta_{1,h}(\tau - T_{delay})d\tau \\
&\quad + \theta_1[-\beta_{1,h}(b_j^{2,h} - T_{delay})\frac{db_j^{2,h}}{d\theta_1} + \theta_2[\beta_{2,h}(b_j^{2,h})]\frac{db_j^{2,h}}{d\theta_1} \\
&= -\int_{b_j^{2,h}}^{t} \beta_{1,h}(\tau - T_{delay})d\tau + \frac{db_j^{2,h}}{d\theta_1}[\theta_2\beta_{2,h}(b_j^{2,h}) - \theta_1\beta_{1,h}(b_j^{2,h})] \\
&= -\int_{b_j^{2,h}}^{t} \beta_{1,h}(\tau - T_{delay})d\tau + \frac{db_j^{2,h}}{d\theta_1}[\theta_2\beta_{2,h}(b_j^{2,h}) - \alpha_{2,h}(b_j^{2,h})] \\
&= -\int_{b_j^{2,h}}^{t} \beta_{1,h}(\tau - T_{delay})d\tau. \qquad (4.27)
\end{aligned}$$

Now we define

$$\begin{aligned}
E_{1,h} &= \bigcup_{i=1}^{K} \bar{E}_i^{1,h}, \\
E_{1,h}^c &= \{t | t \notin E_{1,h}\},
\end{aligned}$$

and

$$\begin{aligned}
E_{1,h}(t) &= \text{Event that queue } x_{1,h}(t) = 0, \\
E_{1,h}(t_1, t_2) &= \text{Event that queue } x_{1,h} = 0 \text{ at sometime } t_1 < t < t_2,
\end{aligned}$$

and then the service rate for the horizontal queue of Light-1 can be expressed as

$$\beta_{1,h}(t;\theta_1) = \begin{cases} \rho_{1,h}(t) & \text{if } t \in E^{1,h} \\ \alpha_{1,h}(t) & \text{if } t \in E_c^{1,h}. \end{cases} \qquad (4.28)$$

Next, substituting (4.28) back in (4.27) we get the following result

$$\frac{dx_{2,h}(t;\theta_1,\theta_2)}{d\theta_1} = \int_{b_j^{2,h}}^{t} \alpha_{1,h}(\tau)\mathbf{1}\{E_{1,h(\tau)}\}d\tau + \int_{b_j^{2,h}}^{t} \rho_{1,h}(\tau)(1 - \mathbf{1}\{E_{1,h}(\tau)\})d\tau. \qquad (4.29)$$

Theorem 4.2.1. *The sample derivative for the workload is given by*

$$\begin{aligned}\frac{dQ_{2,h}(\theta_1,\theta_2)}{d\theta_1} &= \frac{1}{S}\sum_{j=1}^{N_{2,h}}\int_{b_j^{2,h}}^{e_j^{2,h}}\int_{b_j^{2,h}}^{t}[\alpha_{1,h}(\tau)\mathbf{1}\{E_{1,h}(\tau)\} \\ &\qquad + \rho_{1,h}(\tau)(1-\mathbf{1}\{E_{1,h}(\tau)\})]\,d\tau. \qquad (4.30)\end{aligned}$$

Example: Next, let us consider a simple example where $\alpha_{1,h}(t) = \alpha_{1,h}$ and $\rho_{1,h}(t) = \rho_{1,h}$ (constant). In this case,

$$\begin{aligned}\frac{dQ_{2,h}(\theta_1,\theta_2)}{d\theta_1} &= \frac{1}{S}\sum_{j=1}^{N_{2,h}}\int_{b_j^{2,h}}^{e_j^{2,h}}\left\{\int_{b_j^{2,h}}^{t}\alpha_{1,h}\mathbf{1}\{E_{1,h}(\tau)\}d\tau + \int_{b_j^{2,h}}^{t}\rho_{1,h}\mathbf{1}\{E_{1,h}(\tau)\}d\tau\right\} \\ &= \frac{1}{S}\sum_{j=1}^{N_{2,h}}\int_{b_j^{2,h}}^{e_j^{2,h}}\Big\{(t-b_j^{2,h})\alpha_{1,h}P[E_{1,h}(b_j^{2,h},t)] \\ &\qquad + \rho_{1,h}(1-P[E_{1,h}(b_j^{2,h},t)])dt\Big\} \\ &= \frac{1}{S}\sum_{j=1}^{N_{2,h}}\Big\{\alpha_{1,h}P[E_{1,h}(b_j^{2,h},e_j^{2,h})] \\ &\qquad +\rho_{1,h}P[E_{1,h}(b_j^{2,h},e_j^{2,h})]((e_j^{2,h})^2 - 2b_j^{2,h}e_j^{2,h} + (b_j^{2,h}))^2\Big\} \\ &= \frac{(\alpha_{1,h}P[E_{1,h}(0,S)] + \rho_{1,h}P[E_{1,h}(0,S)]}{S}\sum_{j=1}^{N_{2,h}}(e_j^{2,h} - b_j^{2,h})^2. \qquad (4.31)\end{aligned}$$

We can see that this estimator is similar to those derived earlier, the main difference is that now we have a weighted flow rate. Not only do we need the duration of the non-empty period, but we also need the proportion of time that both $x_{1,h}(t)$ and $x_{2,h}(t)$ are empty at the same time.

4.2.2 Periodic Model

Figure 4.7 shows a fluid model where server i's capacity is allocated to the horizontal queue for a period $0 < \theta_i < T_i$ and to the vertical queue for a period $0 < T - \theta_i < T_i$, for $i = 1, 2$. In this model, T_i indicates the period of one cycle

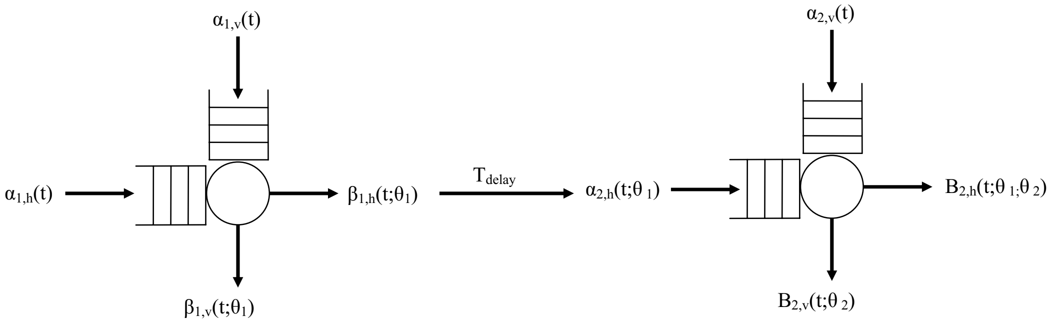

$\alpha_{1,v}(t)$
$\alpha_{1,h}(t)$
$\beta_{1,h}(t;\theta_1)$
$\beta_{1,v}(t;\theta_1)$
T_{delay}
$\alpha_{2,h}(t;\theta_1)$
$\alpha_{2,v}(t)$
$B_{2,h}(t;\theta_1;\theta_2)$
$B_{2,v}(t;\theta_2)$

Figure 4.7: Network of two signalized intersections: Periodic model visual depiction.

from 'green' to 'red' for intersection i. Figure 4.8 shows a typical sample path due to this model.

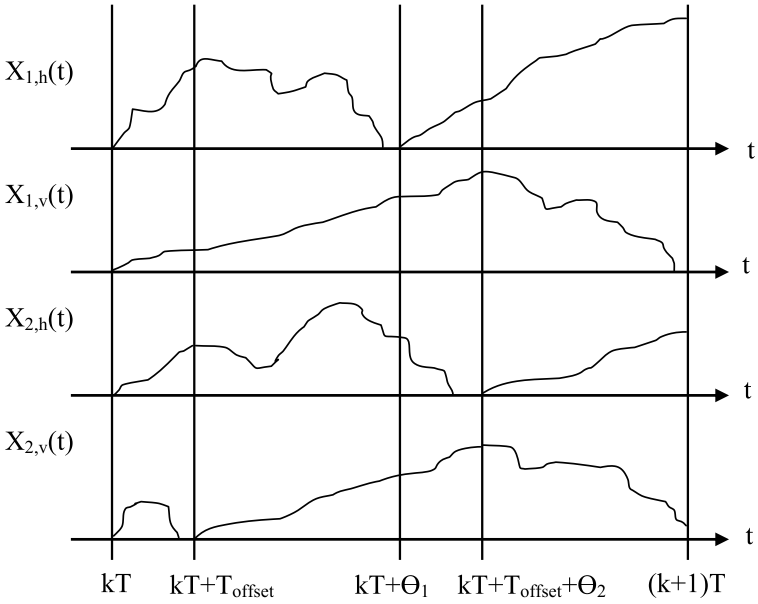


Figure 4.8: Example of sample path for a network of two signalized intersections periodic model.

Sample Path Partition

In this modeling approach the sample path is divided into intervals of length $T_1 = T_2 = T$. The dynamics and estimators for queues $Q_{1,h}$, $Q_{1,v}$ and $Q_{2,v}$ are identical to those described in the previous section. Note that here, the cycle of length T, is further divided into four subintervals. These subintervals are defined as

$$\begin{aligned} S_1^k &= [kT, kT + T_{offset}), \\ S_2^k &= [kT + T_{offset}, kT + \theta_1), \\ S_3^k &= [kT + \theta_1, kT + T_{offset} + \theta_2), \\ S_4^k &= [kT + T_{offset} + \theta_2, (k+1)T). \end{aligned}$$

The dynamics of the horizontal queue related to intersection 2 are as follows

$$\frac{dx_{2,h}(t;\theta_1;\theta_2)}{dt} = \begin{cases} \alpha_{2,h}(t;\theta_1) & \text{if } T \in S_1^k \\ \alpha_{2,h}(t;\theta_1) - \beta_{2,h}(t;\theta_1;\theta_2) & \text{if } T \in \{S_2^k \bigcup S_3^k\} \\ \alpha_{2,h}(t;\theta_1) & \text{if } T \in S_4^k, \end{cases} \tag{4.32}$$

$k = 1, 2, \cdots$. In addition, the service rates are defined as

$$\beta_{2,h}(t;\theta_1;\theta_2) = \begin{cases} 0 & \text{if } t \in S_1^k \\ \alpha_{2,h}(t,\theta_1) & \text{if } t \in \{S_2^k \bigcup S_3^k\} \text{ and } x_{2,h}(t) = 0 \\ \rho_{2,h}(t,\theta_1) & \text{if } t \in \{S_2^k \bigcup S_3^k\} \text{ and } x_{2,h}(t) > 0 \\ 0 & \text{if } t \in S_4^k. \end{cases} \tag{4.33}$$

In this setting we also have variations in the inflow rate for queue $Q_{2,h}$. Namely,

$$\alpha_{2,h}(t;\theta_1) = \beta_{1,h}(t - T_{delay};\theta_1), \tag{4.34}$$

where T_{delay} is the time it takes for the fluid to move from intersection 1 to intersection 2. This is analogous to the transient queue from the DES model. The sample functions of (4.1) can be written as

$$Q_{2,h}(\theta) = \frac{1}{S}\sum_{k=0}^{K-1}\int_{kT}^{(k+1)T} x_{2,h}(t;\theta_1;\theta_2)dt, \tag{4.35}$$

where K is the number of periods included in the interval $[0, S]$. We first start by differentiating with respect to θ_1 and get

$$\frac{dQ_{2,h}(\theta_1;\theta_2)}{d\theta_1} = \frac{1}{S}\sum_{k=0}^{K-1}\int_{kT}^{(k+1)T} \frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_1}dt. \tag{4.36}$$

For simplicity, let us first evaluate a single term from the summation, i.e., we start by analyzing

$$\int_{kT}^{(k+1)T} \frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_1}dt.$$

Given the queue dynamics of (4.32), we determine the queue content based upon which interval t belongs to:

Case A: ($t \in S_1^k$) During this interval, Light-2 is ‘red‘ for the horizontal queue of intersection 2; thus, we get

$$x_{2,h}(t;\theta_1;\theta_2) = x_{2,h}(kT;\theta_1;\theta_2) + \int_{kT}^{t} \alpha_{2,h}(\tau;\theta_1)d\tau. \tag{4.37}$$

Case B: ($t \in \{S_2^k \bigcup S_3^k\}$) During this interval, the light is ‘green‘ for the horizontal queue of intersection 2. This interval is further divided into two more subcases depending on the observation of an empty period. Here we also make an assumption that during a period of ‘green’ light, if the queue becomes empty, then it will not become non-empty before the next ‘red’ light.

B1: During this interval, the queue does *not* become empty, so we get

$$x_{2,h}(t;\theta_1;\theta_2) = x_{2,h}(kT;\theta_1;\theta_2) + \int_{kT}^{t} \alpha_{2,h}(\tau;\theta_1) - \beta_{2,h}(\tau;\theta_1;\theta_2)d\tau. \quad (4.38)$$

B2: During this interval, the queue does becomes empty, so we get

$$x_{2,h}(t;\theta_1;\theta_2) = \begin{cases} x_{2,h}(kT;\theta_1;\theta_2) & \\ + \int_{kT}^{t}[\alpha_{2,h}(\tau;\theta_1) & \\ -\rho_{2,h}(\tau;\theta_1;\theta_2)]d\tau & \text{if } kT + T_{offset} \leq t < e_k^{2,h} \\ 0 & \text{if } e_k^{2,h} \leq t \\ & < kT + T_{offset} + \theta_2. \end{cases} \quad (4.39)$$

Case C: $(t \in S_4^k)$ During this interval, Light-2 is 'red' for the horizontal queue of intersection 2.

$$x_{2,h}(\tau;\theta_1;\theta_2) = x_{2,h}(kT + T_{offset} + \theta_2;\theta_1;\theta_2) + \int_{kT+T_{offset}+\theta_2}^{t} \alpha_{2,h}(\tau;\theta_1)d\tau \quad (4.40)$$

By summarizing, using equation (4.34) and the following set of intervals,

$$\begin{aligned} I_1^k &= [kT, kT + T_{offset}), \\ I_2^k &= [kT + T_{offset}, e_k^{2,h}), \\ I_3^k &= [e_k^{2,h}, kT + T_{offset} + \theta_2), \\ I_4^k &= [kT + T_{offset} + \theta_2, (k+1)T), \end{aligned}$$

we get

$$x_{2,h}(t;\theta_1;\theta_2) = \begin{cases} x_{2,h}(kT;\theta_1;\theta_2) + \int_{kT}^{t} \beta_{1,h}(\tau - T_{delay})d\tau & \text{if } t \in I_1^k \\ x_{2,h}(kT + T_{offset};\theta_1;\theta_2) & \\ \quad + \int_{kT+T_{offset}}^{t} [\beta_{1,h}(\tau - T_{delay}) - \beta_{2,h}(\tau;\theta_1;\theta_2)]d\tau & \text{if } t \in I_2^k \\ 0 & \text{if } t \in I_3^k \\ x_{2,h}(kT + T_{offset} + \theta_2;\theta_1;\theta_2) & \\ \quad + \int_{kT+T_{offset}+\theta_2}^{t} \beta_{1,h}(\tau - T_{delay})d\tau & \text{if } t \in I_4^k, \end{cases} \tag{4.41}$$

where $e_k^{2,h}$ indicates the epoch of the buffer emptying during the kth period. If no such event occurs, then we set $e_k^1 = (kT + T_{offset} + \theta_2)$; thus, the second case does not occur. Next, differentiating (4.41) w.r.t. θ_2 we get

$$\frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_2} = \begin{cases} \frac{dx_{2,h}(kT;\theta_1;\theta_2)}{d\theta_2} & \text{if } t \in I_1^k \\ \frac{dx_{2,h}(kT+T_{offset};\theta_1;\theta_2)}{d\theta_2} & \text{if } t \in I_2^k \\ 0 & \text{if } t \in I_3^k \\ \frac{dx_{2,h}(kT+T_{offset}+\theta_2;\theta_1;\theta_2)}{d\theta_2} - \beta_{1,h}(kT + T_{offset} + \theta_2 - T_{delay}) & \text{if } t \in I_4^k. \end{cases} \tag{4.42}$$

In other words, the derivative $\frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_2}$ is a piecewise constant function. This function can be implemented iteratively using a single accumulator. The derivative $\frac{dQ_{2,h}(\theta_1;\theta_2)}{d\theta_2}$ is just the derivative times the corresponding intervals. Therefore we get

$$\begin{aligned} \frac{dQ_{2,h}(\theta_1;\theta_2)}{d\theta_2} &= \frac{1}{S}\sum_{k=0}^{K-1}\left\{|I_1^k|\frac{dx_{2,h}(kT;\theta_1;\theta_2)}{d\theta_2} + |I_2^k|\frac{dx_{2,h}(kT+T_{offset};\theta_1;\theta_2)}{d\theta_2}\right. \\ &\quad + |I_4^k|\left[\frac{dx_{2,h}(kT;\theta_1;\theta_2)}{d\theta_2}\right. \\ &\quad \left.\left. -\beta_{1,h}(kT + T_{offset} + \theta_2 - T_{delay})\right]\right\}. \end{aligned} \tag{4.43}$$

Next, differentiating (4.41) w.r.t. θ_1 we get

$$\frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_1} = \begin{cases} \frac{dx_{2,h}(kT;\theta_1;\theta_2)}{d\theta_1} & \text{if } t \in I_1^k \\ \frac{dx_{2,h}(kT+T_{offset};\theta_1;\theta_2)}{d\theta_1} + 2\alpha_{1,h}(kT+T_{offset}) & \text{if } t \in I_2^k \\ 0 & \text{if } t \in I_3^k \\ \frac{dx_{2,h}(kT+T_{offset}+\theta_2;\theta_1;\theta_2)}{d\theta_1} + \alpha_{1,h}(kT+T_{offset}) & \text{if } t \in I_4^k. \end{cases} \tag{4.44}$$

In other words, the derivative $\frac{dx_{2,h}(t;\theta_1;\theta_2)}{d\theta_1}$ is a piecewise constant function. This function can be implemented iteratively using a single accumulator. The derivative $\frac{dQ_{2,h}(\theta_1;\theta_2)}{d\theta_1}$ is just the derivative times the corresponding intervals. Therefore we get

$$\begin{aligned} \frac{dQ_{2,h}(\theta_1;\theta_2)}{d\theta_1} &= \frac{1}{S}\sum_{k=0}^{K-1}\left\{|I_1^k|\frac{dx_{2,h}(kT;\theta_1;\theta_2)}{d\theta_1} + |I_2^k|\frac{dx_{2,h}(kT+T_{offset};\theta_1;\theta_2)}{d\theta_1}\right. \\ &\qquad \left. +|I_4^k|\frac{dx_{2,h}(kT;\theta_1;\theta_2)}{d\theta_1}\right\}. \end{aligned} \tag{4.45}$$

Now we derive the gradient estimators w.r.t. T_{offset}. As before, both queue $Q_{2,h}$ and $Q_{2,v}$ will have a non-zero gradient w.r.t. T_{offset}. For clarity, in the remaining equations, we will suppress all queues and flow rate dependence of θ_i, for $i = 1, 2, 3$. So, by differentiating (4.41) w.r.t. $\theta_3 = T_{offset}$ we get

$$\begin{aligned} \frac{dQ_{2,h}}{d\theta_3} &= \frac{1}{S}\sum_{k=0}^{K-1}\left\{|I_1^k|\frac{dx_{2,h}(kT)}{d\theta_3}\right. \\ &\qquad +|I_2^k|\left(\frac{dx_{2,h}(kT+T_{offset})}{d\theta_3} - [\beta_{1,h}(\tau - T_{delay}) - \beta_{2,h}(\tau)]\right) \\ &\qquad \left. +|I_4^k|\left(\frac{dx_{2,h}(kT)}{d\theta_3} - \beta_{1,h}(kT+T_{offset}+\theta_2-T_{delay})\right)\right\}. \end{aligned} \tag{4.46}$$

and by reassigning the intervals as such

$$\begin{aligned}
I_1^k &= [kT, e_k^{(2,v)-1st}),\\
I_2^k &= [kT + T_{offset}, kT + T_{offset} + \theta_2),\\
I_3^k &= [kT + T_{offset} + \theta_2, e_k^{(2,v)-2nd}),
\end{aligned}$$

$$\begin{aligned}
\frac{dQ_{2,v}}{d\theta_3} &= \frac{1}{S}\sum_{k=0}^{K-1}\left\{|I_1^k|\frac{dx_{2,v}(kT)}{d\theta_3}\right.\\
&+|I_2^k|\left(\frac{dx_{2,h}(kT+T_{offset})}{d\theta_3} - [\alpha_{2,v}(\tau - T_{delay})]\right)\\
&+|I_3^k|\left(\frac{dx_{2,h}(kT)}{d\theta_3} - [\alpha_{2,v}(kT + T_{offset} + \theta_2 - T_{delay})\right.\\
&\qquad \left.\left. - \rho_{2,v}(kT + T_{offset} + \theta_2 - T_{delay})]\right)\right\}. \qquad (4.47)
\end{aligned}$$

4.2.3 Optimization

The results of the SFM-IPA (and the other methods from Chapter 3) driven SA for cases SC, NC, and AC can be seen in Figures 4.9, 4.10, and 4.11, respectively. (Note that the case descriptions can be seen in Table 3.2) We observe that optimization was achieved with the use the SFM-IPA estimator for all three cases. The SFM-IPA periodic model gradient estimators reached optimal settings.

In fact, the SFM-IPA estimators appear to reach the optimal parameter settings quicker than any of the other gradient estimation methods. However, it is of importance to note that the SFM-IPA does not completely settle down. The high variance in the SFM-IPA estimators results in changes in the parameter (represented by lots of bumps in the plots), even after the all other methods have converged.

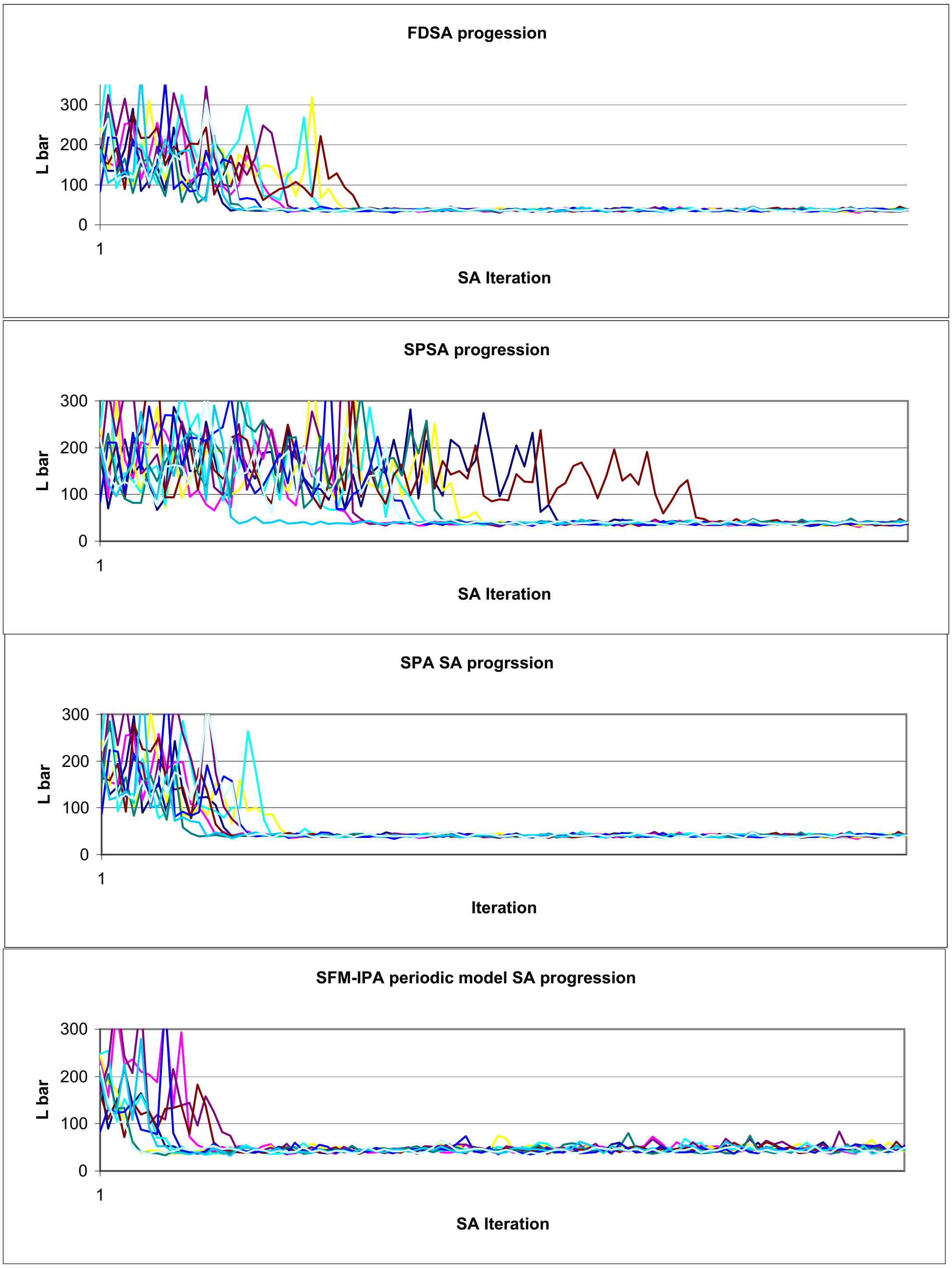


Figure 4.9: Convergence to minimum for 10 replications of the SA algorithm for case “SC” for four different gradient estimation methods: FDSA, SPSA, SPA SA and SFM-IPA SA.

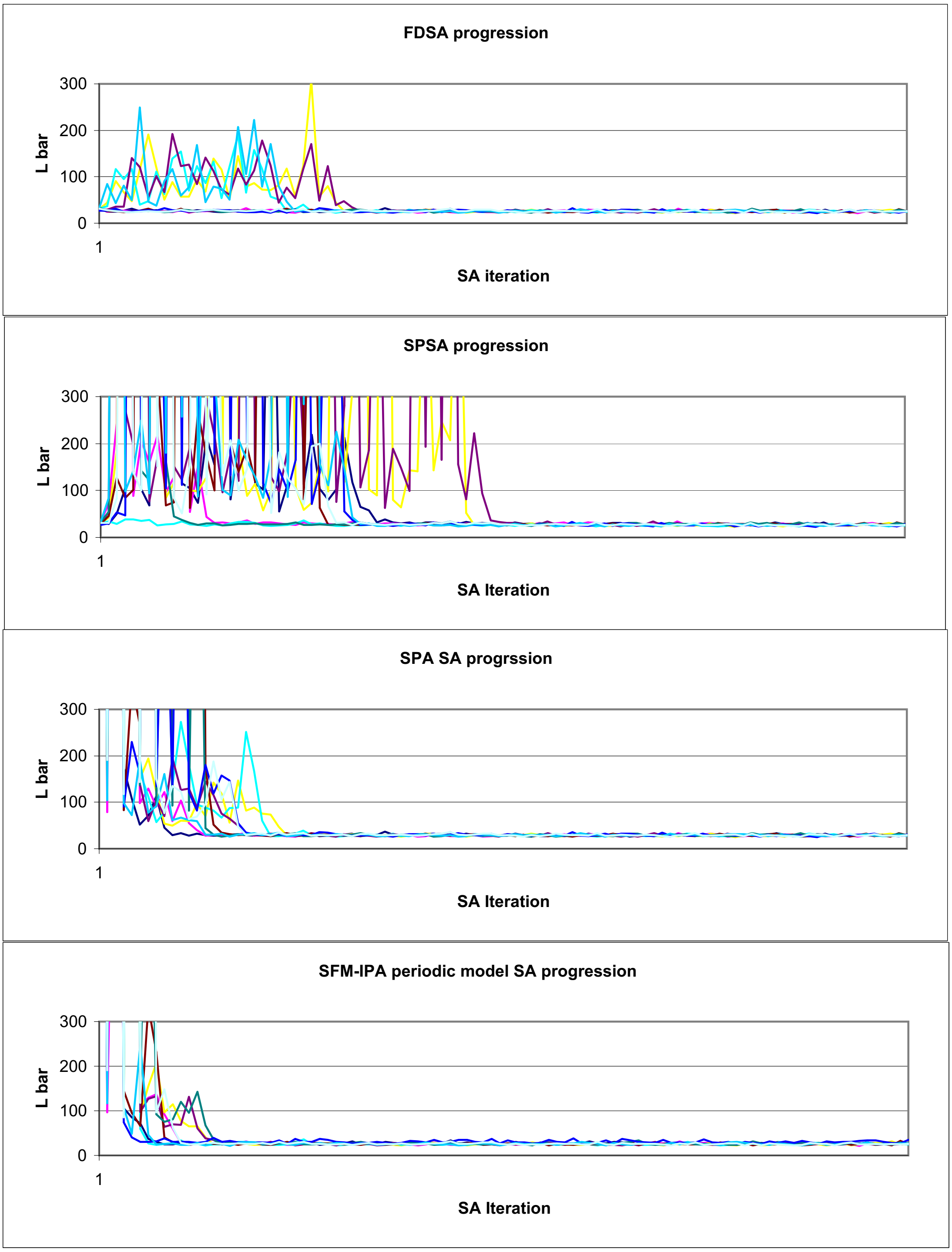


Figure 4.10: Convergence to minimum for 10 replications of the SA algorithm for case "NC" for four different gradient estimation methods: FDSA, SPSA, SPA SA and SFM-IPA SA.

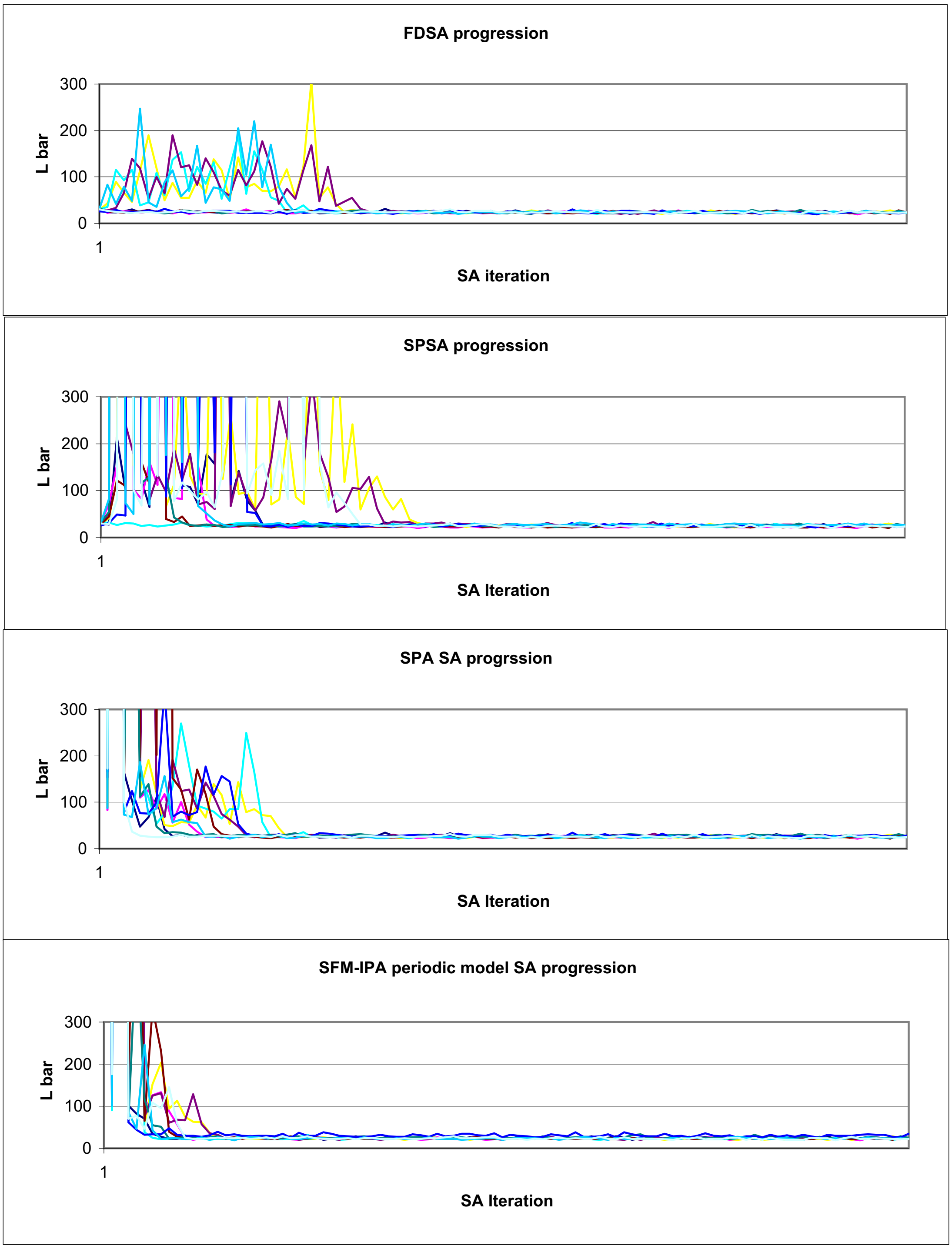


Figure 4.11: Convergence to minimum for 10 replications of the SA algorithm for case "AC" for four different gradient estimation methods: FDSA, SPSA, SPA SA and SFM-IPA SA.

4.2.4 Conclusions

The SFM-IPA estimators for the network of two signalized intersection are similar to those for the isolated intersection. The intervals on which the estimators are constant are more complex in this setting, because there are now more points at which the system dynamics change; however, their final form still allows them to be calculated via simulation using a single accumulator. Again we have shown that the SFM-IPA gradient estimators can be used for system optimization.

Chapter 5

Conclusions and Future Work

We have considered an important issue: the evaluation and optimization of traffic systems. In Chapter 2, we consider an isolated intersection traffic system modeled using DES. We successfully optimized this traffic system. This was the first successful attempt to apply direct stochastic gradient estimation techniques to a traffic flow optimization setting. The resulting estimators demonstrated superior computational performance over FD estimators, and in addition can be used on line with real-time traffic updating systems, because unlike FD estimators, they do not require altering the parameter values. In Chapter 3, we expanded the problem to a network of two signalized intersections. Again, the derived SPA estimators performed well when compared to FD estimates. In Chapter 4, we re-examined the isolated intersection and network of two signalized intersection problems. There we modeled the traffic systems using SFM and derived IPA estimators. These estimators were motivated with the goal of system optimization in mind. We showed that SFMs are promising for the purpose of control and optimization rather than for performance analysis. In Chapter 3 we showed that even if the exact gradient cannot be obtained by such "lower-resolution" models, one can still obtain near-optimal points that exhibit robustness with respect to certain aspects of the model they are based on.

We suggest several avenues for future research. Throughout this dissertation, we used traffic models that only included one-way streets and did not allow turning. So, one avenue to explore is to determine how the estimators change and perform when used with traffic models in which two-way traffic and turning vehicles are present. There are additional changes to the model that would create interesting problems. For example, how do the estimators perform when the arrival rates, $\lambda_q(t)$ are non-stationary, which is more representative of real-world traffic systems. Application of these estimators to more detailed traffic models is a good area of future research.

Further work for estimators derived in this dissertation include the derivation of estimators when the traffic system contains more than two intersections. We were able to see that new analysis was necessary when we expanded from one to two lights; therefore, we would expect even more analysis when additional intersections are added to the traffic system.

The use of SA could also be studied more. Future work could include convergence proofs for SA algorithm. Also a more in depth analysis of SA as it applies to traffic optimization would be useful to avoid local minimums and facilitate global optimization.

Additional work could be done to make the simulation optimization algorithms more practical. The model could be altered to include such thing as pedestrians and additional classes of vehicles.

The SFM-IPA approach could benefit from further exploration. Issues such as

- When will it work well?
- How detailed of a model is needed?

could be addressed.

Appendix A
Portion of C Program for Isolated Intersection Gradient Estimator

Here we provide an excerpt from one of the many programs developed for this dissertation. This program is the header file which returns all gradient estimators, both SPA and SFM-IPA, for the isolated intersection traffic setting.

```
void N_main(double ms1, double ms2, double mi1,double mi2, double
mg1, double mg2, int num_cycles, int nr, unsigned long
a1[6],unsigned long a2[6],unsigned long a3[6],unsigned long a4[6],
int N_bar_flag, double N_bar[3], int SPA_N1_LH_flag, double
SPA_N1_LH[2], int SPA_N2_LH_flag, double SPA_N2_LH[4], int
SPA_N1_RH_flag, double SPA_N1_RH[3],int SPA_N2_RH_flag, double
SPA_N2_RH[3], int SFMe1_flag,     double SFMe1[3], int SFMe2_flag,
double SFMe2[3], int SFMe2mod_flag,  double SFMe2mod[3], int
need_empty_times_flag, double tte_sys_1[Q_LIMIT+1], double
tte_sys_2[Q_LIMIT+1])
{
    int lcv;
    /* Set up seeds for random numbers */
    A1 = RngStream_CreateStream ("A1");
    A2 = RngStream_CreateStream ("A2");
    S1 = RngStream_CreateStream ("S1");
    S2 = RngStream_CreateStream ("S2");

    /*Initialze seeds*/
    RngStream_SetSeed(A1,a1);
    RngStream_SetSeed(A2,a2);
    RngStream_SetSeed(S1,a3);
    RngStream_SetSeed(S2,a4);

    /* Set up system parameters */
    mean_service_green_1        = ms1;
    mean_service_green_2        = ms2;

    mean_interarrival_1         = mi1;
    mean_interarrival_2         = mi2;
    mi1_init                    = mi1;
    mi2_init                    = mi2;
    mean_green_length_1         = mg1;
    mean_green_length_2         = mg2;
```

```
N                             = num_cycles;
num_reps                      = nr;

/* Define a stopping condition */
G1              = mean_green_length_1;
G2              = mean_green_length_2;
T               = G1 + G2;
stop_sim_time   = N*T;

/*Specify the number of events for the timing function*/
num_events = 6;

// Only need maxes for SPA
// if we got maxes just assign then, otherwise we
   need to get them
if (need_empty_times_flag == 1)
{
    make_max();
    // Need to put arrival means back on for both
    mean_interarrival_1         = mi1;
    mean_interarrival_2         = mi2;
    for (lcv = 1; lcv <= Q_LIMIT; lcv++)
    {
        tte_sys_1[lcv] = time_to_empty_sys_1[lcv];
        tte_sys_2[lcv] = time_to_empty_sys_2[lcv];
    }
    //put seeds back to original
    /*Initialze seeds*/
    RngStream_SetSeed(A1,a1);
    RngStream_SetSeed(A2,a2);
    RngStream_SetSeed(S1,a3);
    RngStream_SetSeed(S2,a4);
}
else
{
    for (lcv = 1; lcv <= Q_LIMIT; lcv++)
    {
        time_to_empty_sys_1[lcv] = tte_sys_1[lcv];
        time_to_empty_sys_2[lcv] = tte_sys_2[lcv];
    }
}
main_body();
if (N_bar_flag == 1)
{
```

```
/* Sets and returns value for N_bar, N1_bar and N2_bar */
        N_out(N_bar);
    }
    if (SPA_N1_LH_flag == 1)
    {
        /* Sets and returns value for SPA N1 LH */
        SPA_N1_LH_out(SPA_N1_LH);
    }
    if (SPA_N2_LH_flag == 1)
    {
        /* Sets and returns value for SPA N2 LH */
        SPA_N2_LH_out(SPA_N2_LH);
    }
    if (SPA_N1_RH_flag == 1)
    {
        /* Sets and returns value for SPA N1 RH */
        SPA_N1_RH_out(SPA_N1_RH);
    }
    if (SPA_N2_RH_flag == 1)
    {
        /* Sets and returns value for SPA N2 RH */
        SPA_N2_RH_out(SPA_N2_RH);
    }
    if (SFMe1_flag == 1)
    {
        /* Sets and returns value for SFM1e */
        SFMe1_out(SFMe1);
    }
    if (SFMe2_flag == 1)
    {
        /* Sets and returns value for SFM2e */
        SFMe2_out(SFMe2);
    }
    if (SFMe2mod_flag == 1)
    {
        /* Sets and returns value for SFMe2mod */
        SFMe2mod_out(SFMe2mod);
    }
    //Free memory from streams
    RngStream_DeleteStream(&A1); RngStream_DeleteStream(&A2);
    RngStream_DeleteStream(&S1); RngStream_DeleteStream(&S2);
}
//*************************************************************
void main_body()  /* Main function. */ {
    /* Initialize the simulation. */
    initialize();
/*Run the simulation while more delays are still needed*/
    while (sim_time < stop_sim_time)
    {
        /* Determine the next event. */
        timing();
        /* Update time-average statistical accumulators. */
        if (sim_time > stop_sim_time)
        {
            sim_time = stop_sim_time;
            time_complete = 1;
        }
    update_time_avg_stats();
    if (time_complete == 0)
            {
            /* Invoke the appropriate event function. */
            switch (next_event_type)
            {
                    case 1:
                        arrive_1();
                        break;
                    case 2:
                        arrive_2();
                        break;
                    case 3:
                        depart_1();
                        break;
                    case 4:
                        depart_2();
                        break;
                    case 5:
                        light_1_turn_green();
                        break;
                    case 6:
                        light_2_turn_green();
                        break;
            }
        }
    }
}
//**********************************************************
/* Initialization function. */ void initialize(void) {
    int lcv;
    int lcvr,lcvc;
    numserv2 = 0;
```

```
/* Initialize the simulation clock. */
sim_time = 0.0;
/***************************************************/
// SPA N1 LH part
valid_SPA_entry_to_service = 0;
for (lcv = 1 ; lcv <= N ; lcv++)
{
    NETS_1[lcv] = 0;
    num_departures[lcv] = 0;
}
for (lcvr = 1 ; lcvr <= N ; lcvr++)
    for (lcvc = 1; lcvc < max_per_cycle ; lcvc++)
    {
        NA_1[lcvr][lcvc] = 0;
        NIS_1[lcvr][lcvc] = 0;
        ETS_1[lcvr][lcvc] = 0.0;
    }
car_ID_1 = 0;
/***************************************************/
// SPA N2 LH part
num_valid_IPA_departures_1 = 0;
for (lcv = 0; lcv<= N; lcv++)
{
    E_1[lcv] = 0.0;
    S_1[lcv] = 0.0;
    G_1[lcv] = 0.0;
}
index_SPA_N2_LH = 0;
/***************************************************/
//SPA N1 RH
last_departure = FLT_MAX;
last_entry_to_service = FLT_MAX;
for (lcv = 0; lcv<= N; lcv++)
    E_2[lcv] = 0.0;
index_SPA_RH = 0;
/***************************************************/
// SPA N2 RH
num_valid_IPA_departures_2 = 0;
valid_SPA_entry_to_service = 0;
for (lcv = 1 ; lcv <= N ; lcv++)
{
    NETS_2[lcv] = 0;
    NETS_first_busy_period[lcv]=0;
}
for (lcvr = 1 ; lcvr <= N ; lcvr++)
```

```
    for (lcvc = 1; lcvc < max_per_cycle ; lcvc++)
        NA_2[lcvr][lcvc] = 0;
car_ID_2 = 0;
/***************************************************/
// SFMe2 and SFMe2mod
if (num_in_q_1 == 0)
    current_period_empty_flag_1     = 1;
else
    current_period_empty_flag_1     = 0;
if (num_in_q_2 == 0)
    current_period_empty_flag_2     = 1;
else
    current_period_empty_flag_2     = 0;
for(lcv=0;lcv<=N;lcv++)
{
    e_k_1[lcv] = -1;
    e_k_2[lcv] = -1;
    dx_kT_1[lcv] = 0;
    dx_kT_2[lcv] = 0;
    dx_kTtheta_1[lcv] = 0;
    dx_kTtheta_2[lcv] = 0;
    dx_kTtheta_1mod[lcv] = 0;
    dx_kT_2mod[lcv] = 0;
}
/***************************************************/
// for finding the max
max_in_sys_1 = max_in_sys_2 = 0;
done = 0;
for (lcv = 1 ; lcv <= Q_LIMIT; lcv++)
{
    emptied[lcv] = 0;
    times[lcv]   = 0.0;
}
/***************************************************/
// SFMe1
current_sys_empty_flag_1        = 1;
busy_period_1                   = 0;
current_sys_empty_flag_2        = 1;
busy_period_2                   = 0;
/***************************************************/
/* Initialize the state variables. */
/* Starting condition is both lights red, but light one
   will turn green at time 0 */
server_status_1 = IDLE;
server_status_2 = IDLE;
```

```
    light_1     = RED;
    light_2     = RED;
    num_in_q_1      = start_Q_1;
    num_in_q_2      = start_Q_2;
    time_last_event = 0.0;
    time_complete   = 0;
    index = 0;
    /* Initialize the statistical counters. */
    num_custs_delayed_1     = 0;
    num_custs_delayed_2     = 0;
    num_custs_completed_1   = 0;
    num_custs_completed_2   = 0;
    total_of_delays_1       = 0.0;
    total_of_delays_2       = 0.0;
    area_num_in_sys_1       = 0.0;
    area_num_in_sys_2       = 0.0;
    area_server_status_1    = 0.0;
    area_server_status_2    = 0.0;
    /* Initialize event list */
    time_next_event[1] = get_dist_arrival_1();
    time_next_event[2] = get_dist_arrival_2();
    time_next_event[3] = FLT_MAX;
        /* Red light means no departures allowed */
    time_next_event[4] = FLT_MAX;
        /* Red light means no departures allowed */
    time_next_event[5] = 0.0;
        /* Light 1 turns green at the beginning of simulation */
    time_next_event[6] = FLT_MAX;
}
//************************************************************
/* Timing function. */ void timing(void) {
    int   i;
    double min_time_next_event = FLT_MAX - 1;
    next_event_type = 0;
    /* Determine the event type of the next event to occur. */
    for (i = 1; i <= num_events; ++i)
        if (time_next_event[i] < min_time_next_event)
            {
                min_time_next_event = time_next_event[i];
                next_event_type     = i;
            }
    /* Check to see whether the event list is empty. */
    if (next_event_type == 0)
        {
        /* The event list is empty, so stop the simulation. */
        printf("\nEvent list empty at time %f", sim_time);
        exit(1);
        }
    /* The event list is not empty,
       so advance the simulation clock. */
    sim_time = min_time_next_event;
}
//************************************************************
/* Arrival event function at queue 1. */ void arrive_1(void) {
    int lcv;
    //********************************************************
    // SPA N1 LH
    if ( (light_1 == GREEN) )
        for (lcv = 1; lcv <= NETS_1[index] ; lcv++)
            NA_1[index][lcv] += 1;
    if ( (light_1 == GREEN) && (server_status_1 == IDLE))
    {
        car_ID_1 += 1;
        NETS_1[index] += 1;
        ETS_1[index][car_ID_1] = sim_time;
        NIS_1[index][car_ID_1]= num_in_q_1 + server_status_1 + 1;
    }
    //***********************************************************
    // SPA N1 RH
    if ( (light_1 == GREEN) && (server_status_1 == IDLE))
        last_entry_to_service = sim_time;
    //***********************************************************
    // SFM_1 N1
    if (current_sys_empty_flag_1 == 1)  //if sys is empty
    {
        current_sys_empty_flag_1    = 0;
        busy_period_1           += 1;
        busy_period_start_time_1[busy_period_1] = sim_time;
    }
    //***********************************************************
    /* Schedule next arrival. */
    time_next_event[1] = sim_time + get_dist_arrival_1();
    /* check for time stopping condition */
    if (sim_time >= stop_sim_time)
        time_complete = 1;
    /* Check to see whether stopping condition is met */
    if (time_complete == 0)
    {
        if ((server_status_1 == BUSY )||(light_1 == RED))
        {
            /* There is still room in the queue,
               so store the time of arrival of the
            arriving customer at the (new) end of time_arrival_1. */
            num_in_q_1++;
            /* Check to see whether an overflow condition exists. */
            if (num_in_q_1 > Q_LIMIT)
            {
            /* The queue has overflowed, so stop the simulation. */
                printf( "\nOverflow of the array time_arrival_1 at");
                printf( " time %f", sim_time);
                exit(2);
            }
        }
        else
        /* server is idle, there is no delay in the queue */
        {
            /* Increment the number of customers delayed,
                and make server BUSY. */
            ++num_custs_delayed_1;
            /* Schedule departure of this arrival */
            time_next_event[3] = sim_time+get_dist_service_time_1();
            server_status_1 = BUSY;
        }
    }
}
//***************************************************************
void arrive_2(void)  /* Arrival event function at queue 2. */ {
    int lcv;
    //***********************************************************
    // SPA N2 RH
    if ( (light_2 == GREEN) )
        for (lcv = 1; lcv <= NETS_2[index_SPA_RH] ; lcv++)
            NA_2[index_SPA_RH][lcv] += 1;
    if ( (light_2 == GREEN) && (server_status_2 == IDLE))
    {
        car_ID_2 += 1;
        NETS_2[index_SPA_RH] += 1;
        ETS_2[index_SPA_RH][car_ID_2] = sim_time;
        NIS_2[index_SPA_RH][car_ID_2]=num_in_q_2+server_status_2+1;
    }
    //**************************************************************
    // SFM_1 N2
    if (current_sys_empty_flag_2 == 1)  //if sys is empty
    {
        current_sys_empty_flag_2    = 0;
        busy_period_2           += 1;
        busy_period_start_time_2[busy_period_2] = sim_time;
    }
    //**************************************************************
    /* Schedule next arrival */
    time_next_event[2] = sim_time + get_dist_arrival_2();
    /* check for time stopping condition */
    if (sim_time >= stop_sim_time)
        time_complete = 1;
    /* Check to see whether stopping condition is met */
    if (time_complete == 0)
    {
        if ((server_status_2 == BUSY )||(light_2 == RED))
        {
            /* There is still room in the queue,
               so store the time of arrival of the
            arriving customer at the (new) end of time_arrival_2. */
                    num_in_q_2++;
            /* Check to see whether an overflow condition exists. */
                if (num_in_q_2 > Q_LIMIT)
            {
                /*The queue has overflowed, so stop the simulation*/
                printf( "\nOverflow of the array time_arrival_2 at");
                printf( " time %f", sim_time);
                exit(2);
            }
        }
        else
        /* server is idle, there is no delay in the queue */
        {
            /* Increment the number of customers delayed,
               and make server BUSY. */
            ++num_custs_delayed_2;
            /* Schedule departure of this customer */
            time_next_event[4]=sim_time+get_dist_service_time_2();
            server_status_2 = BUSY;
        }
    }
}
```

BIBLIOGRAPHY

[1] R. B. Allsop, "SIGSET: A Computer Program for Calculating Traffic Capcity of Signal-controlled Road Junctions," *Traffic Engineering & Control*, vol.12, pp.58-60, 1971.

[2] R. B. Allsop, "SIGCAP: A Computer Program for Assessing the Traffic Capcity of Signal-controlled Road Junctions," *Traffic Engineering & Control*, vol.17, pp.338-341, 1976.

[3] J. Banks and J.S. Carson, *Discete-Event System Simulation*, Prentice-Hall, Inc., Englewood, NJ; 1984.

[4] F. Boillot, J.M. Blosseville, J.B.Lesort, V. Motyka, M. Papageorgiou, and S. Sellam, "Optimal Signal Control of Urban Traffic Networks", *6th IEE Intern. Conference on Road Traffic Monitoring and Control*, pp.75-79, 1992.

[5] C. G. Cassandra, Y. Wardi, B. Melamed, G. Sun, and C.G. Panayiotou, "Perturbation Analysis for On-line Control and Optimization of Stochastic Fluid Models," *IEEE Transactions on Automatic Control*, vol.AC-47, No.8, pp.1234-1248, 2002.

[6] X.-R. Cao, "Perturbation Analysis of Discrete Event Systems: Concepts, Algorithms, and Applications," *European Journal of Operational Research*, vol.91, pp.1-13, 1996.

[7] N.A. Chaudhary, A. Pinnoi and C. Messer, “Proposed Enhancements to MAXBAND-86 Program,” U.S. Dept. Transp., Washington D.C., Transp. Res. Record 1324, 1991.

[8] L. De la Bretegue and R. Jezeguel, “Adaptive Control at an Isolated Intersection–a Comparitive Study of Some Algorithms,” *Traffic Engineering & Control*, vol.20, pp.361-363, 1979.

[9] B. De Schutter and B. Be Moor, “Optimal Traffic Light Control for a Single Intersection,” *European Journal on Control*, vol.4, pp.260-276, 1998.

[10] J.-L. Farges, J-J. Henry and J. Tufal, “The PRODYN Real-time Traffic Algorithm,” *4th IFAC Symposium on Transportation Systems*, pp.301-312, 1983.

[11] Federal Highway Administration Office of Operations,“Signal Timings Process Final Report,” 2005.

[12] M.E. Fouladvand, Z. Sadjjadi, and M.R. Shaebani, “Optimized Traffic Flow at a Single Intersection: Traffic Responsive Signalization,” *Journal of Physics A: Mathematical and General*, vol.37, pp.561-576, 2004.

[13] M.C. Fu, “Optimization via Simulation: A Review,” *Annals of Operations Research*, vol.53, pp.199-248, 1994.

[14] M.C. Fu and J.Q. Hu, *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*, Kluwer Academic Publishers, Boston, MA; 1997.

[15] M.C. Fu and J.Q. Hu, "Efficient Design and Sensitivity of Control Charts Using Monte Carlo Simulation," *Management Science*, Vol.45, No.3, pp.395-413, 1999.

[16] M.C. Fu and S.D. Hill, "Optimization of Discrete Event Systems via Simultaneous Perturbation Stochastic Approximation," *IEEE Transactions*, Vol.29, 1997, pp.233-243.

[17] N. Garber and L.A. Hoel, *Highway and Traffic Engineering (THIRD EDITION)* PWS Pub. Co., Boston, MA; 1997.

[18] N.H. Gartner, "OPAC: A Demand-responsive Strategy for Traffic Signal Control," *Traffic Engineering & Control*, No.906, pp.75-84, 1983.

[19] N.H. Gartner, "Road Traffic Control: Progression Methods," In *Concise Encyclopedia of Traffic & Transportation Systems*, M. Papageorgiou, Editor, Pergamon Press, Oxford, UK, 391-396, 1991.

[20] N.H. Gartner, S.F. Assmann, F. Lasaga, and D.L. Hom, "A Multiband Approach to Arterial Traffic Signal Optimization," *Transportation Research B*, vol.25, pp.55-74, 1991.

[21] R.S. Ghaman, L. Zhang, G. McHale, and C. Stallard, "The Role of Traffic Simulation in Traffic Sinal Control System Development," *Proceedings of The IEEE Intelligent Transportation Systems Conference*, pp.872-877, 2003.

[22] W.B. Gong and Y.-C. Ho, "Smoothed Perturbation Analysis of Discrete-Event Dynamical Systems," *IEEE Transactions on Automatic Control*, vol.32, pp.858-867, 1987.

[23] L. Head, F.W. Ciarallo, D. Lucas, and V. Kaduwela, “A Perturbation Analysis Approach to Traffic Signal Optimization,” INFORMS National Meeting, Washington, D.C., May 5-8, 1996.

[24] Y.-C. Ho and X.-R. Cao, “Perturbation Analysis and Optimization of Queueing Networks,” *J. Optim. Theory Appl.*, Vol.40, pp.700-714, 1983.

[25] Y.-C. Ho, “Perturbation Analysis: Concepts and Algorithms,” *Proceedings of the Winter Simulation Conference*, pp.231-240, 1992.

[26] K.N. Hewage and J.Y. Ruwanpura, “Optimization of Traffic Signal Light Timing Using Simulation,” *Proceedings of the Winter Simulation Conference*, pp.1428-1433, 2004.

[27] G. Improta and G.E. Cantarella, “Control Systems Design for an Individual Signalised Junction,” *Transportation Research B*, Vol.18, pp.147-167, 1984.

[28] A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis (3rd Edition)*, McGraw-Hill, Boston, MA; 2000.

[29] D.C. Lee, “Applying Perturbation Analysis to Traffic Shapping,” *Computer Communications*, Vol.24, pp.798-810, 2002.

[30] R.V. Lindgren and S. Tantiyanugulchai, “Microscopic Simulation of Traffic at a Surburban Interchange,” *Institute of Transprtation Engineers 2003 Annual Meeting*, 2003.

[31] J.D.C. Little, “The Synchronisation of Traffic Signals by Mixed -integer-linear-programming,” *Operations Research*, Vol.14, pp.568-594, 1966.

[32] J.D.C. Little, M.D. Kelson, and N.H. Gartner, “MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks,” *Transportation Research Record*, No.795, pp.40-46, 1981.

[33] F.L. Mannering, W.P. Kilareski, and S.S. Washburn, *Principles of Highway Engineering and Traffic Analysis*, John Wiley, Hoboken, NJ; 2005.

[34] A. J. Miller, “A Computer Control System for Traffic Networks,” *Proc. 2nd Int. Symp. Traffic Theory*, pp.200-220, 1963.

[35] M. Papageorgiou, C. Diakaki, V. Dinopolou, A. Kotsialos, and Y. Wang, “Review of Road Traffic Control Strategies,” *Proceeding of the IEEE*, Vol.91, pp.2043-2067, 2003.

[36] D.I.Robertson,“TRANSYT Method for Area Traffic Control”, *Traffic Engineering & Control*, Vol.10, pp.276-281, 1969.

[37] P.B. Hunt, D.L. Robertson, and R.D. Bretherton, “The SCOOT On-line Traffic Signal Optimization Technique,” *Traffic Engineering & Control*, Vol.23, pp.190-192, 1982.

[38] R.P. Roess, E.S. Prassas, and W.R. McShane, *Traffic Engineering*, Prentice Hall, Upper Saddle River, NJ; 2004.

[39] S. Sen and L. Head, "Controlled Optimization of Phases at an Intersection," *Transportation Science*, Vol.31, pp.5-17, 1997.

[40] S. Algers et. al. ,"SMARTEST: Review of Micro-Simulation Models," 1997.

[41] G. Sun, C.G. Cassandras, Y. Wardi and C.G. Panayiotou, "Perturbation Analysis of Stochastic Flow Networks," *Proceedings of the 42nd Conference Decision and Control*, 2003.

[42] J.C. Spall, "Multivariate Stochastic Approximation Using Simultaneous Perturbation Gradient Approximation," *IEEE Transactions on Automatic Control*, Vol.37, pp.332-341, 1992.

[43] J.C. Spall and D.C. Chin, "Traffic-Responsive Signal Timing for System-Wide Traffic Control," *Transportation Research - C*, Vol.5, pp.153-163, 1997.

[44] C. Stamatiadisand and N.H. Gartner, "MULTIBAND96 : A Program for Variable Bandwidth Progression Optimization of Multiarterial Traffic Networks," *Transportation Research Record*, No.1554, pp.917, 1996.

[45] M.L. Tartaro, C. Toress, and G. Wainer, "Defining Models of Urban Traffic Using the TSC Tool," *Proceedings of the Winter Simulation Conference*, pp.1056-1063, 2001.

[46] Y.Wardi, B. Melaned, C. Cassandras, and C. Panayiotou, "IPA Gradient Estimators in Single-node Stochastic Fluid Models," *Journal of Optimization Theory and Applications*, Vol.115, No.2, pp.369-406, 2002.

[47] F.V. Webster, “Traffic Signal Settings,” Road Research Technical Paper No. 39, Research Laboratory, London, UK, 1958.

[48] A. Vogel, C. Goerick and, W. Von Seelen, “Evolutionary Algorithms for Optimizing Traffic Signal Operation,” *Proceedings of the European Symposium on Intelligent Techniques (ESIT 2000)*, Aachen, Germany, pp.83-91, Sept 14-15, 2000.

[49] S. Meyn, “Sequencing and Routing in Multiclass Networks. Part I: Feedback Regulation,” *Proceedings of the IEEE International Symposium on Information Theory*, pp.4440-4445, 2000.

[50] G. Sun C.G. Cassandras, Y. Wardi, C.G. Panayiotou, and G. Riley, “Perturbation Analysis and Optimization of Stochastic Flow Networks,” *IEEE Transaction on Automatic Control*, AC-49, 12, pp.2113-2128, 2004.