Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
SWUTC/14/600451-00014-1		

4. Title and Subtitle	5. Report Date
A Real-time Transit Signal Priority Control System that Considers Stochastic Bus Arrival Times	May 2014
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Xiaosi Zeng, Kevin Balke, Praprut Songchitruksa, and Yunlong Zhang	Report 600451-00014-1

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Texas A&M Transportation Institute The Texas A&M University System College Station, Texas 77843-3135	11. Contract or Grant No. DTRT12-G-UTC06

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Southwest Region University Transportation Center Texas A&M Transportation Institute The Texas A&M University System College Station, Texas 77843-3135	Research Report: May 2012–August 2013
	14. Sponsoring Agency Code

15. Supplementary Notes

Supported by a grant from the U.S. Department of Transportation, University Transportation Centers Program and by general revenues from the State of Texas.

16. Abstract

Transit Signal Priority (TSP) is an effective strategy for providing preferential treatment to move transit vehicles through intersections with minimum delay. However, TSP can disrupt traffic on non-priority phases if not properly implemented. To produce a good TSP strategy, advance planning with enough lead time is usually preferred; this means added uncertainty about the bus arrival at the stop bar, which has been difficult to be accounted for. Researchers proposed a stochastic mixed-integer nonlinear model (SMINP) to be used as the core component of a real-time transit signal priority control system. The SMINP was implemented in a simulation evaluation platform. An analysis was performed to compare the proposed control model with the standard check-in/check-out TSP system implemented in the VISSIM Built-in Ring-Barrier Controller (RBC-TSP). The results showed the SMINP produced as much as 30 percent improvement of bus delay from the RBC-TSP in low to medium volume conditions. In high-volume conditions, the SMINP model automatically recognizes the level of congestion of the intersection and gives less priority to the bus so as to maintain a minimum impact to the traffic on its conflicting phases. In the case of multiple conflicting bus lines, a rolling optimization scheme was developed. A comparison indicated the RBC-TSP systems cannot handle a high degree of saturation when there are significant amount of conflicts between bus lines, while the SMINP can automatically give less priority to bus so as to cause much less impact to other traffic.

17. Key Words	18. Distribution Statement
Transit Signal Priority, Mixed-integer Nonlinear Model, Stochastic Optimization, Degree of Saturation, Simulation Evaluation	No restrictions. This document is available to the public through NTIS: National Technical Information Service Alexandria, Virginia http://www.ntis.gov

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	91	

A Real-time Transit Signal Priority Control System that Considers Stochastic Bus Arrival Times

by

Xiaosi Zeng, Ph.D. Candidate
Graduate Assistant Researcher
Zachry Department of Civil Engineering, Texas A&M University

Kevin Balke, Ph.D., P.E.
Research Engineer
System Reliability Division, Texas A&M Transportation Institute

Praprut Songchitruksa, Ph.D., P.E.
Associate Research Engineer
System Reliability Division, Texas A&M Transportation Institute

and

Yunlong Zhang, Ph.D., P.E.
Associate Professor
Zachry Department of Civil Engineering, Texas A&M University

Report SWUTC/14/600451-00014-1

Southwest Region University Transportation Center
Texas A&M Transportation Institute
The Texas A&M University System
College Station, Texas 77843-3135

May 2014

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented here. This document is disseminated under the sponsorship of the U.S. Department of Transportation, University Transportation Centers Program, in the interest of information exchange. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ACKNOWLEDGMENT

The authors recognize that support for this research was provided by a grant from the U.S. Department of Transportation, University Transportation Centers Program, to the Southwest Region University Transportation Center, which is funded, in part, with general revenue funds from the State of Texas. The authors are also greatly in debt to Mr. Abed Abukar from the Dallas Area Rapid Transit agency for his valuable inputs that provided directions for this project. In addition, we are very appreciative to Dr. Kai Yin for his insightful comments on various aspects of model development.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. PROBLEM STATEMENT

Public transportation is a very economical, efficient, and environmentally friendly means to move a large number of travelers in an urban environment. Urban congestion due to low passenger throughput, high-rise gas prices, and heavy tailpipe emissions caused by private passenger vehicles are major problems that could be mitigated by transit vehicles. Since 2006, the United States Department of Transportation (USDOT) Urban Partnership Agreement program has been aggressively pursuing public transit as one of the four strategies to reducing traffic congestion (*Jackson et al. 2008*). In order to improve transit operations, roadway operators often seek to implement preferential treatments for transit vehicles. Some treatments provide preference to buses via modified roadway segments, such as median bus-way, exclusive lanes, and the like. Others furnish priority through locations that yield the best benefits, such as transit signal priority (TSP), queue jump and bypass lanes, curb extensions, and so on.

Two of the five factors that directly influence customers' perceptions of the transit service quality are travel time and reliability (*Kittleson & Associate et al. 2003*). Providing TSP to the transit vehicles in need is a viable strategy to reduce bus delay and improve bus reliability (*Danaher 2010*). To achieve these goals, a modern TSP system requires at least the knowledge of bus arrival times, but it can also make use of additional information such as passenger load and schedule adherence. Researchers focused the discussion on bus delay at single intersection. The team chose not to incorporate bus schedule information, although it can be easily done, because the discussion of reliability is only meaningful at a corridor/route level.

The state-of-the-practice TSP systems typically use a pair of check-in and check-out detectors to determine a short period of time during which an active priority signal strategy is implemented, after which a recovery strategy is deployed on the next cycle to return time taken from cross-street phased . A popular choice for preferential signal treatments is transit vehicles due its simplicity. However, such systems are not known to handle multiple bus requests, heavy traffic conditions, or uncertain arrival times. (See section 2.1 for why these are inherently difficulty problems for traditional TSP systems.) In reality, these common problems unavoidably dampen the effectiveness of this kind of TSP operation.

Better TSP control strategies are needed. Many adaptive TSP models have been developed by various researchers in recent years. Most of these models employ mathematical programming techniques to systematically search for optimal timings, which yields good balance between providing priority to the transit vehicles and depriving right-of-ways from the conflicting traffic. As effective as they are, these models normally rely on accurate predictions of bus arrival time, which sometimes can be problematic, especially when a near-side bus stop is present. Stochastic arrival times may easily affect the quality of the priority timings, which in turn impacts the effectiveness of the deployed TSP systems. In literature, very few studies mention about stochasticity of bus arrival times in their model development process, let alone account for it.

Finally, many adaptive TSP systems use point-sensors to provide inputs to their core algorithms/models. Information from point-sensors is sometimes too discrete that assumptions about what occur in between these sensors need to be made. When these assumptions are violated, the algorithm fails. For example, a bus is assumed to travel for 15 seconds from the check-in to the check-out detector. If it is following a very slow vehicle, it may not make it through the intersection while priority is given regardless. Continuous surveillance systems such as the connected vehicle technology may provide mitigations to assumption making and even ultimately lead to better models.

1.2. RESEARCH SCOPES AND OBJECTIVES

The overarching objective of this research is to develop a real-time signal control system that addresses all the aforementioned problems and that is practical as well as easily implementable. Specifically, the following objectives are to be achieved:

- Develop an adaptive TSP model, using math programming methods, that is able to accommodate the priority needs of multiple transit buses simultaneously at a single intersection without seriously disrupting other traffic.
- Incorporate the uncertainty principle in the model development process, so that inherent uncertainty of the input information can be explicitly accounted for.
- Investigate the impacts of near-side and far-side bus stops on the bus arrival time at the stop bar and explicitly account for the impacts in the real-time control framework.
- Develop a traffic simulation platform to facilitate the real-time evaluations of the proposed model. Evaluate the mathematical model in an offline and an online

environment and compare the performance of the proposed model with the state-of-the-practice TSP system.

- Identify if and how the connected vehicle technology can be used in the development of the real-time signal control system with adaptive TSP.

In the literature, the first two objectives are separate. There is research for better timing optimization models and other research for better prediction models. In principle, in order to successfully implement a change of timing that is not too disruptive to the general traffic, planning in advance is the key. To one extreme, if all information can be precisely known hours ahead, optimal timing can be guaranteed. However, uncertainty of the input information, such as bus arrival time, grows as one looks further into the future. To avoid dealing with uncertainty, one might wait until just before actual bus arrivals when uncertainty becomes practically negligible. But last-second planning may easily lead to either no flexibilities for timing adjustments or very disruptive timing adjustments. Traditionally, there is a choice between certainty and flexibility. Developing a model that explicitly accounts for uncertainty alleviates the conflicts between these two equally favorable traits of any good system design. Therefore, the first two objectives serve the purpose of bridging this gap.

The existence of bus stops further obscures the predictability of the bus arrival time at the stop bar. The dwell time at bus stop is a significant source of uncertainty for bus arrivals. It is possible to study a single intersection without a bus stop. But models that do not account for bus stops, especially the inherent randomness of dwell time, may not be easily extendible if corridor studies are preferred. Bus stops and the associated dwell times are important elements that a bus will encounter along its route, so they need to be explicitly accounted for. Bus dwell times are typically not a main subject in the literature largely due to the lack of models that are capable of dealing with uncertainties. One of the objectives of this research is to incorporate this element in the development of a stochastic model.

In addition to modeling, it is vital to evaluate the effectiveness of the proposed model and whether it is feasible to be deployed in a real-time traffic environment. The offline evaluation helps to demonstrate certain features of the model independently. However, to achieve real-time control capability, a real-time control scheme that incorporates the mathematical model needs to be developed and evaluated in a real-time traffic environment. The third objective also entails the

development of a simulation test bed that has built-in traffic flow models and signal control components.

Finally, the development of simulation test bed is envisioned to emulate the connected vehicle communications functionalities. The connected vehicle technology uses high-frequency and low-latency vehicle-to-infrastructure wireless communications. Completely different from the traditional point-sensors, this technology provides continuous collections of a wide range of data elements from the subject vehicles. This research explores some of the useful data that can facilitate the modeling and the implementation of the real-time TSP control strategies.

2. BACKGROUND

Transit Signal Priority is an operational signal control strategy whose main purpose is to facilitate transit vehicles passing through intersections. Since its earliest implementation in 1968 (*Evans and Skiles 1970*), TSP has reduced transit delay at intersections, improved transit on-time performance, and maximized intersection person throughput. In recent years, TSP strategies and deployment has been growing in the United States and around the world. Comparing to other transit preferential treatments, a TSP system usually requires relatively minimal infrastructure upgrades and may quickly increase roadway's capacity for buses (*Kittleson & Associate et al. 2003*). It is among one of the most cost-effective preferential treatments that has been widely implemented.

One goal of transit as public transportation is to move people quickly from one place to the other. Travel time is a key service measure of a transit vehicle in the system (*Kittleson & Associate et al. 2003*). In addition, faster route travel time means shorter turn-around time, which may help save transit agencies' investment on adding more buses to maintain a schedule. Accordingly, a TSP system can help cut down the signal delays to the buses. Once a schedule is developed and published, another important goal of transit agencies is to maintain good operational reliability—on-time arrival. In this respect, a TSP system provides quicker access to buses that are late.

2.1. TRADITIONAL TSP STRATEGIES

Traditional strategies can be divided two sub-categories: passive and active priority strategies. Passive priority strategies rely on historical data and do not require any detection system. By assuming non-variable arrival time of the transit vehicles, signal timing settings (i.e., green times and cycle lengths) can be optimized for transit priority. Then, signal priority is provided unconditionally every cycle. A transit-based signal coordination scheme is a good example of where signal progressions are computed based on the speed of buses. This method is easy to implement and does not require a transit detection/priority request generation system (*Smith et al. 2005*). The method is effective when bus arrival patterns are regular and frequent. In most other cases, this treatment causes unnecessary or even excess delays to conflicting traffic.

To avoid making assumptions about bus arrivals, bus priority can also be provided only when a transit bus is detected. It provides effective priority to transit vehicles through more efficient signal operations; it can even allow selective provisions of priorities on a need basis rather than for all detected transit vehicles. There are four basic active priority strategies: phase extension, red truncation, phase insertion, and phase suppression (*Kittleson & Associate et al. 2003*).

- Phase extension—hold the green until the transit vehicle clears the intersection.
- Red truncation—advance the start of the green for the phase(s) serving transit vehicles.
- Phase insertion—insert a new phase that can serve the transit vehicle at the moment it arrives at the intersection.
- Phase suppression—skip one or more phases that are conflicting with the priority phase in order to give green time to the priority bus request earlier. Usually, the suppressed phases will be given back later in the cycle. It can also be called phase rotation strategy.

Advanced techniques based on these strategies were proposed and studied. Balke (*1998*) provided an excellent and comprehensive review of many of the advanced TSP techniques. Comparing to passive strategies, the state-of-the-practice active TSP systems rely on fix-point sensors (i.e., check-in/check-out system) to detect the approaching buses and later confirm the departures of the buses. Such approach of providing priority is straightforward, and the system can be relatively easy to setup. When used in the right conditions, active TSP systems are found to yield as much as a 34 percent decrease in bus delay in large metropolitan areas such as Seattle (*Danaher 2010*). Active TSP priority strategies have gained large popularity and become the industry standard that has been implemented in many modern traffic signal controllers, such as Econolite ASC/3 controllers.

Active TSP is not known for handling multiple conflicting priority requests at the same time. One primary reason is the enforcement of the First-Come-First-Serve (FCFS) policy and the need to recover afterward (*Econolite 2009*). Once the priority is given to the first arriving bus, no further priority requests can be processed even if these later requests have more urgent needs. Zlatkovic et al. (*2012*) showed that the FCFS policy may be worse than a policy that provides no priorities at all, and they further developed an algorithm to circumvent the problem for relatively simple cases of conflicting priorities. Another main reason for the difficulty in handling multiple requests is attributed to the reliance of active TSP systems on a well-defined set of decision

rules. It means an active TSP system works as well as its designer can envision, and the system is not likely to work well when unexpected situations arise.

Even though complex algorithms can be carefully designed, none of the existing active TSP systems is known to handle uncertain transit vehicle arrival times. One main reason is that any decision rule method is itself a deterministic process that is based upon deterministic inputs and produces deterministic outputs. However, uncertainty is generally not a problem when the time between priority request detection and service is very short. For example, 15 seconds from the transit vehicle check-in to its check-out. However, having a short lead time in heavy traffic conditions means only one of the two things: (1) no flexibilities for timing adjustments or (2) very disruptive timing adjustments. The ability to account for uncertainty is crucial to overcome the limitations of short advance planning time. Uncertainty can be accounted for only if it can be explicitly modeled. As the next few sections show, an adaptive TSP system that employs mathematical models is capable of modeling uncertainty explicitly. Another reason for the inability of an active TSP strategy to account for uncertain arrival time is the use of the check-in/check-out detection mechanism; see section 2.4 for details.

2.2. ADAPTIVE TSP STRATEGIES

Adaptive priority, sometimes called real-time priority (*Kittleson & Associate et al. 2003*), usually refers to a TSP strategy whose control decision is derived from mathematical models that found their basis on the theory of optimizations. The models, at minimum, use the arrival information of buses to optimize main signal timing parameters (e.g., green duration, max/min green cycle length). This type of priority model takes a systematic approach to make the best decisions that can take all traffic into consideration simultaneously. It was difficult to implement since the cost for real-time computation was expensive decades ago. With the advancement of electronic technologies, modern solid state traffic control systems became more flexible and computationally capable to accommodate the implementation of adaptive priorities. Recent research has focused more on designing intelligent priority models. To address TSP problems, there are at least two objectives:

- Models that aim at minimizing bus as well as vehicle delays.
- Models that strive to maintain bus headway deviation.

Table 1 summarizes recent works that use mathematical programming and artificial learning approaches to address the TSP problems. Detailed descriptions and the main problems with current research are provided in the following subsections.

Table 1: Summary of Primary Literatures for TSP.

Year	Authors	Model	Considerations for Transit Vehicles						Considerations for Non-Transit Vehicles			Active TSP Capable			Controller Considerations			
			Uncertain Arrival Time	Interval Arrival Time	Multiple Requests	Position in Queue	Headway Deviation	Passenger Load	Residual Queue	Person Delay	Stochastic Arrival	Green Extension	Red Truncation	Phase Insertion	Ring-Barrier	Coordination	Phasing Sequence	Actuated Control Capable
2008	Qing He	MILP		x	x			x		x		x	x		x	x		x
2008	Stevanovic	GA	x								x	x	x		x	x	x	
2011	Li Meng	MINP				x		x	x			x	x		x	x	x	
2011	Eleni Christofa	MILP			x	x		x	x	x		x	x					
2012	Qing He	MILP			x			x		x		x	x		x	x		x
2012	Wanjing Ma	DP			x		x	x				x	x	x				
2004	Ling & Shalaby	RL					x											
2005	Vasudevan	DP			x		x	x		x	x	x	x			x		
2011	Tlig & Bhouri	MA-AI		x	x	x	x					x	x	x		x		

MILP – Mixed Integer Linear Programming
MINP – Mixed Integer Nonlinear Programming
RL – Reinforce Learning
GA – Genetics Algorithm
DP – Dynamic Programming
MA-AI – Multi-Agent Artificial Intelligence

2.2.1. Minimizing Delay

The original objective of transit signal priority is to allow higher-priority vehicles to pass through a signalized intersection as quickly as possible. A large number of research projects started with this objective.

Li et al. (*2011*) presented an adaptive TSP optimization model that optimizes green splits for three consecutive cycles to minimize the weighted sum of transit vehicle delay and other traffic delay, considering the safety and other operational constraints under the dual-ring structure of signal control. By computing not only the green but also the red time for each phase, the model was able to capture the evolution of TSP-induced queues and their delays using deterministic queuing theory. Due to the nonlinear nature of phase red-time and vehicle delays, the optimization model is Mixed Integer Nonlinear Programming (MINP). A field study showed a 43 percent bus delay reduction and a 12 percent delay increase on passenger cars.

Christofa and Skabardonis (*2011*) presented a traffic responsive signal control system for signal priority on conflicting transit routes that also minimizes the negative impacts on the auto traffic based on person delay. The vehicle delays are estimated using deterministic queuing theory, where arrivals and departures are constant. The position of a bus in a vehicle queue is explicitly modeled to obtain the bus delay. In addition, the passenger load of each bus is used as the weighting factor among multiple priority calls as well as between bus and passenger vehicles.

Stevanovic et al. (*2008*) presented a genetic algorithm model that works in a micro-simulation environment to optimize four basic signal timing parameters (i.e., cycle length, offset, splits, and phase sequence) and transit priority settings. The objective of the optimization is the sum of total delay and weighted number of stops for all vehicles. Two TSP strategies are made possible by optimizing the transit priority parameters: green extension and red truncation. Taking advantage of the random seeds in the micro-simulation, the stochasticity characteristics of vehicle arrivals are implicitly addressed.

Ma et al. (*2012*) developed a TSP control framework that uses a dynamic programming approach to determine a timing plan with minimal bus delays. In a multi-request scenario, each request is weighted by bus occupancy and schedule deviations. Three active priority strategies are explicitly modeled: green extension, red truncation, and phase insertion. Although the delay to non-transit vehicles are not computed, the degree of saturation is set as a constraint to ensure the impact to other traffic is not too large. The framework further implements a rolling horizon approach to enhance its real-time control capability. A simulation study showed up to a 30 percent reduction of bus delays compared to fixed time control with no TSP implementations.

He et al. (*2012*) proposed a unified platoon-based framework called PAMSCOD that considers multiple models of travel, excluding pedestrian and bicyclists. The framework includes a Mixed Integer Linear Programming (MILP) model that searches the optimal signal plan by feeding priority requests (buses and/or vehicular platoons) and phasing data to signal controller in real-time. The objectives of the optimization model are to minimize the total of bus and platoon delays and to maximize the slack green time. The slack green is the extra green time available for a typical actuated controller to extend phases until gap-outs or max-outs. This method addresses the shortcoming that an adaptive signal controller usually operates on a fixed split basis, which cannot take advantages of industrial-standard controllers that are based on vehicle actuations.

2.2.2. Minimizing Headway Deviation

With a bus schedule published, transit agencies generally strive to closely operate the buses to the schedule. The consistency between the actual and the published arrival times of a bus along its route is a primary measure for quality of service from a passenger's point of view (*Kittleson & Associate et al. 2003*). Maintaining bus headways are also important to resist a notorious transit operational problem called bus bunching. Bus bunching decreases the bus capacity utilization and causes further delays to passengers (*Kittleson & Associate et al. 2003*). The TSP has the potential to help pull or push bus operations to maintain bus service regularity and alleviate the bus bunching problem. Following this idea, the other research direction looks into using TSP to minimize headway deviation.

Ling and Shalaby (*2004*) used a reinforcement learning (RL) algorithm to optimize the duration of each signal phase such that transit vehicles can gradually recover to the scheduled headway. By pairing up the current phase status and the bus schedule deviation, an RL agent calculates the best phase duration while taking into consideration all practical phase length constraints. A simulation study reported that the RL algorithm brings down the headway deviation by more than 20 percent.

Vasudevan (*2005*) proposed a real-time robust arterial signal control system that consists of three levels: progression control, intersection control, and bus priority levels. The first two levels determine the progression bands and corresponding bandwidths. Using decisions from the first two levels as constraints, the bus priority level employs a dynamic programming scheme to minimize passenger, vehicle, and bus schedule delays.

Tlig and Bhouri (*2011*) developed an innovative multi-agent system that simultaneously regulates general traffic and promotes bus service regularity. The system employs four agents: bus agent, bus route agent, intersection agent, and stage (phase) agent. A set of protocols are established for each agent to compute their own properties and to communicate/negotiate with other agents. The priority of a bus is modeled by its schedule lateness. Four TSP strategies are possible: extension, truncation, phase insertion, and rotation. A simulation study was conducted on a network with six intersections and three bus routes. The results showed the proposed method gives the lowest bus headway deviation.

2.3. ADAPTIVE TSP CONTROL SYSTEMS

A signal control system that implements an adaptive TSP strategy is usually an adaptive signal control system itself. An adaptive traffic signal control system entails an algorithm that uses vehicle detections to predict the real-time traffic conditions, during which controllers optimize the signal plan to achieve shorter queue and lower delay (*Koonce et al. 2008*). In the literature, there are generally two paradigms of controls for an adaptive signal control system: (1) the control that uses a binary approach, and (2) the control that uses a rolling horizon approach. These two paradigms are sometimes called acyclic and cyclic controls (*Conrad et al. 1998*).

The binary approach (acyclic control) makes very short-term prediction, such as 2 or 3 seconds, and computes the performance index of switching the signal state and that of keeping it. The signal light is switched only if switching is determined to be advantageous. SCATS (*Sims and Dobinson 1980*), OPAC-1(*Gartner 1982*), auction-based control (*Box and Waterson 2012*), ISD-based (*Qiwu and Jianguo 2012*) control are typical examples of the adaptive signal system that adopts this approach. However, the binary control makes incremental short-term decisions that may not necessarily be optimal in the long run. The overall system optimum is not guaranteed.

The rolling horizon approach (cyclic control) employs advanced prediction modules to determine how traffic conditions may evolve in a time horizon that are normally in the multiples of cycle lengths (e.g., 100, 120 seconds). The predicted traffic condition is input to an algorithm or a math models so that key signal parameters, such as cycle length, splits, and offsets, can be optimized. The prediction and optimization routines are performed every few seconds to keep the timing updated, e.g., MOVA, SCOOT (*Hunt et al. 1982*), SPPORT (*Yagar and Han 1994*), RHODES (*Head et al. 1992*), ACS-prototype, and OPAC-2 (*Gartner et al. 1991*).

2.3.1. Existing Adaptive Signal Systems with TSP

Not many adaptive signal systems explicitly address the priority problem of transit buses at an intersection. Although it is possible to make minor modifications to these systems to accommodate buses to some degree, considering TSP in the design phase of an adaptive system would greatly help in the true implementation of TSP-capable signal system. The following are the popular systems that can be found in the literature.

- Urban Traffic OPtimization by Integrated Automation system (UTOPIA) (*Mauro and Taranto 1990*) is an adaptive signal control system in Turin, Italy. The system provides unconditional priority to selected bus routes by continuously optimizing the signal settings on a rolling horizon and simultaneously improving mobility for private vehicles.
- The Real-Time, Hierarchical, Optimized, Distributed, and Effective System for traffic control (RHODES) is a network adaptive control framework first presented by Head et al. (*1992*). The system provides a complete solution from network flow estimation to local signal timing generations. The signal phasing and timing for an intersection are optimized with consideration of delay, stops, and queue using a dynamic programming approach (*Sen and Head 1997*).
- Signal Priority Procedure for Optimization in Real Time (SPPORT) (*Yagar and Han 1994, Conrad et al. 1998*) is a rule based model that provides transit priorities from rolling decision renewal process. The system analyzes the individual priority levels of all incoming requests and computes the aggregated priority level for each phase. A set of possible signal plans is determined for a planning horizon (e.g., 90 seconds) based on current signal status. The plan with the lowest performances (such as delays to the highest priorities), which are evaluated in simulations, is implemented for the next implementation period (e.g., 5 seconds).

2.3.2. Critical Design Factors for an Adaptive TSP System

2.3.2.1. Impacts to Other Traffic

Studies listed above (e.g., the field studies [Li et al. (*2011*)] or the simulation evaluations [Ma et al. (*2012*)]) all confirmed the trend that the attempt to reduce bus delay will necessarily increase the delay to other vehicles, especially those on the conflicting phases. Almost all adaptive TSP studies involve some forms of considerations of the TSP impacts to other traffic in their mathematical formulations. The deterministic queuing model presented by Christofa and Skabardonis (*2011*) directly compute vehicle delays through the queuing polygons. Li et al. (*2011*) also applied a similar queuing model to derive the formulation of auto and bus delays as a function of the green time durations. Given the strict assumptions hold, it is more accurate to compute vehicle delays directly. However, these computations can be cumbersome, especially when oversaturation is temporarily allowed during parts of the planning horizon. There are other

ways to simplify the estimation of TSP impacts to other traffic. Ma et al. (*2012*) set the maximum degree of saturation as a constraint to put a lower limit to the feasible green time for each phase. The platoon-based framework proposed by He et al. (*2012*) used yet another approach to limit or alleviate the impacts to other traffic. The model simultaneously minimizes bus priority delay and maximizes the slack green times, then allocates the slack times to the movements with higher demands. Regardless to the ways that impacts in other traffic are modeled, some form of estimation is also important because it gives the system managers/operators the ability to assign weights to the respective traffic flows. It is then possible for users to apply *a priori* to influence what kind of the priority service policy [e.g., first-come, first-served (FCFS), first-come, last-served (FCLS)] will be used based upon real-time varying situations.

In this study, degree of saturation for each phase is used as a main parameter to proxy the vehicle delays on each phase for at least two reasons: (a) the computation of this parameter is very easy and it serves as a good proxy to vehicle delays under under-saturated to close-to saturated conditions; (b) the degree of saturation and the green duration of a phase are inversely proportional given a fixed cycle length or design period, optimizing one variable as adequate. Combining (a) and (b), one can see that the degree of saturation serves as a bridge between the decision variable (i.e., green time) and a variable of interest for performance measure (i.e., delay), which in principle simplifies the construction of and improves the solvability of any mathematical programs.

2.3.2.2. Randomness of Arrival Time

Another key design factor for successful transit priority implementation is the ability to accurately predict the arrival time of the bus at the stop bar (*Chin-Woo et al. 2008*). Models have been developed to estimate vehicle arrival times along a corridor (*Dailey et al. 2001*,*Chien et al. 2002, Cathey and Dailey 2003*). After all, if a bus that was projected to arrive at the stop bar does not arrive, the extended green time is wasted. However, almost all the studies listed above are based on the assumption that bus arrival times can be accurately predicted and can be used as fixed inputs. This assumption is risky at best. It is not hard to argue that all traffic arrivals are subject to at least some degree of randomness, and the degree of uncertainty increases as one looks further into the future. Making matters worse, it seems a better time plan can only be

devised if the planning/design process can be carried out as early as possible. Early detection of a transit vehicle is the key to provide more time to adjust the signals to provide priority while minimizing traffic impacts (*Smith et al. 2005*). These two keys (i.e., the need for early planning and the reliance of accurate arrival information) to successful implementations have been an inevitable design conflict that usually leaves the engineers no option but to choose one. Therefore, the state-of-the-practice TSP system that uses check-in and check-out system assumes very short advance duration from the time of detection to the time of service (e.g., 10–20 seconds), which minimizes the uncertainty of the input information to their planning process.

The significance of arrival uncertainty is also recognized by other researchers. He (*2010*) argued that it is important to consider the fact that a bus may not arrive precisely at the time that was predicted. To consider the arrival uncertainty, an interval arrival time was used instead of a point arrival time. A robust optimization model was developed building on the precedence model by Head et al. (*2006*). However, the robust optimization was originally from Wald's maximum model to treat severe uncertainty. Simply put, it is designed for worst case scenarios. Using robust optimization to address the randomness of bus arrival inevitably leads to the tendency to select larger green time for the priority phase. This method is designed for when uncertainty is relatively low (e.g., 3–5 seconds standard deviation). Stevanovic et al. (*2008*) also expressed the importance of modeling randomness in the design of bus priority schemes. Their approach is simulation-based, which evaluates the performance of a finite number of strategies implemented in a finite number of scenarios and selects the one with the best performance measure. Such approach is interesting and inherently incorporated the traffic flow models into their planning/design process. However, the testing any one scenario and/or one strategy will require full-scale simulations of all the traffic into the future, which can be computationally cumbersome for any practical consideration.

Although the above studies provide an initial solution to address the uncertainty of bus arrival times, their approaches are effective when only a limited variation of arrival time or a limited number of scenarios exist. However, when a bus travels along a corridor, one of the major contributors to bus arrival time uncertainty is the time it dwells on various bus stops. Although this study looks at only a single intersection, ignoring the possible impacts of bus stops on bus arrival time will make the current study useless when one attempts to extend the work to a corridor case. It is not difficult to see that when a bus stop exists, the interactions among the

vehicle queues, the bus travel time, and the bus dwell time become very complicated. This complication is problematic to pinpoint exactly when a bus will need a green to pass through the intersection. As a result, little research has considered bus dwell time and its stochasticity during the prediction process of the bus arrival time at stop bar. Current practice typically ignores or circumvents the problem. For example, the user manual of the RBC controller in VISSIM (*PTV America 2010*) recommended placing the detector at the exit of the bus stop and sending a departure signal to the controller when the bus closes its door or exits from the bay. Such approach eliminates the need to consider bus dwell time. However, this work-around approach inevitably cuts too much valuable time from the planning process and leaves any control strategies very little room to implement a good timing plan. In summary, the two keys to successful implementation of a bus signal priority system require the ability of the system to capture the uncertainty of the bus arrival times in its planning/design process.

2.3.2.3. Real-Time Capability

Any practically useful signal control system, either adaptive or non-adaptive, would require real-time capabilities, which have to include at least three components: detection, planning, and implementation. A real-time adaptive TSP system also requires these components. The detection of one or multiple buses approaching the intersection of interest initiates the TSP control sequence. Immediately following approaching detection, a planning procedure allows the identifications of the best timing based upon the arrival information using an existing model or a priority policy. A planning procedure shall be equipped with some form of predictions that generate a predicted bus arrival time at the stop bar. With a good or the best timing determined, a real-time system will be able to implement the new timing based upon what has occurred and what is currently ongoing in terms of signal timing. A true TSP online system will have these three processes cycled automatically on a continuous basis.

2.4. CONNECTED VEHICLE TECHNOLOGY

As mentioned previously in section 2.1, another reason for the inability of an active TSP strategy to account for uncertain arrival time is due to the limitations inherent to the check-in/check-out mechanism. The most critical information that a pair of check-in/check-out detectors collects is the bus arrival and departure time. That means the TSP system has no information about the

location, speed, and operations of the bus in between these two detectors. Assumptions need to be made for any timing adjustment strategies to build on. Typically, constant travel time from the check-in to check-out detectors is assumed. Optional expiration time can be added to prevent excessive impacts to conflicting movements and/or the failure of detectors. The period is rather short between when the bus is first detected and the bus actually needs a green signal. For such a short look-ahead time period, there may not be many feasible timing adjustment strategies. Consequently, it is likely that either the bus does not get priority or the conflicting traffic gets disrupted seriously. The positive benefits of signal priorities on bus delays are often just moderate, and some reported benefits can be as low as a 1.1 percent delay reduction (*Danaher 2010*).

The Connected Vehicle (CV) technology refers to the suite of wireless communications technologies that greatly facilitate the transmission of information between vehicles and any CV-enabled devices (V2X). This technology provides a continuous surveillance mechanism that overcomes the limitations of the check-in/check-out detector system. Continuous data flow gives an adaptive TSP system options to make necessary changes to model inputs when actual bus arrival times deviate significantly from the original prediction. In essence, the CV technology enables an updated system that can rectify actions if the original assumption about vehicle arrival is far off. In other words, the consequence of making a wrong assumption is greatly mitigated when the CV technology is in place.

2.4.1. Overview of the Technology

In particular, the dedicated short-range communications (DSRC) provides short range, low-latency, high reliability two-way transmissions of digital contents over the air. Large amounts of information can be exchanged for traffic safety and mobility applications. The design range of a typical DSRC unit is about 3,000 feet (1,000 meters), and the actual range may be less than 1,000 feet (300 meters) due to line-of-sight obstructions and other environmental varieties (*Andrews and Cops 2009*). In order to promote the standardization of using the DSRC protocol, the Society of Automobile Engineers (*2009*) complied a message set dictionary. The probe vehicle data (PVD) message is a standard message in the dictionary and is used for a vehicle to send vehicle attributes and a snapshot of the recent vehicle's running status to a roadside DSRC

unit. Each snapshot can support up to 42 vehicle data elements, including basic and customized vehicle operational statistics.

2.4.2. TSP Control Using Connected Vehicle Technology

Current detection systems for bus priority provide mostly point detection, but the connected vehicle technologies can be deployed to continuously monitor the position, speed, and other parameters of both private vehicles and buses. Researchers have used the connected vehicle data set differently in aid of signal controls.

One way is to use the enriched data set to provide more accurate arrival predictions. He et al. (*2012*) proposed a unified platoon-based framework called PAMSCOD that optimizes real-time signal timing for multiple modes of travel. The foundation of the framework is a hierarchical platoon recognition algorithm that fuses detailed motorized vehicle data collected using the connected vehicle communications. The fusion of data extracts the most critical information and simplifies the data input into an optimization model. The vehicle clustering algorithm developed by Smith et al. (*2010*) at the University of Virginia is another such example.

Another direction is to redesign the signal control paradigm along with the TSP algorithms based on the characteristics of the connected vehicle data set. Smith et al. (*2010*) developed a predictive microscopic simulation algorithm (PMSA). The PMSA collects speed, heading, and location from all CV-enabled vehicles and uses them to recreate the intersection in micro-simulation. Then the simulation is repeatedly run for the next 20 seconds by only changing the control parameters, such as green duration and phase sequence that are allowed within the 20 seconds. The set of control parameters that gives the lowest delay will be implemented in the intersection controller. Although this algorithm was not intended for TSP control, the principles are the same.

Redesigning a signal control paradigm that fully makes use of the CV technology may still be unrealistic or impractical for a while because the algorithms that use CV data to manage the right of way at an intersection require a very high percentage of market penetration. Practically speaking, there will be a long period of time that the CV technology and the traditional traffic detection infrastructure coexist to provide estimations on traffic conditions. It makes more sense to develop models and algorithms that help bridge the gap between current and future traffic signal control systems. Following this realization, this research assumes the buses are all

equipped with the connected vehicle technology, and the bus instantaneous information (e.g., speed, location, heading, and passenger count) are available any time within the coverage of the CV-enabled infrastructure.

3. BASIC MATHEMATICAL MODEL

The heart of an adaptive TSP system is a functioning optimization model that searches for the best timing based on a series of inputs such as bus arrival times and current signal timings. This section first provides a general background on stochastic programming, followed by a subsection that describes how the stochastic programming framework is applied in modeling the adaptive TSP system. The features and characteristics of the model are demonstrated in the last subsection through a series of proof-of-concept experiments that are conducted in an offline environment.

3.1. TWO-STAGE STOCHASTIC PROGRAMMING

Stochastic programming is a branch in the field of mathematical programming specifically for modeling optimization problems that involve uncertainty. Deterministic programming models consider parameters that are well-known at the time of the modeling, whereas data collected in the real-world typically are not precisely known in advance. Originated from Dantzig (*1955*) and Beale (*1955*), a stochastic mathematical program is to find an (expectedly) optimal solution to a problem by explicit modeling of parameter uncertainties that can be characterized by some probability distribution functions. Stochastic programming techniques have been applied in many areas including vehicle routing, fleet assignment (*Ferguson and Dantzig 1956*), production planning (*Charnes et al. 1958*), to name a few. It is out of the scope of this report to fully discuss the theory and background of the stochastic program. Birge and Louveaux (*1997*) provided an excellent review of the fundamentals of stochastic programming.

In one of its simplest forms, a stochastic program typically consist of two stages, each of which can be thought of a particular timeline in the decision making process. Stage one is the "now" stage that corresponds to the time that one has to make a decision on a set of decision variables. Let x denote an n_1-element vector of first stage decision variables. All parameters related to the x can be collected precisely and deterministically, and can be formulated in the now stage. Stage two is the "future" stage that represents processes that would occur in the future. In this stage, because these future processes have not been observed yet, researchers cannot use the associated parameters as deterministically as the now processes. To make a now decision, which is not compatible with the future process, researchers need to pay a cost for such a

decision. This cost is generally termed as the recourse cost, quantified by the second stage decision variable *z*. To summarize the recourse costs as a function of the now decision and the future processes (called recourse function, denoted as $f(\cdot)$), then a best strategy is to find the now decision that minimizes the recourse costs under all future scenarios.

Depending on the integrality requirements and the parameter uncertainties, there is a large variation of stochastic programs. Here is a brief mathematical description for a generic two-stage stochastic program model:

$$\begin{aligned} &\text{Min} && c^T x + E[f(x,\tilde{\omega})] \\ \text{Stage 1:} \quad &\text{s.t.} && Ax \geq b \qquad [0] \\ & && x \geq 0 \end{aligned}$$

where *c*, *A*, and *b* are parameters with known value at the timing of decision making, while $\tilde{\omega}$ is a random parameter defined on a probability space (Ω, F, P). $f(\cdot)$ is the recourse function that gives the penalty of a selection of second stage decision variable on the first stage objective function. E[·] denotes the expected value that is finite and evaluable. Therefore, for a given x and an outcome $\omega \in \Omega$, the recourse function can be written as:

$$\begin{aligned} f(x,\tilde{\omega}) = &\text{Min} && q^T z \\ \text{Stage 2:} \quad &\text{s.t.} && Wz \geq r(\omega) - Tx \qquad [0] \\ & && z \geq 0 \end{aligned}$$

where *z* is an n_2-element vector of second stage decision variable; *q*, *W*, and *T* are parameter matrices that do not vary according to the realization of scenario ω, while r is the parameter matrix that do vary for scenario ω. There can be a large number of second-stage programs, the recourse function in each of which needs to be evaluated. Associating the values of the recourse functions for all scenarios with probabilities accordingly, the expected function value in the first stage can by calculated. Via a well-designed iterative scheme and under certain conditions, the overall objective function value in [0] can be minimized.

3.2. PRIORITY SIGNAL CONTROL MODEL FORMULATIONS

In a TSP control system, advance planning is the key to any successful strategies. Once detected upstream, the signal control system may need to decide if timing adjustments will be needed to prepare for the arrival of a priority vehicle. To know precisely the second that the priority bus will arrive, it is immediately apparent what timing should be implemented. However, when the

arrival time of the bus is not certain, the decision for a certain timing to be implemented now may or may not be the consistent with the actual bus arrival time in the future. It is easy to compute the bus delay would occur if a timing is chosen that is not consistent with the actual bus arrival time. The bus delay can be thought of the recourse cost, which is a function of the now decisions of signal timing and the future bus arrival time. Following this logic, a stochastic two-stage mixed integer nonlinear program (SMINP) for a typical TSP problem can be built. Before formal formulations of the two stages, define all the variables that are needed for the formulations.

3.2.1. Variable Definitions

Sets

J the set of all phases.

K the set of cycles within the analysis horizon , usually K=2 or 3.

Decision variables

t_{jk} the start time for phase j of cycle k.

g_{jk} the green time for phase j of cycle k.

v_{jk} the split for phase j of cycle k.

y_{jk} the deviation of green time on phase j of cycle k from optimal green time.

d_j the priority delay of a bus requesting for phase j.

θ_{jk} the priority service decision for a bus at phase j of cycle k.

Data

C cycle length.

c_{jk} weighting parameter for green time deviation of phase j of cycle k.

Y,R yellow time and red clearance time.

S_j the saturation flow rate on phase j.

X_{jk} the degree of saturation for phase j.

g'_{jk} the background green time for phase j of cycle k.

$g_{jk,\min}$ the minimum green time for phase j of cycle k.

BR_j the cycle time of a bus arrival on phase *j*.

t'_{jk} background start time for phase *j* in cycle *k*.

V_{jk} the average flow rate for phase *j* in cycle *k*.

3.2.2. First-Stage Model

The decision variables in the first stage are the timing parameters, such as green splits and cycle lengths for the planning horizon. Therefore, the formulation in this stage shall realistically model the behavior and the characteristics of the signal controller in question. Head et al. (*2006*) proposed a precedence relationship to model the United States' standard ring-barrier signal timing structure. The precedence model constrains the structural relationships among different phases. Many of the contemporary traffic signal logics can be easily realized under this modeling framework. Later, He et al. (*2012*) applied the framework to develop a deterministic priority model which minimizes the delay of priority requests. This research directly applies their basic framework with certain modifications.

3.2.2.1. Objective Function

The first stage objective function can be considered as the overall objective function that also considers the expected recourse cost computed from the second-stage objective function. The objective function minimizes the sum of the changes in green times for all phases and the expected delay of the priority request. It is formulated as follows:

$$\text{Minimize:} \qquad \sum_{k\in K}\sum_{j\in J} c_{jk}\, y_{jk}^2 + E[Q(\boldsymbol{t}, \boldsymbol{v}, \overline{BR})] \qquad [0]$$

The weight, c_{jk}, on first term controls the distributions of priority needs in terms of seconds among all the conflicting phases. For example, if 10 seconds total green times from the two conflicting phases are required, and the weights are equal, then both phases can be compressed for 5 seconds. In doing so, phases that are more congested may be shortened less from the optimal green time comparing to those phases that are less congested. The use of the quadratic function on the deviation variable, y_{jk}, gives the math program the ability to penalize higher deviation values. The second term, $E[Q(\boldsymbol{t}, \boldsymbol{v}, \overline{BR})]$, is an expectation function of the recourse function, $Q(\boldsymbol{t}, \boldsymbol{v}, \overline{BR})$. $\boldsymbol{t}$, $\boldsymbol{v}$ are vectors of start times and splits of all phases, respectively, which

are variables obtained from the first stage formulation. $\overline{BR}$ is a random parameter of the bus arrival time.

3.2.2.2. Constraints

Constraints in the first stage are mostly defined for the precedence relationships of all phases of all look ahead cycles within the planning horizon. The validity and illustration of the precedency are clearly documented in Head et al. (*2006*) and He et al. (*2012*). The phase relationships are formulated as:

$$t_{1,k} = 0; \qquad \forall k \qquad [0]$$

$$\begin{aligned} &t_{2,k} = t_{1,k} + v_{1,k}; \quad t_{3,k} = t_{2,k} + v_{2,k}; \quad t_{4,k} = t_{3,k} + v_{3,k} \\ &t_{6,k} = t_{5,k} + v_{5,k}; \quad t_{7,k} = t_{6,k} + v_{6,k}; \quad t_{8,k} = t_{7,k} + v_{7,k} \end{aligned} \qquad \forall k \qquad [0]$$

$$t_{1,k} = t_{5,k}; \;\; t_{7,k} = t_{3,k}; \;\; t_{6,k} = t_{2,k} \qquad \forall k \qquad [0]$$

$$t_{4,k} + v_{4,k} = kC \qquad \forall k \qquad [0]$$

$$v_{jk} = g_{jk} + Y + R \qquad \forall j, \forall k \qquad [0]$$

$$g_{jk} \geq g_{jk,\min} \qquad \forall j, \forall k \qquad [0]$$

$$t_{jk}, g_{jk}, v_{jk} \geq 0 \qquad \forall j, \forall k \qquad [0]$$

The formulation explicitly models the ring-barrier control structure that is widely used in North America. Constraint [0] defines the timelines and sequences of all the phases in both rings. Constraint [0] indicates which phases are serving as barriers. Constraint [0] defines the end time of each cycle within the planning horizon. The minimum green requirement is defined in [0]. Note that although no maximum green constraint is enforced, the program will not increase the green of any phase indefinitely because of the constraint [0] and the precedence relationship. The phasing sequence may be changed by redefining the precedence relationship in constraints [0]–[0]. It is also possible to enable the optimizations of phasing sequence by adding indicators variables. But it is not in the scope of this research.

$$g_{jk} \geq \frac{V_j C}{S_j X_C} \qquad \forall j, \forall k \qquad [0]$$

Given a maximum degree of saturation, X_c, the minimum green time of each phase can be easily computed for all the look ahead cycles assuming stable traffic flow in a short period.

Constraints [0] dictate the minimum allowed green time for a phase restricted by the maximum allowable degree of saturation.

$$\begin{aligned} y_{jk} &\geq g'_{jk} - g_{jk} \\ y_{jk} &\geq 0 \end{aligned} \qquad \forall j, \forall k \qquad [0]$$

Constraint [0] defines the deviations of the new green times g_{jk} from the optimal background green times g'_{jk}. The two inequalities effectively linearizes $y_{jk} = \max\{g'_{jk} - g_{jk}, 0\}$, which implies only the positive deviations are penalized. Any expansion of g_{jk} from g'_{jk} has no direct cost to the objective function. However, due to precedence relationship, it would compress the conflicting phases that incur penalties.

3.2.3. Second-Stage Model

3.2.3.1. Objective Function

The function inside the expectation of [0] is the so called recourse function in stochastic programming term. For a given, $\mathbf{t}, \mathbf{v}$ and a number of random events $\omega \in \Omega$, the function is deterministically computable. With a well-defined probability space (Ω, F, P), the expectation is can be evaluated by $E(Q) = \sum_{\omega \in \Omega} p(\omega) Q(\omega)$. Therefore, for a given discrete random event, ω, the second stage recourse function of a classical two-stage stochastic program with fixed recourse can be formulated as the following:

$$Q(\mathbf{t}, \mathbf{v}, BR(\omega)) := \min \sum_{j \in J} o_j d_j \qquad [0]$$

$BR(\omega)$ represents a realized bus arrival time out of all the possible arrival scenarios in Ω. For notational convenience (ω) is omitted from further discussions. d_j denotes the delay to the priority request placed on phase *j*, which is a function of the bus arrival time and current signal timings. The weight, o_{jn}, of the priority delay determines the level of priority that a bus should receive. The bus priority weights can be estimated via different measures of importance that the system designer deemed applicable to the problem at hand. Normally, the priority can be formulated as a function of the bus passenger loads or bus schedule lateness.

3.2.3.2. Constraints

The constraints in the second stage mostly concerns with the computations of bus priority delay using the timing variables from the first stage and the random arrival parameters.

$$BR_j \geq t_{j,k-1} + g_{j,k-1} - (1-\theta_{jk})M \qquad \forall k \in K \setminus \{1\}, \forall j \qquad [0]$$

$$BR_j \leq t_{jk} + g_{jk} + (1-\theta_{jk})M \qquad \forall k, \forall j \qquad [0]$$

$$\sum_{k \in K} \theta_{jk} = 1 \qquad \forall j \qquad [0]$$

$$\theta_{jk} = \{0,1\} \qquad \forall j, \forall k \qquad [0]$$

where θ_{jk} is a binary variable identifies which phase and cycle the bus will be served. For under-saturated conditions, the bus arriving after end of phase *j* of cycle *k*-1 (i.e., inequality [0]) and before the end of phase *j* of cycle *k* (i.e., inequality [0]) at cycle *k*. For all other cycles, θ_{jk} are zeros. M is a large constant that can be set as the end time of the planning horizon (i.e., |*K*|*C*).

Assuming no delays caused by vehicle queues dissipating before the bus, the delay to the bus is simply $d_j = \max\{t_{jk} - BR_j, 0\}$ if the bus is to be served at phase *j* of cycle *k* (i.e., $\theta_{jk} = 1$), which can be equivalently expressed as:

$$d_j \geq t_{jk} - BR_j - (1-\theta_{jk})M \qquad \forall j, \forall k \qquad [0]$$

$$d_j \geq 0 \qquad \forall j \qquad [0]$$

The formulation to compute priority delay via constraints [0]–[0] is for a single bus. However, it can be easily extended to a multiple-bus scenario by adding a separate set of these constraints for each additional bus to be considered in the optimization.

3.3. PRELIMINARY PROOF-OF-CONCEPT EXPERIMENTS

One objective of conducting the proof-of-concept experiments is to show the feasibility of the stochastic model that uses green deviation as a way to proxy the impacts to the traffic on the conflicting phases. Another objective is to demonstrate some features of the math model by focusing on specific aspects of the model. Six offline proof-of-concept experiments were performed, each of which demonstrated a specific feature of the model. All experiments were conducted based on the signal timing and volume input in Table 7. The splits in the background

timing were optimized using a commercially popular offline signal optimization software called SYNCHRO.

Table 2: Background Timing for Proof-of-Concept Experiments.

Background Timing 1: Natural Cycle Length =90 sec								
Phase	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$	$\phi 5$	$\phi 6$	$\phi 7$	$\phi 8$
# of lanes	1	2	1	2	1	2	1	2
Volume	150	820	130	540	100	1350	150	250
v/c	0.55	0.71	0.80	0.83	0.79	0.94	0.83	0.40
Optimized splits	19	36	13	22	11	44	14	21
Approach Delay		29.3		42.8		31.9		38.7
Intersection Delay	34.1							
Background Timing 2: Extended Cycle Length = 120 sec								
Volume	150	820	130	540	100	1350	150	250
v/c	0.58	0.65	0.73	0.80	0.74	0.90	0.58	0.48
Optimized splits	23	51	17	29	14	60	23	23
Approach Delay		33.7		46.3		33.6		48.5
Intersection Delay	37.7							
Background Timing 3: Extended Cycle Length = 120 sec								
Volume	150	820	130	540	100	1350	80	410
v/c	0.58	0.65	0.74	0.51	0.74	0.90	0.27	0.94
Non-optimized splits	23	51	12	34	14	60	26	20
Approach Delay		33.7		38.4		33.6		68.9
Intersection Delay	39.4							

3.3.1. Experiment 1: Deterministic Arrival – Traffic Volume

This experiment showed that the math program is sensitive to traffic volumes on different phases, and the priority is given only to the degree that it is not changing the background timing too much nor inflicting too much delay surge.

- Scenario 1 (SS1_1): bus arrival time $BR_6 = 40$ and background timing 1 is assumed.
- Scenario 2 (SS1_2): bus arrival time $BR_6 = 52$ and background timing 2 is assumed.
- Scenario 3 (SS1_3): bus arrival time $BR_6 = 45$ and background timing 3 is assumed.

Table 3: Background Timing for Proof-of-Concept Experiments.

	Phase	ϕ1	ϕ2	ϕ3	ϕ4	ϕ5	ϕ6	ϕ7	ϕ8
SS1_1	v/c	0.55	0.67	0.90	0.88	0.92	0.88	0.92	0.43
	Optimized splits	19	38	12	21	10	47	13	20
	Green time change	0	+2	-1	-1	-1	+3	-1	-1
	Approach Delay		31.2		50.6		24.5		46.3
	Intersection Delay	33.8							
	Bus Delay	3							
SS1_2	v/c	0.58	0.59	0.95	0.87	0.95	0.87	0.65	0.57
	New splits	23	56	14	27	14	27	21	20
	Green time change	0	+5	-3	-2	-3	-2	-2	-3
	Approach Delay		34.0		51.4		51.4		63.8
	Intersection Delay	37.2							
	Bus Delay	1							
SS1_3	v/c	0.58	0.56	0.98	0.61	0.92	0.77	0.39	0.94
	New splits	23	58	10	29	12	69	19	20
	Green time change	0	+7	-2	-5	-2	+9	-7	0
	Approach Delay		32.6		43.0		24.4		77.4
	Intersection Delay	37.0							
	Bus Delay	6							

For scenario SS1_1, all the changes of phase splits are at cycle 1. As a result of the arrival of bus at time 40 requesting for phase 6, the start time of phase 6 green is brought earlier by 3 seconds. The 3 seconds are distributed to phases 7, 8, and 5. One thing to point out that the program can allocate extra green time to non-transit phases according to their degree of saturation. For example, both phase 1 and 2 have benefited from the extra green time as a result of the moving of the barrier to 2 seconds earlier; however, since phase 2 is originally more saturated than phase 1, so phase 2 is getting all the 2 seconds of extra green time while phase 1 is not getting any green time.

Scenario SS1_2 is also similar to SS1_1, but in this case there are more extra green times that can be taken away from non-transit phases to provide the priority. So a total of 7 seconds are extracted from phases 7, 8, and 5 to bring phase 6 earlier.

Scenario SS1_3 uses a background timing where phases 1, 2, 5, and 6 are non-optimized to show how different background degrees of saturation affect the phase green time reduction. Notice that phases 3 and 8 are more saturated than phases 4 and 7, respectively. Therefore, when 7 seconds of green time are taken from both rings, the formulation was able to recognize the difference and take more green time from the less saturated phases, 4 and 7.

3.3.2. Experiment 2: Deterministic Arrival – Bus Priority Level

This experiment showed that as the priority level increases the bus delay decreases and passenger car delay increases. The minimum green and saturation flow constraints are removed.

- Scenario 1 (SS2_1): bus arrives at phase 2 at time $BR_2 = 0$. Background timing 2 is assumed. Weight on bus delay ranges are 1, 2, 5, 10, and 15.
- Scenario 2 (SS2_2): one bus arrives at phase 1 at time $BR_1 = 99$, and another bus arrives at phase 2 at time $BR_2 = 39$. Background timing 2 is assumed. The ratio of priority on the first bus (BR_1) to the priority on the second bus (BR_2) ranges from 1:10, 5:10, 10:10, 10:5, and 10:1.

Table 4: Timing Changes and Resulting Bus Delays.

	Delay weights	1	2	5	10	15
SS2_1	Changes in $\phi 1$	-6.7	-13.5	-19	-19	-19
	Changes in $\phi 3$	-2.9	-5	-13	-13	-13
	Changes in $\phi 4$	-2.1	-3.7	-11	-23.7	-25
	Bus Delay	57.2	45.4	26	13.3	12
	Delay weight ratios	1:10	1:2	1:1	2:1	10:1
SS2_2	Changes in $\phi 1$	-17.2	-13.5	-6.7	-6.7	+34
	Changes in $\phi 2$	+26	+19.5	+7.8	+7.8	-34
	Changes in $\phi 3$	-7.4	-5.7	-2.9	-2.9	0
	Changes in $\phi 4$	-5.4	-4.2	-2.1	-2.1	0
	Bus 1 (BR_1) Delay	62	62	62	56.9	0
	Bus 2 (BR_2) Delay	0	6.4	18.2	18.2	64

From SS2_1, as this weight increases, the changes of non-transit phases increase and bus delay decreases. The empirical analysis above shows that the weight for a single bus ranges from 0 to 10, with 10 means the highest priority. The relationship between the priority level and bus occupancy level remain to be determined. From SS2_2, it shows two things: (a) the program can handle multiple bus arrivals, and (b) the weights on bus delays can be used to explicitly control the priority level of each bus.

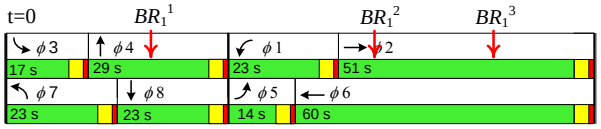
3.3.3. Experiment 3: Deterministic Arrival – Arrival Times

This experiment showed that the bus actual arrivals impact the timing plan.

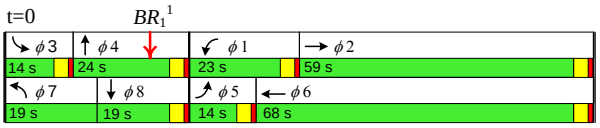
- Scenario 1 (SS3_1): $BR_1 = 30$ on with priority 10. Background timing 2 is assumed. Early green on phase 1 is expected.
- Scenario 2 (SS3_2): $BR_1 = 75$. Background timing 2 is assumed. Green extension on phase 1 is expected.
- Scenario 3 (SS3_3): $BR_1 = 100$. Background timing 2 is assumed. No change is expected in the first cycle, but early green on phase 1 in the second cycle is expected.

Table 5: Optimized Timing for Different Arrival Scenarios.

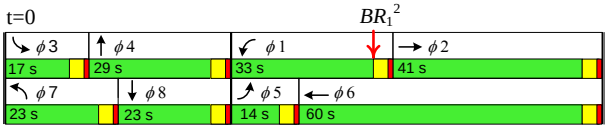
	Phase	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$	$\phi 5$	$\phi 6$	$\phi 7$	$\phi 8$
SS3_1	v/c	0.58	0.55	0.95	1.0	0.74	0.79	0.74	0.60
	Optimized splits	23	59	14	24	14	68	19	19
	Green time change	0	+8	-3	-5	0	+8	-4	-4
	Approach Delay		31.2		50.6		24.5		46.3
	Intersection Delay	39.5							
	Bus Delay	8							
SS3_2	v/c	0.38	0.82	0.73	0.80	0.74	0.90	0.58	0.48
	Optimized splits	33	41	17	29	14	60	23	23
	Green time change	+10	-10	0	0	0	0	0	0
	Approach Delay		42.1		46.3		32.7		48.5
	Intersection Delay	39.6							
	Bus Delay	0							
	Phase	Cycle2 $\phi 1$ ~ $\phi 8$							
SS3_3	v/c	0.58	0.55	0.95	1.0	0.74	0.79	0.74	0.60
	Optimized splits	23	59	14	24	14	68	19	19
	Green time change	0	+8	-3	-5	0	+8	-4	-4
	Approach Delay		31.2		50.6		24.5		46.3
	Intersection Delay	39.5							
	Bus Delay	58							



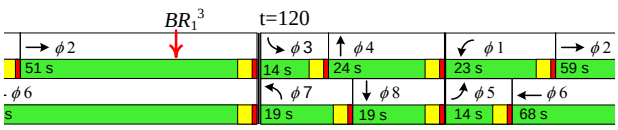
(a) Background timing 2



(b) Early Green on Phase 1 (SS3_1)



(c) Green Extension on Phase 1 (SS3_2)



(d) Early Green on Next Cycle (SS3_3)

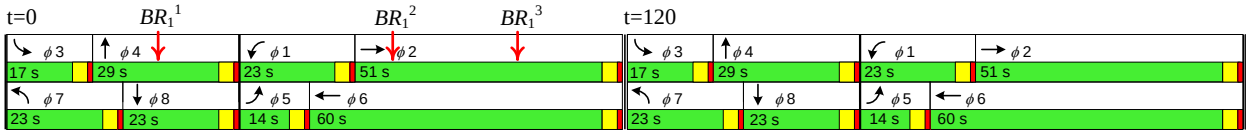
Figure 1: Formulation Behavior under Different Arrival Scenarios.

As the early green is provided to its max potential in SS3_1, the green duration of transit phase 1 is not changed; this is because phase 2 is more saturated than phase 1, so extra green time is given to phase 2 while providing bus priority to phase 1.

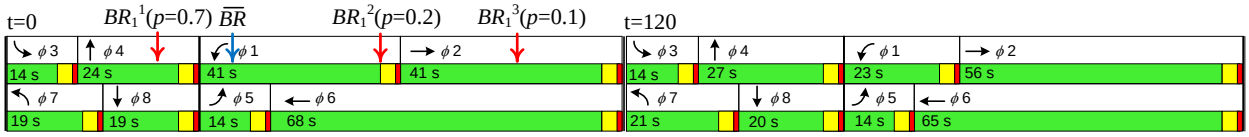
3.3.4. Experiment 4: Uncertain Arrival

This experiment showed that depending what the distribution looks like, the program would consider that but will not change as much as the deterministic case.

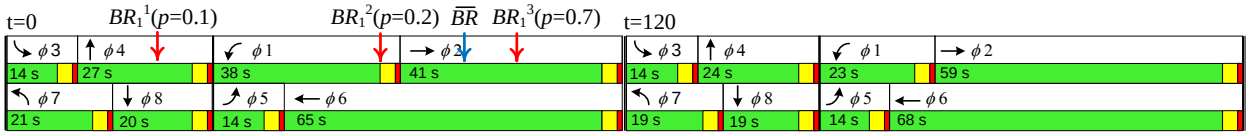
- Scenario 1 (SS4_1): BR_1 = (30, 75, 100) with p = (0.7, 0.2, 0.1), with priority level 10, mean arrival time = 46.
- Scenario 2 (SS4_2): BR_1 = (30, 75, 100) with p = (0.1, 0.2, 0.7), with priority level 10, mean arrival time = 88.
- Scenario 3 (SS4_3): BR_1 = (30, 75, 100) with p = (0.1, 0.2, 0.7), with priority level 2, mean arrival time = 88.



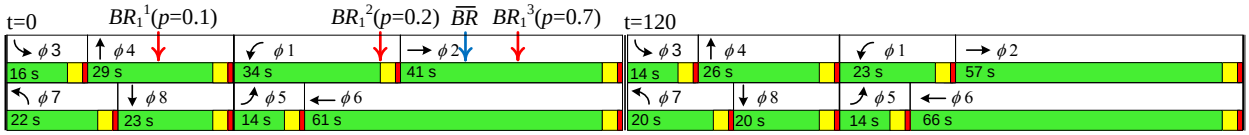
(a) Background timing 2



(b) Random arrival pattern 1 with priority 10 (SS4_1)



(c) Random arrival pattern 2 with priority 10 (SS4_2)



(d) Random arrival pattern 2 with priority 2 (SS4_3)

Figure 2: Optimized Timing with Uncertain Arrival Times.

There are a few points to make here:

- All three scenarios have moved the end of phase 1 green time to be equal to the second possible bus arrival time ($BR_1^2 = 75$). This shows that the mathematical program will strive to accommodate priority whenever possible. But it will not waste any second of green time to a call that cannot be accommodated ($BR_1^3 = 100$), even if the call is very likely.
- SS4_1 has the longest green time for phase 1, while SS4_3 has the shortest. This is because that bus arrival in SS4_1 has a 70 percent chance of arriving at time 30 versus a 10 percent chance in SS4_3. The math program determines that bringing the start of phase 1 green early for SS4_1 is more beneficial than doing the same for SS4_3.
- If one does not use the complete information of bus arrival time, but instead using only the mean arrival time, the expected bus delay may be much higher. The following two computations prove this possibility:

$$(46-30)\times 0.7 + (166-75)\times 0.2 + (166-100)\times 0.1 = 36 \qquad [0]$$

$$(38-30)\times 0.7 + \text{Max}\{38-75, 0\}\times 0.2 + (161-100)\times 0.1 = 11.7 \qquad [0]$$

- Comparing SS4_2 and SS4_3, one sees the effect of priority on phase time selection.

3.3.5. Experiment 5: Deterministic Arrival – Passage Interval

This experiment showed the formulation can accommodate an interval of bus arrival with minimal formulation change. The interval is to account for the bus passage time, *W*, at an intersection. Some buses run slower and need longer time to pass through intersections; the green extension should give more time to these buses. Let the arrival time of the bus be the $\underline{BR}_{jn}$ and requires *W* to pass through the intersection, so the leave time is $\overline{BR}_{jn} = W + \underline{BR}_{jn}$, and the constraints [0] and [0] are changed into:

$$\overline{BR}^s_{jn} \le t_{jk} + g_{jk} + (1-\theta^s_{jkn})M \qquad \forall k, \forall j, \forall n, \forall s \qquad [0]$$

$$\overline{BR}^s_{jn} \ge t_{j,k-1} + g_{j,k-1} - (1-\theta^s_{jkn})M \qquad \forall k \setminus \{1\}, \forall j, \forall n, \forall s \qquad [0]$$

And the delay calculation has to refer to the arrival time of the bus, so [0] changed into:

$$d^s_{jn} \ge t_{jk} - \underline{BR}^s_{jn} - (1-\theta^s_{jkn})M \qquad \forall j, \forall k, \forall n, \forall s \qquad [0]$$

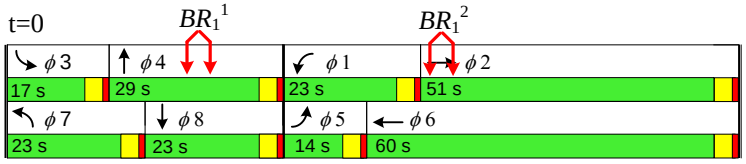
To ensure that the green time of the service phase has enough time for a bus to pass through the intersection, a constraint can be added:

$$g_{jk} \geq W - (1 - \theta^{s}_{jkn})M \qquad \forall j, \forall k, \forall n, \forall s \qquad [0]$$

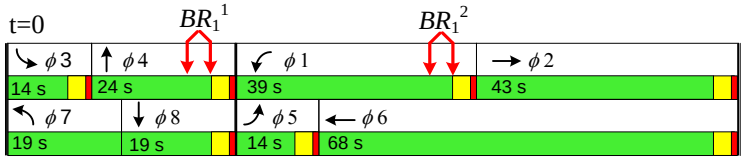
Since W is modeled here as the passage time, and is typically small comparing to the phase green time, practically this constraint is not the binding constraint. Constraint [0] is considered as redundant in this experiment.

Table 6: Timing Changes and Bus Delay by Setting Passage Interval.

Phase Timing	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$	$\phi 5$	$\phi 6$	$\phi 7$	$\phi 8$
Optimized splits	39	43	14	24	14	68	19	19
Green time change	+16	-8	-3	-5	0	+8	-4	-4
Bus	BR_3^1				BR_3^2			
Passage Interval	30 - 33				70 - 73			
Delay	8				0			



(a) Background timing 2



(b) Bus arrival with interval passage time

Figure 3: Optimized Timing with Passage Interval.

With minor modification, the formulation can ensure the passage of the bus upon arrival by extending the green interval for the required passage time as in BR_1^2. In the case that accommodation is not possible (i.e., BR_1^1), the bus delay is correctly calculated in reference to the beginning of the passage interval.

3.3.6. Experiment 6: Start Time of the Optimization

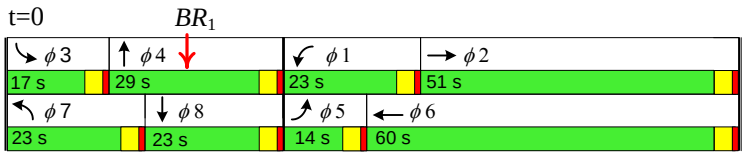
The original formulation assumes all priority requests are known before the onset of a cycle. In fact, a bus can send a priority request as soon as it comes into range of the detection area. In the connected vehicle case, this range can extend up to 1000 feet. There may be enough lead time before the bus arrives at the intersection and the above assumption is reasonable. However, if the detection area is relatively close to the intersection, immediate priority needs to be given instead of waiting for the new start of a cycle. This consideration is particularly important for real-time control purposes. To accommodate real-time need, two sets of constraints can be added:

$$g_{jk} = g_{jk}^{past} \qquad k = 1, \forall j \in J^{past} \qquad [0]$$

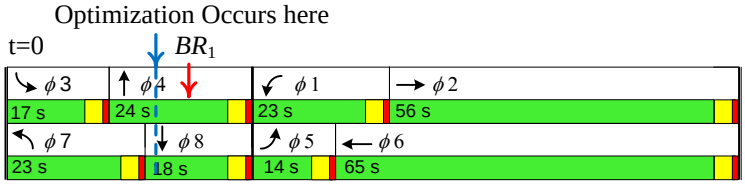
$$g_{jk} \geq g_{jk}^{current} \qquad k = 1, \forall j \in J^{current} \qquad [0]$$

where J^{past} and $J^{current}$ are the set of phases that are already elapsed and ongoing, respectively; g_{jk}^{past} and $g_{jk}^{current}$ are the green times of phase *j*s that are elapsed in the past phase and current phase, respectively. It can be easily seen that constraint [0] set the green time of the past phases to the exact green time that just went by for these phases, while constraint [0] ensures the current phases have a green time no smaller than the green times that just elapsed for current phases. The following experiment shows this concept.

Assume a bus request is received at time t = 25, phases 3 and 7 are already over with the background green time, and phases 4 and 8 are the current phase. The bus is requesting for phase 1, and the arrival time is determined to be BR_1=30.



(a) Background timing 2



(b) Optimization occurred not from the beginning of the cycle

Figure 4: Optimized Timings by Optimization Conducted during the Cycle.

The optimization has taken place at t = 25 as soon as the bus priority request is received. The priority is given to the bus by bringing phase 1 forward to start at t = 41, without changing the green times of the past phases 3 and 7.

4. ENHANCED MATHEMATICAL MODEL

The proof-of-concept experiments showed the behaviors and features of the basic model. The preliminary tests also showed a very limited feasibility region when we strictly enforce certain constraints. In actual implementation of the model, handling multiple conflicting bus lines is crucial for developing a real-time adaptive TSP system. To enhance the model, the section 4.1 explores variations of the basic formulation to achieve higher flexibility that can yield higher benefits. Section 4.2 develops a computation procedure to capture the delays of the bus incurred by the vehicle queues and how such delay computation is incorporated into the formulation. The effects of both the linear and the nonlinear assumptions on bus trajectories are discussed. Section 4.3 describes a rolling horizon optimization scheme that is critical for continuous online implementation of TSP control system.

4.1. FIRST STAGE FORMULATION ENHANCEMENT

Three components in the first stage formulation are modified: (a) the computation of weight for green time deviation, (b) the cycle length constraint, and (c) the degree of saturation constraint. The change made (a) is to improve the program's ability to differentiate phases with high degree of saturation from low degree of saturation, (b) allows variable cycle length during the planning horizon, and (c) restricts the maximum total degree of saturation over several cycles.

4.1.1. Calculating Weight for Deviations

The first stage objective function controls the balance between the phase green time deviations and the bus delay. For each phase, the weight *c* determines the distribution of the deviations among all the phases. It is reasonable that a phase shall be panelized higher if it has already suffered from congestion than the one that is relatively less saturated. Researchers compared four different ways to compute the weights.

Option 1: $c_{jk} = \frac{X_{jk}}{\sum_{j \in J} X_{jk}}$ (Weight -1)

Option 2: $c_{jk} = \frac{X_{jk}^{\ p}}{\sum_{j \in J} X_{jk}^{\ p}}$, set $p = |J|$ (Weight -2)

Option 3: $c_{jk} = \dfrac{1/(1-X_{jk})}{\sum_{j\in J}\sum_{k\in K} 1/(1-X_{jk})}$ (Weight-3)

Option 4: $c_j = \dfrac{1/(1-X_j)}{\sum_{j\in J} 1/(1-X_j)}$ where $X_j = \dfrac{V_j \sum_{k\in K} C_k}{S_j \sum_{k\in K} g_{jk}}$ (Weight -4)

First, each weight is normalized by the sum of all weights. The normalization ensures the weights only dictate the relative importance among different phases, not the relative importance between the total phase deviations and the bus delay. The changes made here only affects the ways the program distribute the total deviations that are needed to reduce certain amount of the bus delay.

Option 1 and option 2 base the importance of each phase directly on the values of the degree of saturations. Specifically, option 2 makes the linear proportionality nonlinear in order to magnify the significance of higher X_{jk} values. The polynomial order used in option 2 is set as equal to the number of phases in consideration.

Option 3 and option 4 base the importance of each phase on the reciprocal of the remaining under-saturation, which is defined as 1-*X*. One can easily see that option 3 is problematic if the degree of saturation is equal to or larger than 1. Option 4 rectifies the problem by computing the degree of saturation over the entire planning horizon. That means, if the underlying prevailing traffic condition is under-saturated, the optimization program will ensure under-saturation after the end of the planning horizon, and it does allow temporary oversaturation; see section 4.1.2. Option 4 requires some modifications of the original objective function:

Minimize: $\sum_{j\in J} c_j y_j^2 + E[Q(\boldsymbol{t}, \boldsymbol{v}, \overline{BR})]$ [0]

It subjects to one additional constraint for each phase in a cycle as follows:

$$y_j = \sum_{k\in K} y_{jk} \quad [0]$$

Figure 5 shows a comparison of the performance of all four weight formulation options under the same network and traffic condition setups. In general, their ability to give priority to buses under various traffic conditions are very comparable. However, Weight-3 and Weight-4 seems to give the lowest impact on general traffic under low to medium degree of saturation levels, while Weight-1 and Weight-4 are less disruptive to general traffic on the high degree of

saturation level. Weight-4 appears to be the most robust because it consistently performs above average to the best over all traffic conditions.

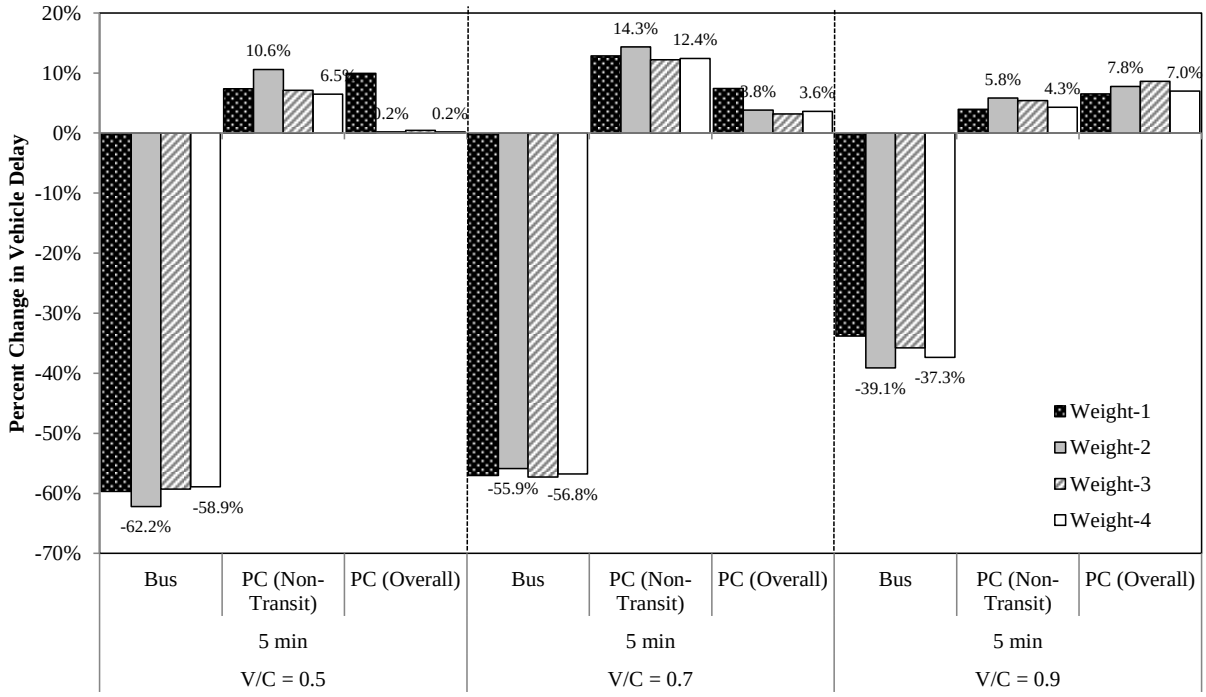


Figure 5: Comparisons of Weight Formulations.

A constant value can be applied to the weights of each phase. This constant value (e.g., the number of lanes) is tied to the significance of the phase. In this test run, researchers did not apply any constant value. The importance of the phase is completely determined by the degree of saturation. That is, if two seconds of extensions are needed by the bus phase, the conflicting phases with the same degree of saturation on the same ring will shorten their green time by equal amount.

4.1.2. Allowing Temporary Oversaturation

It adds more flexibility to timing adjustment if any of the signal phases are allowed to go temporarily oversaturated in one cycle. But the precaution is that all phases shall be kept under-saturated for the entire planning horizon. Mathematically, all phases have to be restricted by the following constraint:

$$\frac{\sum_{k \in K} V_{jk} C_k}{\sum_{k \in K} S_j g_{jk}} \leq X_C \qquad \forall j \in J \qquad [0]$$

and dropping the degree of saturation constraint for each cycle as defined in constraint [0]. This formulation may provide more flexibility in adjusting the timing in favor of the transit bus, but the resulting oversaturation may have undefined behavior. One of the most infamous consequences is left-spillback or blockage (*Yin et al. 2010*). If this is a concern, dropping the summations on both sides of the constraint will guarantee no oversaturated conditions in any phase of any look ahead cycles.

In addition, temporary oversaturation makes the computations for weight options 1, 2, and 3 invalid because the degree of saturation of these options are defined separately for each cycle and the behavior for oversaturation in one cycle is undefined. Option 4 defines an overall degree of saturation for the phase over the entire planning period. When temporary oversaturation is allowed for one cycle, constraint [0] guarantees the sum of green times of phase *j* for the rest of the cycles is large enough to ensure overall under-saturation. In a rolling optimization scheme, as described in section 4.3, the past demand and the past capacity have to be recorded during the implementation of a previous optimization session. Additionally, it is also necessary to compute the residual queue length of the phase to better estimate the bus queue delays, as described in section 4.2, that are caused by the blocking queue.

The flexibilities for timing adjustment gained from allowing temporary oversaturation bring about one major problem—additional delay to vehicles that have to wait one more cycle. In the formulation that does not directly compute the vehicle delays, the additional delay for a vehicle to wait one more cycle due to oversaturation is impossible to capture. That is to say, the degree of saturation may not be a good proxy for vehicle delay when it is allowed to be oversaturated for a few cycles.

4.1.3. Variable Cycle Length and Fixed Planning Horizon

Allowing variable cycle lengths within the planning horizon gives much greater flexibilities in terms of adjusting signal timings to accommodate priority needs. Figure 6 illustrates the difference in cycle length between the background cycle and the cycle after an optimization session is conducted, given a two-cycle planning horizon.

The design principle is to allow individual cycle length to be different from the background operating cycle length, but to not allow the end time of the planning horizon to change. For example, cycle lengths of cycles 1 and 2 are different from the optimal background cycle, but both cycles have to come back to the expected end time of cycle 2. In doing this, cycle 3 is not affected by the optimization and is running on the optimal cycle. The same principle shall apply if there are more cycles used in the planning horizon. This design principle essentially confines the changes done for the priority request within a certain time period.

It may be also possible to dynamically adjust the planning horizon based on the bus arrival pattern. However, it is not within the scope of the research to analyze the impact of variable planning horizon. This research only selects a fixed planning horizon in all the analyses.

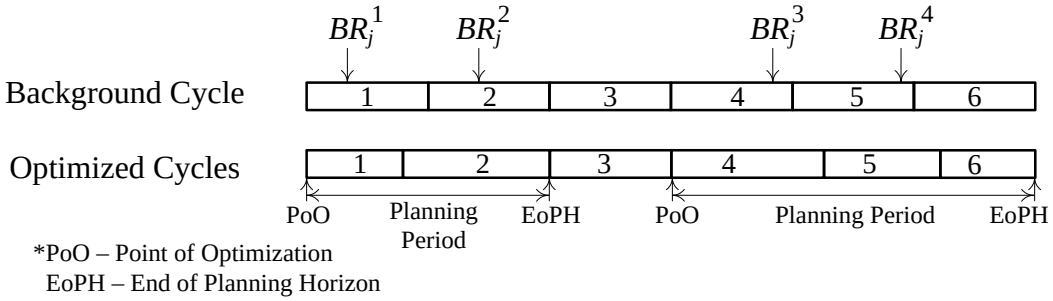


Figure 6: Variable Cycle Length Implementation.

4.2. COMPUTATION OF QUEUE DELAY

As argued in the introduction, one of the two keys for successful implementation of a bus signal priority system is early planning. A reliable early planning requires early detection and an accurate projection of the bus arriving at the stop bar—the time when a bus clears the intersection. It is not unlikely that a bus is blocked by the queue waiting for a green. Without the present of a bus stop, it may be possible to quickly change the light and flush out all the buses before the bus arrives. However a bus still experiences some delay if the queue is not flushed quick enough. Making matters worse, if a bus stop is present, it makes the process even more complicated.

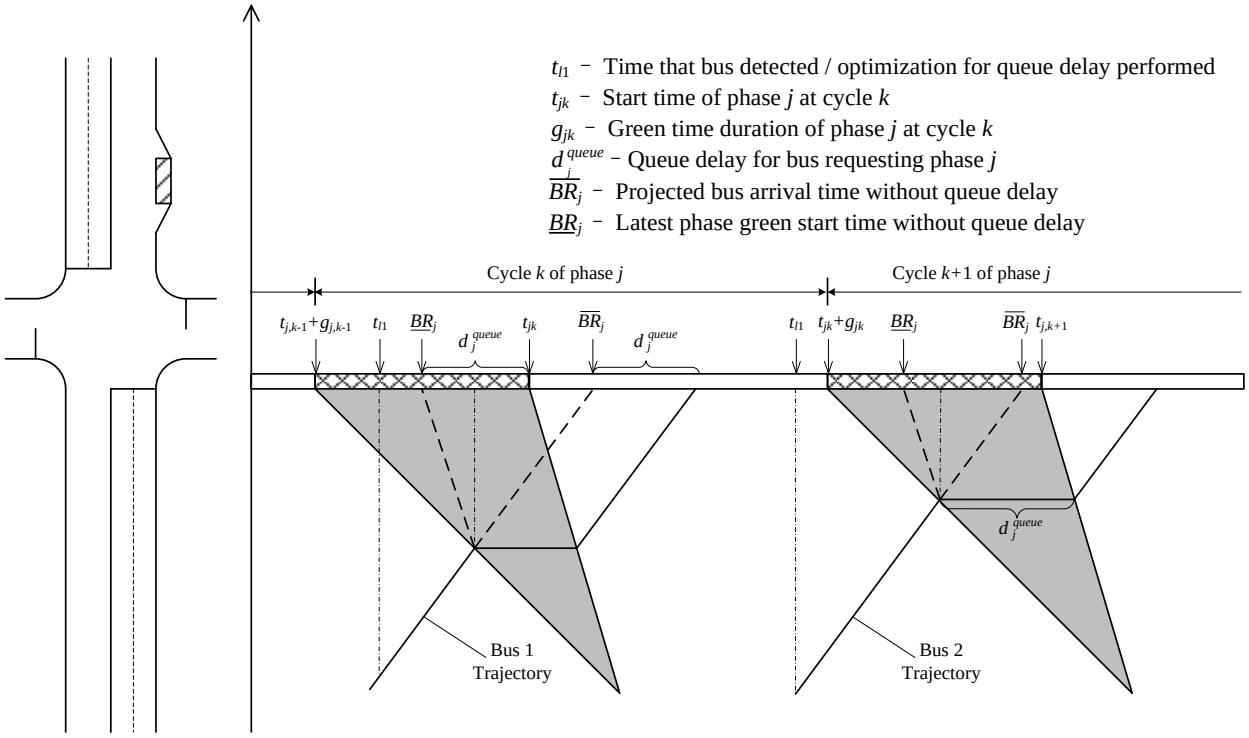
Furth and SanClemente (*2006*) discussed the impacts of various factors on the bus arrival time, including bus stops. For the purposes of this research, the researchers expanded uponthe interactions in more detail in the concept of mathematical formulations. Both far-side bus stop

configurations and near-side bus stop configurations are considered in this implementation. The following assumptions are made:

- Flow rates are known, and the arrival pattern is relatively stable, in order to make good projections of queue lengths.
- The prevailing traffic condition is under-saturated. Although this assumption may be relaxed thanks to the capability of the formulation, it requires a user to define a higher number for look-ahead cycles when optimize, this would add higher complexity to the optimization routine.
- No interactions between two buses at bus stops. Meaning a preceding bus cannot block the entry of a following bus to the bus stop.
- Buses that need to request for signal priority have to have an onboard unit (OBU) that is capable to collect its position and instantaneous speed data, but the precision requirement is not high.

4.2.1. Far-Side Bus Stop Configuration

Figure 7 depicts two similar scenarios for the far-side bus stop configuration with the difference that the bus is detected during different time point in a cycle. Case (a) is that the bus is detected while the phase is red, and case (b) is when the phase is green. First, notice for both scenarios that because of the far-side bus stop, the bus on current link will approach the stop bar with only one possible source of delay, the queue delay. The actual bus arrival time is a function of the projected arrival time and the queue delay. So the actual bus arrival time is used in the mathematical formulation to determine when the green time of phase *j* at cycle *k* shall end. Second, the queue delay is in turn a function of when the green time of phase *j* starts.



(a) Bus 1 detected when phase is red (b) Bus 2 detected when phase is green

Figure 7: Projected and Actual Bus Trajectories for Far-Side Bus Stop Configuration.

Let the projected arrival time of the bus under free flow conditions be the $\overline{BR}_j$, so the bus can be served at cycle *k* when the queue delay, d_j^{queue}, is not causing it to arrive later than the end time of phase *j* green of this cycle. Mathematically:

$$\overline{BR}_j + d_j^{queue} \le t_{jk} + g_{jk} + (1-\theta_{jk})M \qquad \forall k, \forall j \qquad [0]$$

$$\overline{BR}_j + d_j^{queue} \ge t_{j,k-1} + g_{j,k-1} - (1-\theta_{jk})M \qquad \forall k \setminus \{1\}, \forall j \qquad [0]$$

Let $\underline{BR}_j$ be the latest start time for phase *j* green that would not cause queue delay. From the graph, if the green time of phase *j* does not start before $\underline{BR}_j$, some queue delay will exist. Therefore, queue delay of phase *j* is $\max\{0, t_{jk} - \underline{BR}_j\}$, or can be equivalently defined using the following relationship:

$$d_j^{queue} \ge t_{jk} - \underline{BR}_j - (1-\theta_{jk})M \qquad \forall j, \forall k \qquad [0]$$

$$d_j^{queue} \geq 0 \qquad \forall j \qquad [0]$$

The above constraints apply to both case (a) and (b). But in case (b), this scenario allows the real-time strategy to start the red time of phase *j* earlier so that the green time on the phases that are conflicting with phase *j* is not shortened. Since the deterministic MINP formulation minimizes the deviations from current green durations, it will automatically attempt to bring the red time early so as not to impact other phases as much.

This configuration also represents a bus skipping a near-side bus stop. If information is available from either the bus OBU or the bus stop infrastructure that the bus will skip the approaching bus stop or simply no passengers need to be picked up, this model can be applied.

4.2.2. Near-Side Bus Stop Configuration

A near-side setup for a bus stop is more complicated in that a bus may encounter a queue before and/or after it stops for service at a bus stop. Hence, it is likely that a bus needs to stop as many as three times at an approach with near-side bus stop, even under unsaturated traffic conditions. This configuration is a generalized version of the far-side configuration, especially considering a bus can skip the bus stop entirely. Figure 8 illustrates such a case and all the components necessary in estimating the arrival time of the bus at the stop bar. Vehicle accelerations are not considered.

Let the projected arrival time of the bus under free flow conditions be $\overline{BR}_j$, which can be easily computed with the location of the bus and its running speed. Queue delay on cycle *k* for a bus that requests phase *j* is defined as d_{jk}. Notice that d_{jk} is a generalization for d_j^{queue} that is previously defined. If D_{dwell} is the dwell at the bus stop, then the actual bus arrival time is readily $BR_j = \overline{BR}_j + D_{dwell} + \sum_{i=1}^{K} d_{jk}$. Replace the arrival time of the original formulation (inequality [0] and [0]) to get:

$$\overline{BR}_j + D_{dwell} + \sum_{i=1}^{K} d_{ji} \leq t_{jk} + g_{jk} + (1-\theta_{jk})M \qquad \forall k, \forall j \qquad [0]$$

$$\overline{BR}_j + D_{dwell} + \sum_{i=1}^{K} d_{ji} \geq t_{j,k-1} + g_{j,k-1} - (1-\theta_{jk})M \qquad \forall k \setminus \{1\}, \forall j \qquad [0]$$

The implication of generalizing d_j^{queue} to a cycle-dependent variable d_{jk} is that it needs (*k*-1) constraints for one queue delay to be computed correctly. If *k* is large, it becomes a

computationally very difficult problem to solve. To see this, let $\underline{BR}_{jk}$ be the latest time to start phase *j* green of cycle *k* so that no queue delay on cycle *k* for the bus would exist. This means that except when *k* = 1, all $\underline{BR}_{jk}$ will be dependent on all d_{jk} from previous cycles. Mathematically, this is equivalent to computing delay as:

$$d_{j,k-r} \geq t_{j,k-r} - \underline{BR}_{j,k-r} - (1-\theta_{jk})M \qquad \forall j, \forall k \backslash \{1,...,r\}, \forall r \in \{0,..,K-1\} \qquad [0]$$

$$d_{j,k-r} \geq 0 \qquad \forall j, \forall k \backslash \{1,...,r\}, \forall r \in \{0,..,K-1\} \qquad [0]$$

$$d_j = \sum_{k=1}^{K} d_{jk} \qquad [0]$$

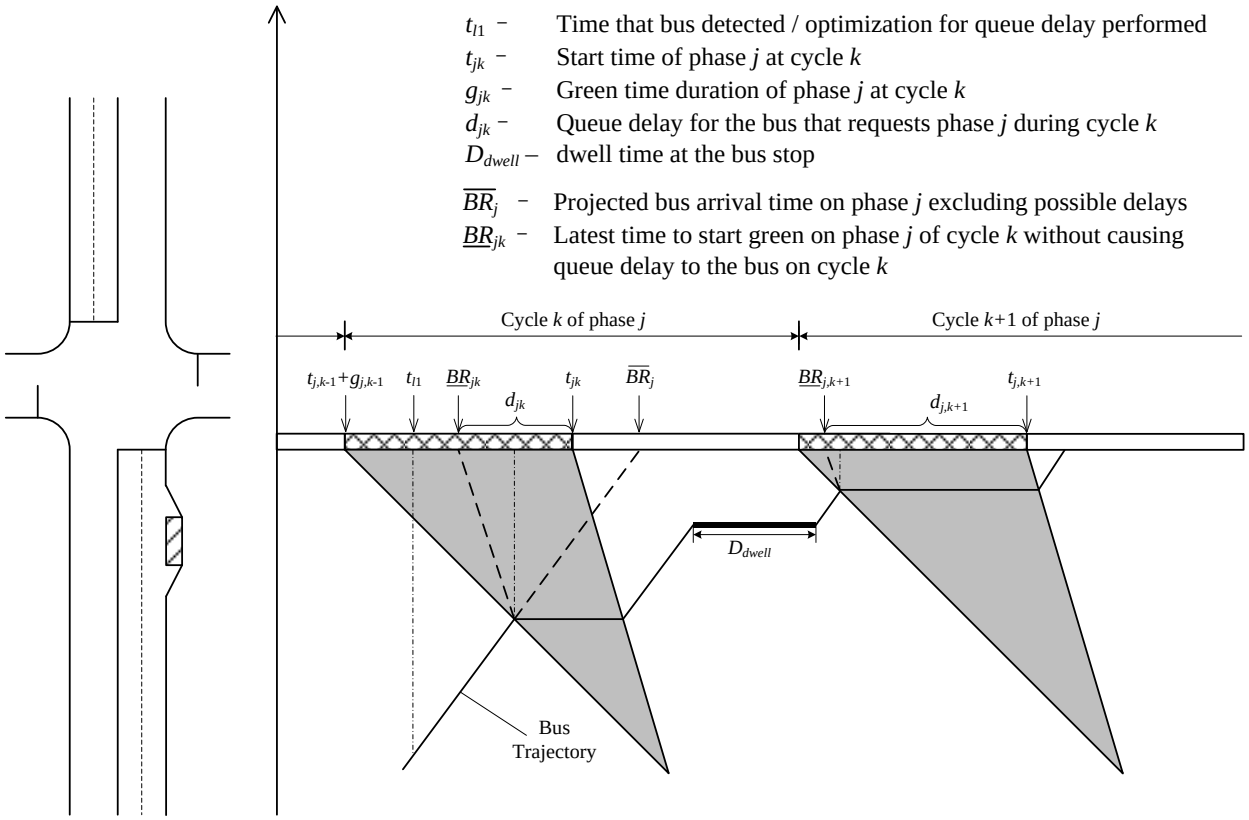


Figure 8: Projected and Actual Bus Trajectories for Near-Side Bus Stop Configuration.

Therefore, minimizing the overall bus delay due to queue, d_j, will result in minimal delay in all cycles. To compute $\underline{BR}_{jk}$, find the intersection of the bus trajectory and the expected end of queue trajectory in the time space diagram.

4.2.3. Computation of Queue Delays

To enable the estimation of queue delays at current and future cycles from the queuing diagram, the most critical time points to be computed are $\underline{BR}_{jk}$. To do this, first examine the scenarios of a bus trajectory when approaching an intersection stop bar. Figure 9 simplifies all possible cases of the interactions between a bus and queues over several cycles in a time-space queuing diagram. These cases are summarized as following:

- Case 1: the bus will meet the end of the queue before arriving at the bus stop or the intersection stop bar (e.g., bus No.1 trajectory in cycle *k* of phase *j*).
- Case 2: the bus arrives at a bus stop and dwells for a short duration then leaves the bus stop and joins the queue downstream (e.g., bus No. 1 trajectory after leaving the first queue it met in cycle *k* of phase *j*).
- Case 3: the bus arrives at a bus stop and dwells for a long duration that the queue backs up to the bus stop and the bus closes its door before the queue dissipates (e.g., bus No.2 trajectory after leaving the first queue it met in cycle *k+1* of phase *j*).
- Case 4: the bus arrives at a bus stop and dwells for a long duration that the queue backs up to the bus stop and the bus closes its door after the queue dissipates (e.g., bus No.3 trajectory).

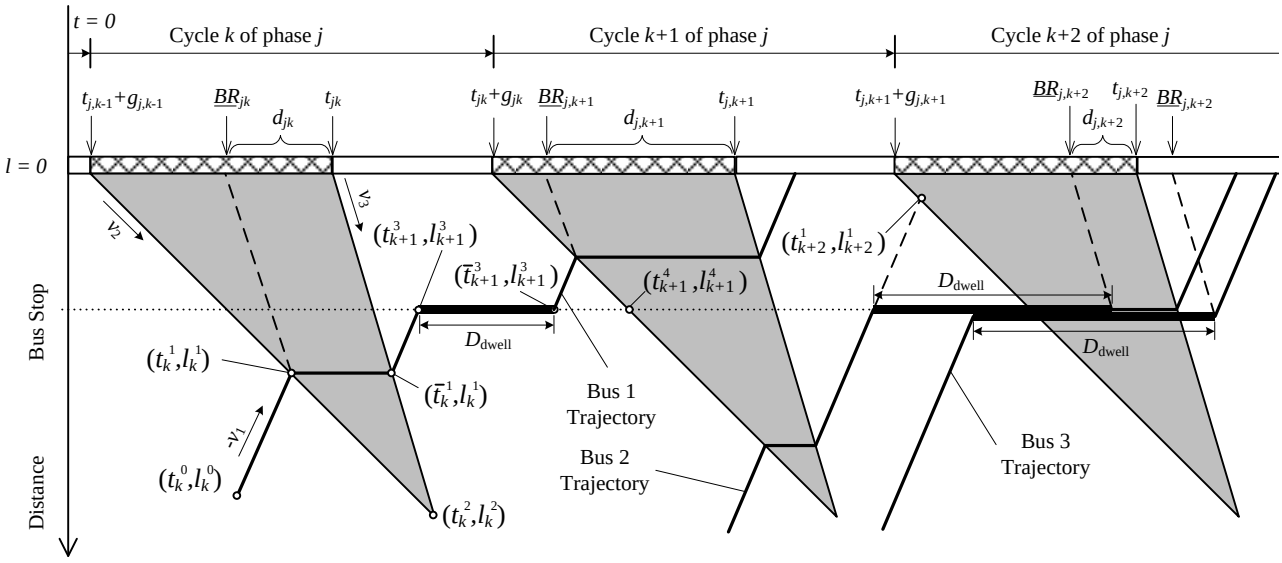


Figure 9: Critical Temporal and Spatial Pairs for the Computation of Queue Delays.

Cases 2 and 4 are variations of cases 1 and 3, respectively. Therefore, studying cases 1 and 3 are sufficient for capturing all possible scenarios. Figure 9 demonstrated the critical time points to give a reasonable estimate to the bus delay caused by queue. However, for computing these time points, it is necessary to make some simplifying assumptions as follows:

- The timings of the start (t_{jk}) and the end ($t_{jk} + g_{jk}$) of phase *j* are known for the current cycle as well as the immediately past and the next few cycles.
- The bus travels at the desired speed (v_1) as soon as it is not dwelling at a bus stop or within a blocking queue. Note, use -v_1 for all computations using bus speed.
- The speeds of queue forming (v_2) and dissipating (v_3) shockwaves are known and are relatively stable.

4.2.3.1. Computation Algorithm

To be more specific, a total of five critical time-space pairs need to be computed for every cycle of phase *j* in order to determine the case with which the bus is projected to encounter. These pairs are dubbed as follows: (t_k^0, l_k^0) denotes the initial state of bus at cycle *k*; (t_k^1, l_k^1) denotes the intersection of bus and queue trajectories; (t_k^2, l_k^2) denotes when and where the signal queue

is expected to dissipate completely; (t_k^3, l_k^3) denotes when bus will arrive at the bus stop, or stop bar if no bus stop downstream; and (t_k^4, l_k^4) denotes when the queue will back up to where the bus stop is, with l_k^4 always equals to the distance of the bus stop from stop bar I^{bus}.

Let *k* denote the cycle of phase *j* when the computation of bus trajectory is to be performed. It is convenient to set the most recent end time ($t_{j,k-1}+g_{j,k-1}$) of phase *j* in the past as time zero. Consider the trajectory of bus No. 1. Given the bus No.1 is detected at (t_k^0, l_k^0), it follows that:

$$t_k^1 = \frac{l_k^0 + v_1 t_k^0 + v_2(t_{j,k-1} + g_{j,k-1})}{v_1 + v_2} \quad \text{and} \quad l_k^1 = -v_1(t_k^1 - t_k^0) + l_k^0 \qquad [0]$$

$$t_k^2 = \frac{v_3 t_{jk} - v_2(t_{j,k-1} + g_{j,k-1})}{v_3 - v_2} \quad \text{and} \quad l_k^2 = v_3(t_k^2 - t_{jk}) \qquad [0]$$

By comparing t_k^1 and t_k^2, the bus is projected to be blocked by the queue for d_{jk} (case 1). It is then very easy to compute $\underline{BR}_{jk}$ and $\bar{t}_k^1$. Case 2 starts immediately following the bus being released from the queue at $(\bar{t}_k^1, l_k^1)$. Given a bus stop (I^{bus}) exists downstream of l_k^1 and the bus is not skipping this stop, it immediately follows for the cycle *k*+1 of phase *j*:

$$l_{k+1}^3 = l^{\text{bus}} \quad \text{and} \quad t_{k+1}^3 = \frac{l_{k+1}^3 - l_k^1}{-v_1} + \bar{t}_k^1 \quad \text{and} \quad \bar{t}_{k+1}^3 = t_{k+1}^3 + D_{\text{dwell}} \qquad [0]$$

$$t_{k+1}^4 = \frac{l_{k+1}^3}{v_2} + (t_{jk} + g_{jk}) \qquad [0]$$

By comparing $\bar{t}_{k+1}^3$ and t_{k+1}^4, the bus is projected to be able to leave the bus stop before the queue starts to back up to the bus stop again. Then, using $(\bar{t}_{k+1}^3, l_{k+1}^3)$ as starting point, the computation for the next segment of the bus trajectory is the same as that of the beginning segment.

On the other scenario when $\bar{t}_k^3 > t_k^4$, it results in case 3. Two points are to make for this case: (a) it is not certain at the time of computation that whether the bus will meet the queue first or the bus stop first; and (b) the part of dwell time that extends into the duration of queue blockage shall not be counted as the delay to be minimized. The former point implies that it is

necessary to compute the projected point (t_{k+2}^1, l_{k+2}^1) intersecting by the free flow trajectory as if no bus stop and the backward forming queue starting from $t_{j,k+1} + g_{j,k+1}$. Point (b) suggests $d_{j,k+2}$ be the duration between when the bus is ready to exit the bus stop to when the queue dissipates to the bus stop. Additionally, (b) further implies $d_{j,k+2}$ be negative if the bus is ready to exit after the queue has dissipated, as in case 4.

To summarize, a recursive procedure is developed to compute $\underline{BR}_{jk+1}$ for all four cases over several consecutive cycles starting from the time the bus is first detected at (t_k^0, l_k^0):

Step 1: Find the immediate past end time of phase *j*, ($t_{j,k-1} + g_{j,k-1}$), set it as zero and compute all future timings about phase *j* in reference to $t_{j,k-1} + g_{j,k-1}$. $\underline{BR}_{jk} = +\infty$.

Step 2: Compute critical time points for different cases:

- If bus stop downstream of l_k^0 and no skipping set $l_k^3 = l^{\text{bus}}$ otherwise $l_k^3 = 0$.
- Compute t_k^3 as if free flow for bus to bus stop, and (t_k^1, l_k^1), (t_k^2, l_k^2) as well.
- If $t_k^1 \le t_k^2$, $t_k^1 \le t_k^3$, $l_k^1 > 0$ and $t_k^1 > t_k^0$ Then: [// equivalently case 1]
  - $l_k^{\text{ready}} = l_k^1$ and $t_k^{\text{ready}} = t_k^1$,
  - Compute $\bar{t}_k^1$, and set $t_{k+1}^0 = \bar{t}_k^1$, and $l_{k+1}^0 = l_k^1$.
- Else:
  - If $l_k^3 = 0$
    - If $t_k^3 \le t_{jk} + g_{jk}$:
      - Go to step 5. [//There is no queue delay from cycle *k* onward]
    - If $t_k^3 > t_{jk} + g_{jk}$:
      - Set $\underline{BR}_{jk} = +\infty$. [// no queue delay for current cycle]
      - Go to step 4. [//but there will be delay for next cycle]
  - If $l_k^3 > 0$,

- Compute $\bar{t}_k^3$ and t_k^4.
- If $\bar{t}_k^3 < t_k^4$ Then: [// equivalently case 2]
  - Update $t_k^0 = \bar{t}_k^3$ and $l_k^0 = l_k^3$.
  - Go back to step 2 for current cycle.
- If $\bar{t}_k^3 \geq t_k^4$ Then: [// equivalently case 3 or 4]
  - Set $l_k^{\text{ready}} = l_k^3$ and $t_k^{\text{ready}} = \bar{t}_k^3$.
  - Compute $\bar{t}_k^1 = l_k^{\text{ready}} / v_3 + t_{jk}$.
  - Set $t_{k+1}^0 = \max\{\bar{t}_k^1, \bar{t}_k^3\}$ and $l_{k+1}^0 = l_k^{\text{ready}}$.

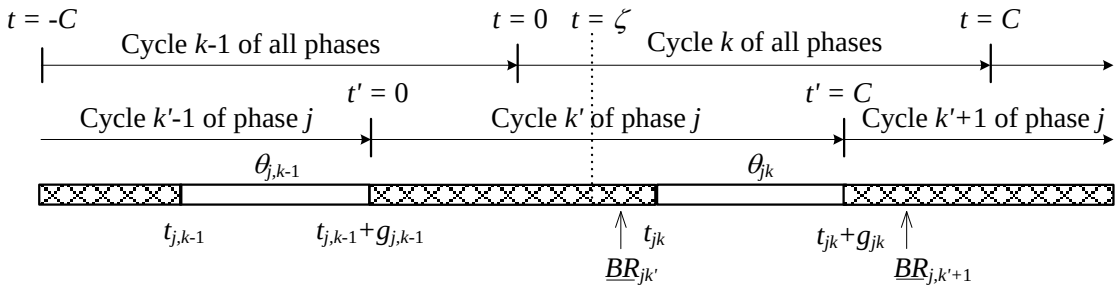
Step 3: Compute $\underline{BR}_{jk} = t_k^{\text{ready}} - l_k^{\text{ready}} / v_3$.

Step 4: Set $k = k+1$, if $k \leq K$, continue from step 1.

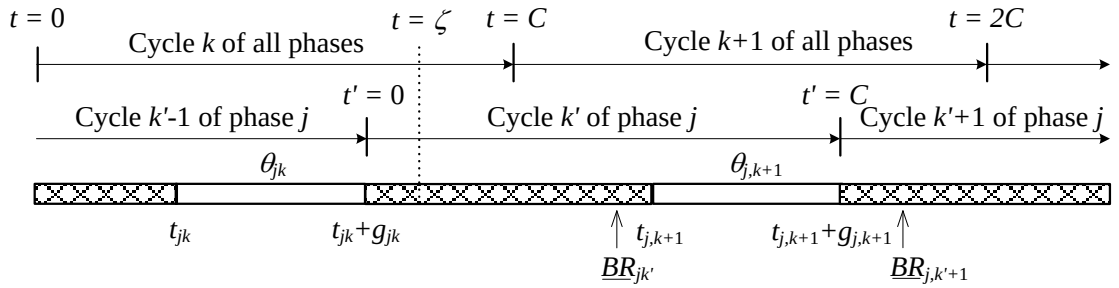
Step 5: Terminate.

4.2.3.2. Cycle Definitions

During the implementation process, it is essential to clearly define the cycle of phase *j* and how it is related to the cycle of all phases. Figure 10 explains the how are the definitions of the cycles related to the detection time of the bus in a cycle. The time zero for all phases (i.e., $t = 0$) shall refer to the beginning of the first phase in the cycle. However, in the computation of queue delays, the reference zero time (i.e., $t' = 0$) is always the most recent end time of phase *j* green before the bus detection time (i.e., $t = \zeta > 0$). For case shown in Figure 10a, the numbering for cycles are the same for the two temporal referential systems; the numbering differs when bus detection occurred after the end time of phase *j* green in current cycle, as in Figure 10b.



(a) Bus Detection before Phase *j* Green Ends in Cycle *k* of all phases



(b) Bus Detection after Phase *j* Green Ends in Cycle *k* of all phases

Figure 10: Definitions of Cycles in Relation to Detection Time.

4.2.4. Nonlinear Bus Trajectory

The objective for computing the five critical time-space pairs is to provide estimates of a set parameters $\underline{BR}_{jk}$ and one parameter $\overline{BR}_j$, from which the queue delay d_j of a bus can be computed. The definition of these points remains valid even if the linear assumption about the bus trajectory is relaxed, but the computation procedures for these points may not. Therefore, adjustments on the computation procedures help improve the estimation of the critical parameters to better represent realistic bus trajectories.

4.2.4.1. Computation of Nonlinear Bus Trajectory without Queue Delay

The parameter $\overline{BR}_j$ is most critical in the computation of queue delay, because it is the reference time when the bus actually needs the green time. Estimation of this parameter needs to be as accurate as possible. Fortunately, this parameter is defined by assuming no interactions of the bus with the queue of other vehicles. Hence, it can be easily computed using either free flow travel time if no bus stop is present or parabolic vehicle trajectory in and out of the bus stop. For

the latter case, the exact locations before and after the bus stop where the bus starts to decelerate and accelerate at a constant rate can be easily determined. The bus trajectory is nonlinear in the area enclosed by these two locations and is linear outside of it. The standard rates 1.2 and 1.3 m/s are used for acceleration and deceleration, respectively.

4.2.4.2. Adjustments for Bus Stop Entry speed

Figure 11a illustrates the difference between linear and nonlinear trajectories. Let Δ_1 denote the time from the detection of the bus to when the bus arrives at the bus stop; let $\Delta_1 + \Delta_2$ denote the time the bus would actually need to apply a constant deceleration rate to stop at the bus stop, assuming no queue blockage. It is possible that the bus is projected to meet bus stop first, which does not incur any queue delay according to the discussions before. In a real situation, the curved trajectory implies that the bus may actually meet the queue first, which would be delayed until the queue dissipates. Therefore, a fine-tuning of bus arrival time is needed. To equate the two trajectories with the same distance *l*, and assumes constant *a*:

$$\begin{aligned} v_1\Delta_1 = l = \frac{v_1^2}{2a} \quad &=> \quad \Delta_1 = v_1 / 2a \\ v_1 = a(\Delta_1 + \Delta_2) \quad &=> \quad \Delta_2 = v_1 / 2a \end{aligned} \qquad [0]$$

v_1' is exactly half of v_1 (i.e., $v_1' = v_1 / 2$). That means using a constant entry speed between 50–100 percent of the detected speed can give a good approximation to the nonlinear bus trajectory.

However, 50 percent range is still wide. To select a better percentage, researchers broke it down into three cases based on whether the backward forming queue shockwave is projected to arrive at the bus stop before or after the linear and nonlinear bus trajectory. Figure 11a, b, and c clearly illustrate the three cases: (a) queue shockwave arrives at bus stop between the arrival times projected by both trajectories; (b) queue shockwave arrives after the nonlinear trajectory; and (c) queue shockwave arrives before the linear trajectory. Seventy-five percent of the detected speed should be used as the entry speed for the bus for case (a), 50 percent should be used for case (b) (i.e., v_1'), and 100 percent be used for case (c) (i.e., v_1).

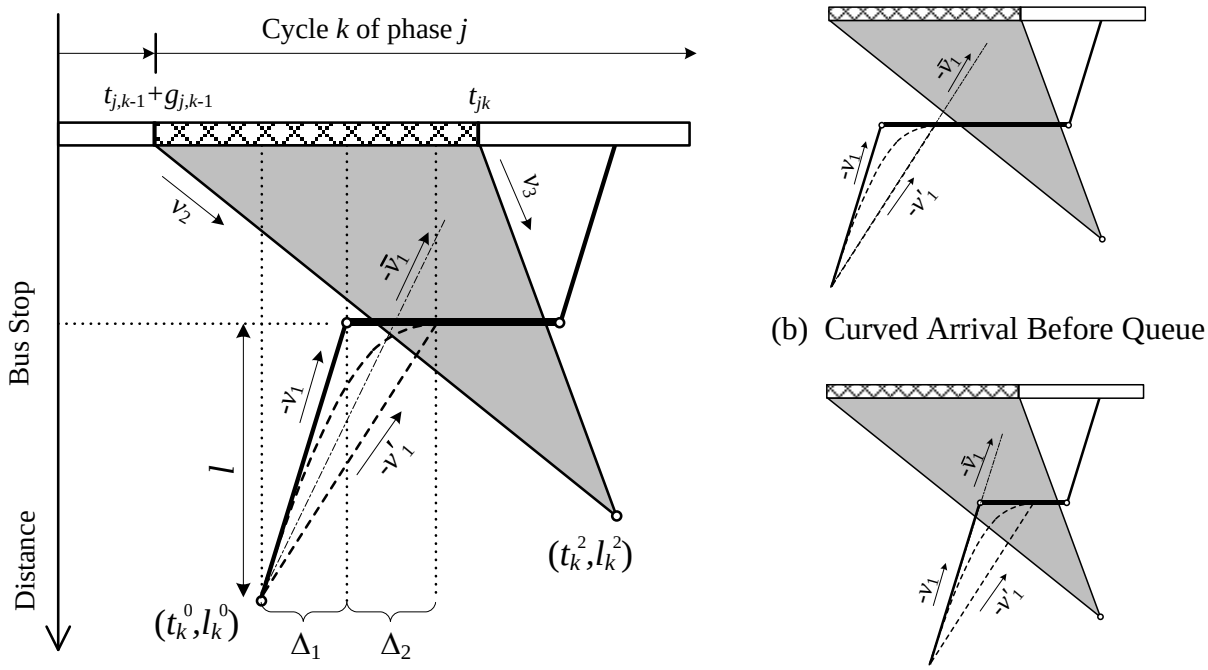


Figure 11: Adjustment for Nonlinear Bus Trajectory.

4.3. ONLINE IMPLEMENTATION SCHEMES FOR MULTIPLE BUSES

As mentioned before, real-time capability is an important design factor for an adaptive TSP system. In order to achieve this, the system will continue to operate regardless of when and how many buses arrive at the intersection. Specifically, the system should be able to conduct optimization sessions and implement the timing results at any point on a time horizon. Since an optimized timing result typically lasts at least two cycles before it returns to normal cycles, it is likely more than one bus arrives and needs priorities within the planning horizon. This implies a real-time control system has to be developed that can account for multiple buses either at the same time or sequentially.

The mathematical formulation developed above allows the arrival inputs from multiple buses. When bus arrival information is available simultaneously, one optimization session that uses all bus arrival times is needed. However, when the arrival information of buses during the planning horizon is available separately, the optimizations have to be done in an incremental fashion. Two implementation schemes of an online TSP system emerge: (a) fixed-interval

optimization and (b) rolling-optimization. Figure 12 illustrates the differences of these two methods and shows when the optimizations are taken place for the same bus arrival scenarios.

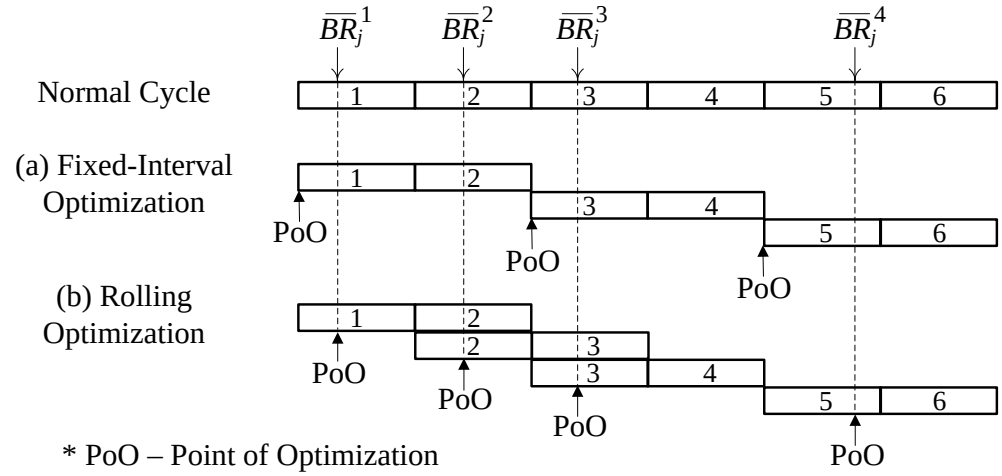


Figure 12: Techniques of Optimization for Multiple Buses.

4.3.1. Fixed-Interval Optimization Scheme

The fixed-interval optimization requires only one optimization session per planning horizon. This optimization scheme considers all buses are done once at the beginning of a planning period, and the results are implemented once for the planning period. No changes need to or should be made during the planning period. For example, $\overline{BR}_j^1$ and $\overline{BR}_j^2$ arrive during the first planning period (i.e., cycles 1 and 2) in Figure 12; only one optimization session is run at the beginning of cycle 1. This resembles an offline control method where all information is available at the time of signal optimization and there is no overlapping of the planning periods. This method guarantees an optimal timing for all buses. However, to obtain the information of all buses potentially arriving at the intersection would normally require an advanced prediction model of bus arrival time based on schedule information, traffic conditions, and perhaps the signal timing of upstream intersections. The predictions need to look at minutes into the future to make estimations of the arrival times that have precisions in the scale of seconds. The longer the planning horizon, the more complicated and uncertain the predictions will become.

4.3.2. Rolling Optimization Scheme

The rolling optimization scheme allows an optimization to be made as soon as a new bus or a change in conditions is detected. This method may only require very short-term predictions of bus arrival to be made, so it is not likely that the model would suffer from unnecessary delay caused by inaccurate arrival information. The trade-off for this real-time control capability is that the resulting timing for all buses is not guaranteed to be optimal. This is because an early optimization session does not consider the arrivals of later buses, while a later optimization session is subjected to the timing changes already implemented by an early session. In addition, the optimization for the later bus may modify the timing such that the first bus cannot pass as previously expected. To avoid the optimal timing for the first bus being overwritten by the arrival of the second bus, current practice normally enforces a recovery period, during which no new TSP requests will be processed. This can easily lead to a FCFS control strategy, which is the biggest disadvantage of the conventional TSP strategies with fixed-location check-in and check-out system. Unfortunately, if the bus arrival information can be collected only in separate times, the FCFS control seems to be the only option for real-time TSP implementation. This is because the decisions for priority have to be made in separate time for each bus separately.

Researchers propose using the background optimal timing concept, where bus priority does not have to be granted on a FCFS basis even if the arrival information is available only separately. The background optimal timing is redefined as not only the normal operations given the prevailing traffic conditions but also all the priority requests previously granted. For example, the optimization for the first bus, $\overline{BR}_j^1$, changes the signal timing originally optimized for only the general traffic in order to accommodate the priority need of the bus; the planning horizon is cycles 1 and 2. When the second bus, $\overline{BR}_j^2$, arrives soon after and requests for a priority, the background timing is the optimal timing considering both the prevailing traffic conditions and the first bus priority request. Any further deviation from this background timing will incur cost not only to the general traffic but also to the first bus. The planning horizon is cycles 1, 2, and 3. By using this concept, the optimizations planning horizon can roll on continuously until no more bus arrivals before the end of the dynamic planning period.

With the aid of the connected vehicle technology, the rolling optimization technique can be further enhanced. Since the connected vehicle technology provides continuous communications

between a bus and the infrastructure, the control system can request updated arrival information from the first bus when a second priority request is received. The second optimization is made by using the most current arrival information from both buses.

4.3.2.1. Variable Cycle Length in Rolling Optimization Scheme

When rolling optimization scheme is implemented for multiple buses, a practical issue emerges for allowing the cycle lengths to vary. When the start of a cycle is not fixed, it is possible that after a few optimizations, the optimized timing will completely fall out of sync with the background cycle timing. To ensure the synchronization of the ends of the optimized and the background cycles, it is important to make a record about the amount of offset between the expected and the actual start times of the optimization cycle. Figure 13 illustrates how the variable cycle length procedure can be implemented.

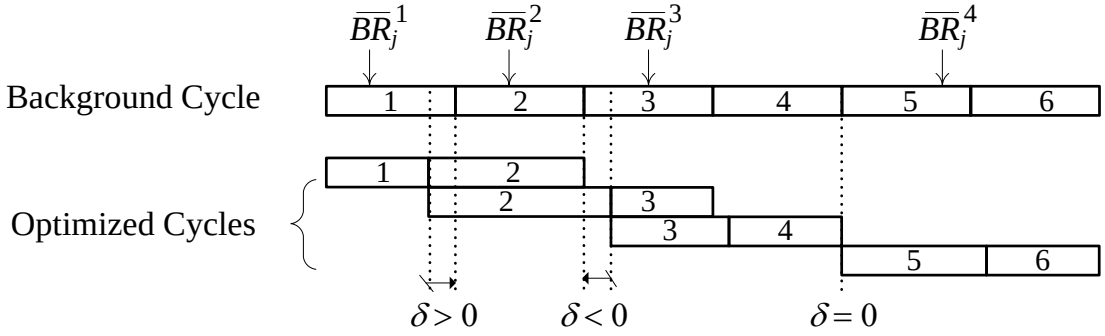


Figure 13: Variable Cycle Length Implementation in a Rolling Optimization Scheme.

Assume the optimization takes the timings of the next two cycles into consideration. When bus 1 arrives after phase *j* in cycle 1, the optimization program cuts the entire cycle 1 short to bring up phase *j* in cycle 2 earlier. Normally, the optimized timing will bring the timing back to normal in the third cycle. However, if a bus 2 arrives in the second cycle when the optimized timing is being implemented, the difference between the background start time of cycle 2 and the actual start time of cycle 2 is added to the cycle length constraint:

$$t_{jk} + v_{jk} = kC + \delta; \qquad j = J\{last\}, k = |K| \qquad [0]$$

δ is positive if actual start time of the cycle is earlier than the background start time of the cycle, and it is negative otherwise. This procedure can be applied to any number of look-ahead

cycles. If multiple buses arrive in consecutive cycles, this procedure ensures the intersection timing returns back to normal synchronization after all look-ahead cycles are implemented.

5. SIMULATION TEST BED AND NUMERICAL EXPERIMENT

5.1. SIMULATION TEST BED ARCHITECTURE

A simulation platform is developed to implement the proposed SMINP model and to evaluate its performance against current state-of-the-practice TSP-enabled signal control system. The entire platform is coded and complied using the Microsoft Visual Studio C++ compiler. Figure 14 illustrates the general architecture of the simulation platform, which consists of the following three main modules:

- Optimization: the IBM CPLEX solver through the CPLEX Callable Library.
- Signal Control: self-developed C++ functions to implement the optimized timing splits.
- Simulation: the PTV VISSIM traffic simulator and a fixed-time VAP signal controller.

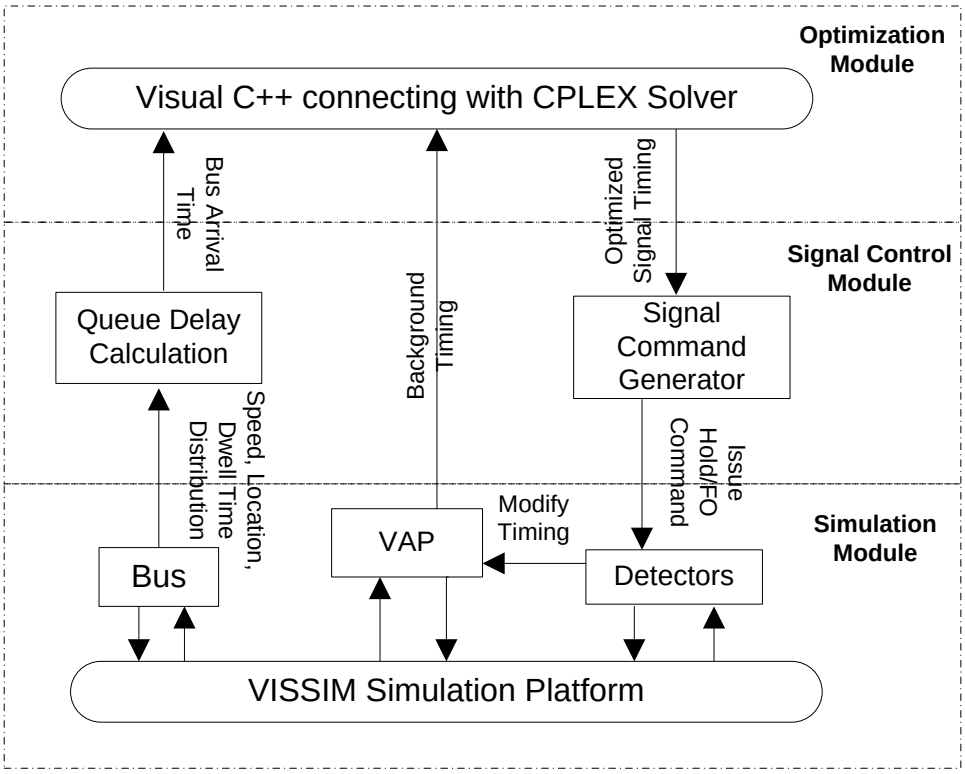


Figure 14: General Architecture of the Simulation Evaluation Platform.

5.1.1. Simulation Module

The simulation module fully utilizes the component object model (COM) available in the standard VISSIM 5.4 package. Via the COM programming language, two types of vehicles are

able to be modeled and controlled: (1) buses with connected vehicle OBU, and (2) a parked vehicle on the roadside simulating a connected vehicle roadside unit (RSU).

At a user-defined interval, buses with OBUs take snapshots of signal phase it is requesting, whether it intends to skip the bus stop, desired speed, passenger load, and other instantaneous data, such as the bus current speed, location, the time it has spent on the bus stop loading/unloading and so forth. An OBU also estimates the dwell duration at a downstream bus stop, which can be utilized to estimate the bus queue delay. Currently, an OBU does not collect vehicle information from other nearby vehicles because general traffic is assumed to be unequipped. Researchers believe the additional information from other vehicles can help improve the accuracy of queue delay estimation. The RSU uses a search radius (communications range) and constantly searches for approaching OBU/buses at a fixed time interval (e.g., every second). If a bus with an OBU enters the range of the RSU, a communications link will be established and information about the RSU and the OBU is exchanged. The RSU stores all the signal timing parameters, bus stop location, lane configuration, current prevailing traffic conditions in terms of volume, if the current timing is affected by another bus, and so on. Figure 15 illustrates the main data communicated between the OBU and RSU as well as the time sequence and communication direction of all data. Communications between an OBU and RSU is assumed to occur instantly without any delays.

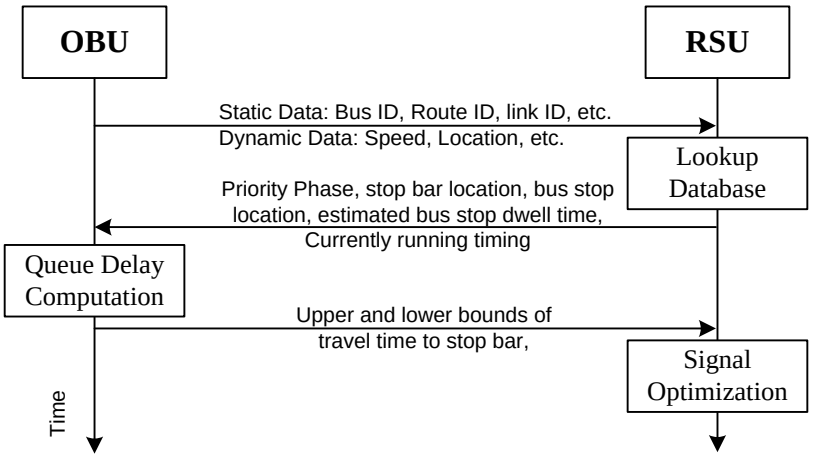


Figure 15: Data Flow between an OBU Equipped Bus and the RSU.

5.1.1.1. Signal Control in the Simulation Module

A fixed-time signal control is implemented in VISSIM using the built-in vehicle actuation programming (VAP) language. The control runs on a standard eight phase two ring timing structure. When no optimization routine is performed, the VAP control runs as designed. The architecture of the system does not restrict the use of VAP as the only signal control method. Other control systems can also be implemented, because the signal control module implements a two universal signal control command—force-off and hold; see next section.

5.1.1.2. Calibration for Saturation Flow Rate

Saturation flow rate is one of the most important parameters in the simulation that will affect the computation of queue delay, degree of saturation, objective function weighting factor, and so on. This parameter is not always in agreement over various traffic simulation and/or optimization packages. In SYNCHRO, the saturation flow rate was determined as 1624 vehicles per hour per lane (vphpl). The default acceleration rate in VISSIM renders a higher saturation flow rate at about 1800 vphpl. Calibrating the vehicle acceleration rate alone is sufficient to adjust the saturation flow rate to a desired value (e.g., 1624 vphpl in this case).

5.1.2. Signal Control Module

The signal control module serves as the primary link between the simulation and the optimization modules. The control module extracts information from the simulation model and determines if an optimization routine should be triggered. If an optimization is needed, the module formats relevant vehicle information into usable inputs to the optimization model. After an optimization routine is completed, it extracts the optimization outputs, and makes decision on when and how to implement the results. The control and data flow back and forth in this module. The two control flows are described in the Figure 16. To achieve the data flow, the following sub-processes are necessary:

- Get vehicle data – This process extracts and formats useful vehicle data.
- Get the timing status – This process monitors the current signal status for each phase, so that other sub-processes can implement routines accordingly.

- Countdown timer – Each phase in the planning horizon has one timer. For example, if there are 8 phases per cycle, and the planning period has two cycles, then there are 16 timers. This process is called every second to countdown the active timers.
- Set and reset timer – After an optimization routine, the new signal timings are implemented in the timers. So this process set or reset the timers accordingly.
- Issue control command – During the implementation of optimized timing, the signal control module take full control of the signal system. When the countdown timers of the active phases reach zero, the force-off commands are issued. Otherwise, hold commands are placed every second until the force-off commands. The use of only two commands (i.e., force-off and hold) allows the system to be easily extended to any other types of signal controllers.

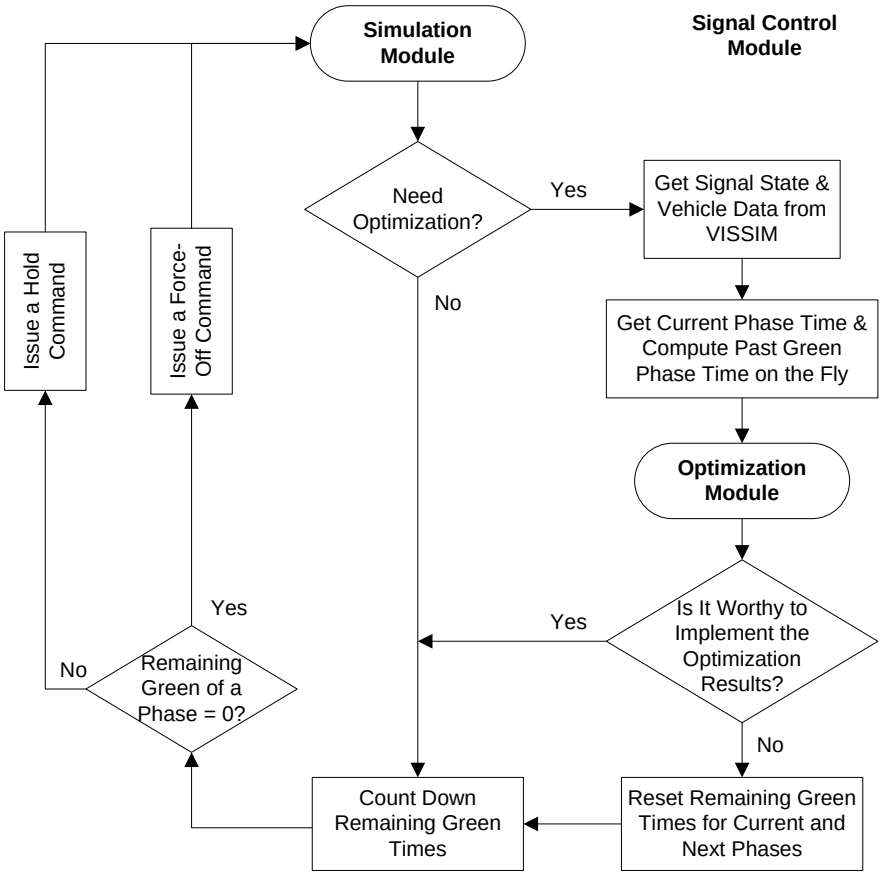


Figure 16: VISSIM Signal Control Module.

In addition to relaying the vehicle and timing information back and forth between the simulation and optimization modules, the signal control module also keeps an independent timer that archives the signal timing. The independent timer is critical for the rolling optimization scheme.

5.1.3. Optimization Module

The optimization is the core module where the model for a TSP strategy is implemented. Upon receiving the bus data and the signal timing data from the controller, the optimization module formulates an initial SMINP model with only one stochastic scenario. The module then reformulates the SMINP model into its deterministic equivalent program (DEP) by enumerating all possible combinations of stochastic scenarios. If the number of stochastic scenarios is relatively small, the DEP can be directly solved using the standard procedures in the CPLEX solver.

In stochastic programming term, the number of scenarios could grow exponentially. Using the bus dwell time as an example, discretize the dwell time of one bus into *S* distinct outcomes (assigning each a probability) and *N* such buses arriving at the same time, then the total number of scenarios is S^N. Each scenario corresponds to a set of second-stage constraints, m_2. That means the mathematical program will grow into a large program with $(m_1 + m_2 S^N)$ number of constraints, where constant m_1 is the number of first-stage constraints.

In this research, the number of buses arriving at a given short-term period (i.e., two cycles) is small, no more than 3. With a small number of discretized outcomes for each bus, the size of the DEP is still manageable and it can be solved quickly. However, if the number of stochastic scenarios is large, using DEP may not be a viable option because the computation time for a very large program may become prohibitive; advanced optimization algorithm/routines need to be developed to ensure the solvability of the stochastic program. Unless the current MILP formulation is changed significantly, such advanced optimization algorithms have to handle a two-stage stochastic program with the binary variable in the second stage.

5.2. TEST INTERSECTION SETUP

The DEP was formulated for the proposed SMINP optimization model. The background optimal timings used in as the inputs for the SMINP model were obtained from a commercial signal

optimization package, SYNCHRO. The DEP form of the SMINP with optimized background timing was then applied to a hypothetical four-leg intersection, as shown in Figure 17, with a near-side bus stop at about 60 meters (196 feet) from the stop bar. It is assumed that the intersection is equipped with one RSU that can detect the presence of the approaching bus and obtain information related to the bus speed, current location, and possibly a most updated dwell time data collected and maintained by the transit agency. The collection of bus data is continuous as long as the bus is within the coverage of the RSU.

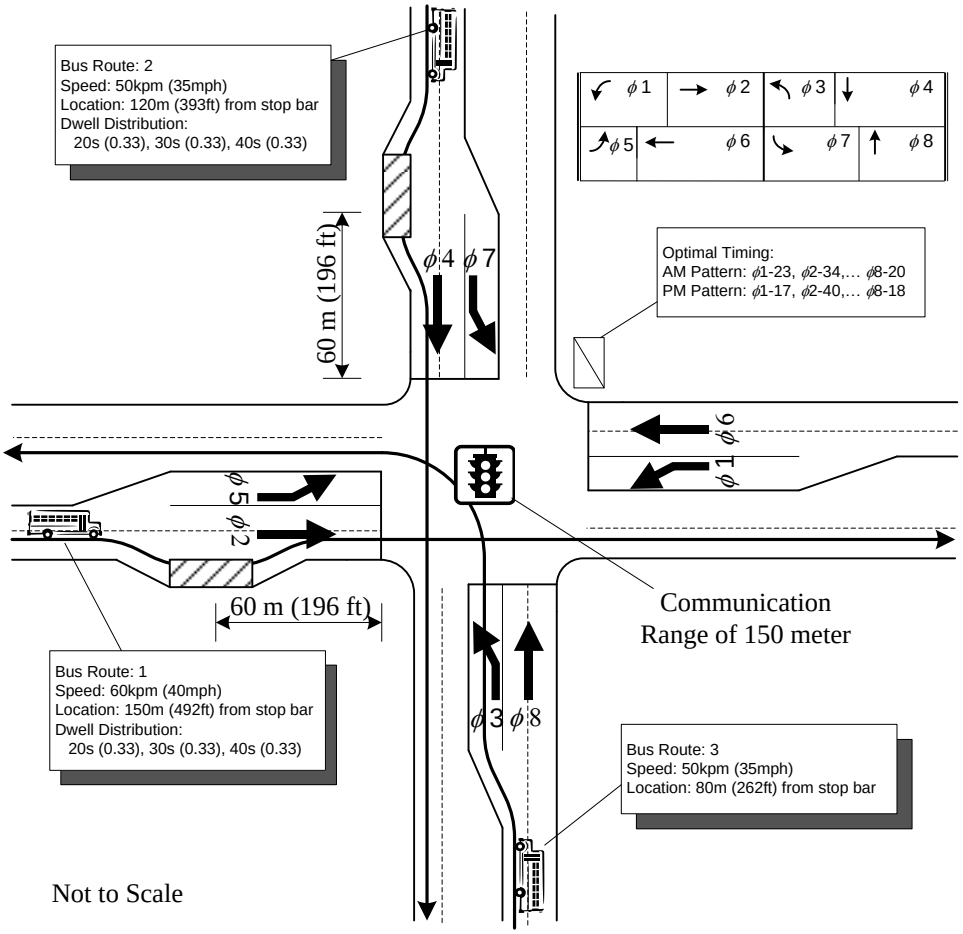


Figure 17: Hypothetical Intersection with Near-Side Bus Stop for Model Testing.

5.3. NUMERICAL EXPERIMENTS

5.3.1. Level of Priority Tests

As mentioned previously, the weight of the priority delay, o_{jn}, is a crucial factor that allows the user to define the level of importance for the priority request. Different settings of this weighting

coefficient may change the outcome of signal timing. In this test, researchers used a medium congested volume setup to test how the vehicular delays for both the bus and passenger cars respond to the change of this coefficient setting, starting from 0.1 to 6.65 at an increment of 30 percent.

Table 7: Background Optimal Timing for Evaluations.

Background Timing 1: Cycle Length = 100 sec								
Phase	ϕ1	ϕ2	ϕ3	ϕ4	ϕ5	ϕ6	ϕ7	ϕ8
# of lanes	1	2	1	2	1	2	1	2
Volume	150	820	130	540	100	1350	150	250
Optimized splits	21	42	14	23	12	51	16	21
v/c	0.54	0.66	0.80	0.87	0.76	0.88	0.76	0.44
Intersection Delay	34.1							

Table 7 shows the volume, the optimized split, and the degree of saturation for each phase of a 100 second cycle. By fixing the dwell time, the arrival of the bus is controlled at a fixed point of the cycle, so there is no other variation except the priority weighting coefficient. Five random simulation seeds are used across all priority scenarios. Figure 18 illustrates the general trend of the bus delays and the passenger car delays with respect to the levels of priority. As expected, the increase of the priority weights for the bus decreases its delay and increases the delay for traffic on conflicting phases. The reason is that programs tend to keep the split; this way it is not hurting traffic on non-transit phases as much. Notice that the bus priority starts to level out at the priority level at around 13, meaning increasing the priority further would not generate any benefits to buses. Of course, this number would change according to a different congestion level. The next experiment indirectly showed this.

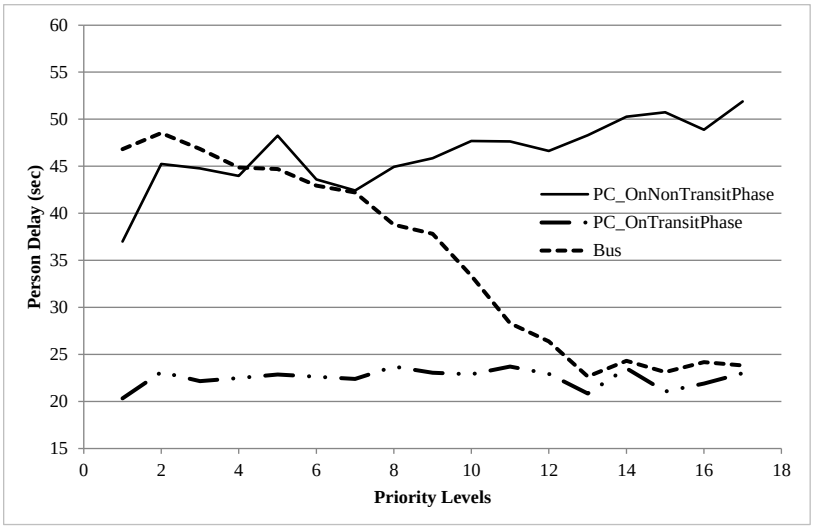


Figure 18: The Impact of Priority Setting on Bus Delays.

5.3.2. Comparison of Control Systems

Researchers compared the proposed model with TSP feature implemented in the Ring Barrier Controller in VISSIM (*PTV America 2010*). The RBC is a unified signal control emulator, which has implemented many of the most significant features of a real-world signal controller. Although it is not developed to exactly replicate the interface of a certain signal control model, its features are realistic enough to represent the existing functionalities of a typical modern signal controller. The RBC uses a pair of check-in and check-out detectors to enable its TSP feature. Upon the detection of a bus at the check-in detector, a constant travel time with a constant slack time is applied to estimate its arrival time interval at the stop bar and performs either green extension or red truncation. With a near-side bus stop, the check-in detector is recommended to be placed at the bus stop (*PTV America 2010*). The need to account for the random dwell time is eliminated.

The TSP strategies implemented by both the RBC and the SMINP are compared with the baseline fix-time do-nothing control strategy. To compare these three control types on fair ground, fixed cycle splits are implemented in the RBC controller as well. Table 8 shows the setup of three congestion levels represented by the volume-to-capacity (V/C) ratios. All splits are optimized in SYNCHRO with the respective volume levels. The dwell time is assumed to be discrete uniformly distributed with possible outcomes of 20, 30, and 40 seconds. Two bus arrival frequencies are tested: 5 and 10 minutes. Five random seeds are simulated for each of the

volume and arrival frequency combination. A fixed priority coefficient for the SMINP was used for all cases.

Table 8: Parameter Setup for Simulation Evaluations.

Background Timing: Cycle Length = 110 sec Dwell Time Distribution: 20 sec (0.333), 30 sec (0.333), 40 sec (0.333)								
Phase	$\phi 1$	$\phi 2$	$\phi 3$	$\phi 4$	$\phi 5$	$\phi 6$	$\phi 7$	$\phi 8$
# of lanes	1	2	1	2	1	2	1	2
V/C = 0.5								
Volume	112	616	90	381	78	784	101	280
Optimized splits	23	40	20	27	19	44	21	26
V/C = 0.7								
Volume	156	858	125	530	109	1092	140	390
Optimized splits	22	44	17	27	16	50	19	25
V/C = 0.9								
Volume	200	1100	160	680	140	1400	180	500
Optimized splits	21	46	15	28	14	53	17	26

5.3.2.1. Evaluation with Single Bus Line

Assuming only bus route No. 1 in Figure 17 has regular bus arrival at the intersection, researchers tested two arrival frequencies under all three degrees of saturation levels in Table 8. The bus headways for both frequency scenarios (i.e., 5 and 10 minutes) are larger than the planning horizon (i.e., two cycles of 110 seconds). That implies there will be no overlapping period between two consecutive optimization sessions. The impacts of priority services are independent from one another.

Figure 19 illustrates the changes of vehicle delays comparing to the baseline fix-time control for each combination of volume and arrival frequency. It can be seen that both Built-in Ring-Barrier Controller (RBC-TSP) and SMINP give signal priority to the bus, resulting in much lower bus delay across all scenarios. The SMINP outperforms the RBC-TSP at all scenarios. In some scenarios, the difference is as large as a 30 percent improvement from the RBC-TSP and a 60 percent improvement from the baseline do-nothing scenario. This means that the proposed model was able to better capture the bus arrival time and adjust the timing to favor the bus more. Another reason for the significant improvement is due to the ability of the proposed model to plan ahead. The optimization was done at the time the bus was detected before the bus stop,

while the RBC-TSP only performs calculations of signal timings for the bus at the time it is leaving the bus stop. There are about 30–50 seconds more time for SMINP to adjust the timing. The benefits of this are that not only the bus delay has reduced significantly, the disturbance to other traffic is comparable or smaller.

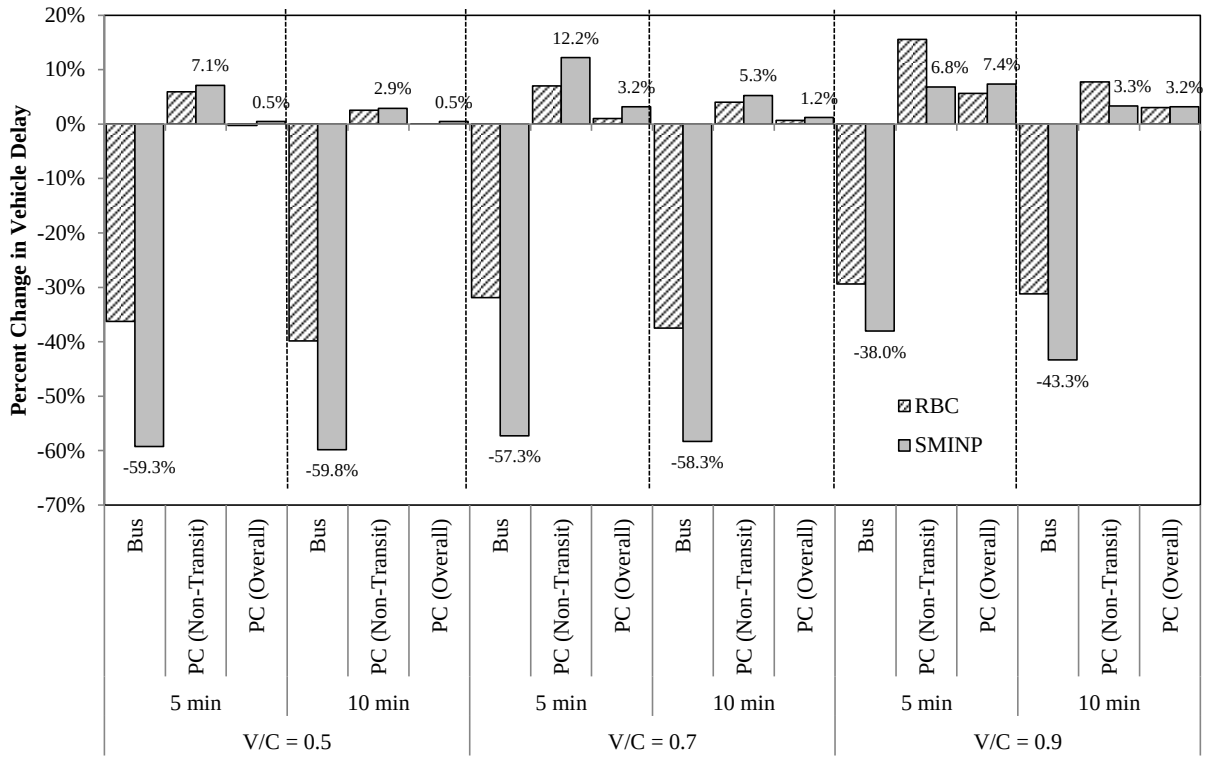


Figure 19: Percent Change in Vehicle Delays for RBC and SMINP vs Fix Time Control under Single Bus Arrival Scenario.

On the other hand, the SMINP is much more responsive to the expected traffic conditions than the RBC controller. This is especially evident at high volume conditions (i.e., V/C = 0.9). At this volume level, when the bus is arriving less frequently, the delay of traffic on non-transit phases are about 8 percent better than the RBC-TSP. When a bus arrives at about 5 minute intervals, the delay to non-transit vehicles have skyrocketed to about 20 percent more than the baseline fix time control, while the SMINP maintains only about 5 percent increase from baseline. The ability to be responsive to the traffic condition is because the mathematical model uses the normalized degree of saturation for each phase to spread-out the total number of seconds across all phases in the planning horizon to satisfy the bus priority needs. In this way, the start time of the phases may change significantly but the duration of the phase tends to be kept at their

optimal values. The result is a much improved bus delay with much less cost to the traffic on its conflicting phases. The delay values of the all compared scenarios are shown in Table 9.

Table 9: Vehicle Delays by Control Types with Single Bus Line.

Intersection Degree of Saturation	Arrival Frequency	Vehicle Delay Type	Control Model		
			Fixed	RBC	SMINP
V/C = 0.5	5 min	Bus	40.3	25.7	16.4
		PC (Overall)	34.2	34.1	34.3
		PC (Non-Transit)	40.1	42.4	42.9
		PC (Transit)	30.3	28.6	28.7
	10 min	Bus	42.9	25.8	17.2
		PC (Overall)	34.1	34.1	34.3
		PC (Non-Transit)	40.1	41.1	41.2
		PC (Transit)	30.2	29.5	29.7
V/C = 0.7	5 min	Bus	39.6	27.0	16.9
		PC (Overall)	34.9	35.3	36.1
		PC (Non-Transit)	43.6	46.6	48.9
		PC (Transit)	29.2	27.8	27.6
	10 min	Bus	42.6	26.6	17.8
		PC (Overall)	34.9	35.1	35.3
		PC (Non-Transit)	43.5	45.2	45.8
		PC (Transit)	29.2	28.5	28.4
V/C = 0.9	5 min	Bus	42.3	29.8	26.2
		PC (Overall)	39.2	41.4	42.0
		PC (Non-Transit)	51.3	59.3	54.8
		PC (Transit)	31.2	29.7	33.7
	10 min	Bus	44.3	30.5	25.1
		PC (Overall)	39.0	40.2	40.2
		PC (Non-Transit)	51.3	55.3	53.0
		PC (Transit)	30.9	30.3	31.9

Note: PC (Overall) – All passenger cars on all approaches

PC (Non-Transit) – Passenger cars on phases conflicting with the bus requested phase

PC (Transit) – Passenger cars on phases concurrent with the bus requested phase

5.3.2.2. Evaluation for Multiple Bus Lines

Assuming there are more than one bus routes running through the intersection regularly, researchers varied the number of conflicting bus routes (i.e., two and three) under all three degrees of saturation levels as in Table 8. The headways for bus routes No. 1, 2, and 3 as in

Figure 17 are set to 5, 6, and 8 minutes, respectively. Consequently, in any one scenario, the timing optimization process for one priority service is inevitably affected by the timing changes for another priority service request. The impacts of priority services are dependent from one another. In these complicated cases, the rolling optimization scheme has to be deployed to ensure the priority signal control can be performed continuously. Refer to section 4.3.2 for details.

In particular, the SMINP model in this experiment used the incremental rolling method, where each optimization is done with the inclusion of only one vehicle. The priority level for each route is now set to 5, 3, and 2, respectively. So route 1 has the highest priority and route 3 has the lowest since it is a cross-street left-turn phase. Routes 1 and 2 have to come to a stop at their respective bus stops before arriving at the stop bar while route 3 does not need to stop at any bus stops. The dwell time for both routes 1 and 2 follow the same discrete uniform distribution with equiprobable outcomes of 20, 30, and 40 seconds. A rule was applied in the system to prevent the rolling optimization from continuing indefinitely. The rule ignores the all the priority requests after the dynamic planning horizon has been extended to 5 cycles or more. After the timing recovers back to the background optimal timing at the end of the 6 cycle, new priority requests will be considered.

Figure 20 illustrates the changes in vehicle delays in terms of percentage when comparing the RBC-TSP and SMINP controls with the fixed-time control, and Table 10 shows the absolute delay values. From the figure, several interesting observations can be drawn immediately. First, the RBC-TSP is slightly better than SMINP in terms of non-transit phase delay and overall PC delay in low to medium degrees of saturation levels when only routes 1 and 2 are running. In all the other cases, the RBC-TSP under-performs the SMINP. Especially when V/C = 0.9, the RBC-TSP has failed to maintain the impacts of the priority service to an acceptable level, yielding 50 ~ 110 percent increase in terms of overall PC delay and 40 ~ 70 percent increase in terms of non-transit phase delay. This is because in high V/C cases the RBC-TSP has no mechanism to capture the intensity of traffic to dynamically underplay the importance bus priority requests in real-time. It is possible, in an offline setting, to fine-tune some of the RBC-TSP settings (*PTV America 2010*) such as the priority min green, recovery min green, etc. But even by doing this, a large amount of refined settings need to be done in order to adjust the RBC-TSP setting in response to the changing traffic conditions. On the contrary, the SMINP can intelligently

recognize the degree of saturation for each phase and automatically finds the balance between the general traffic and the buses in real-time for multiple bus routes.

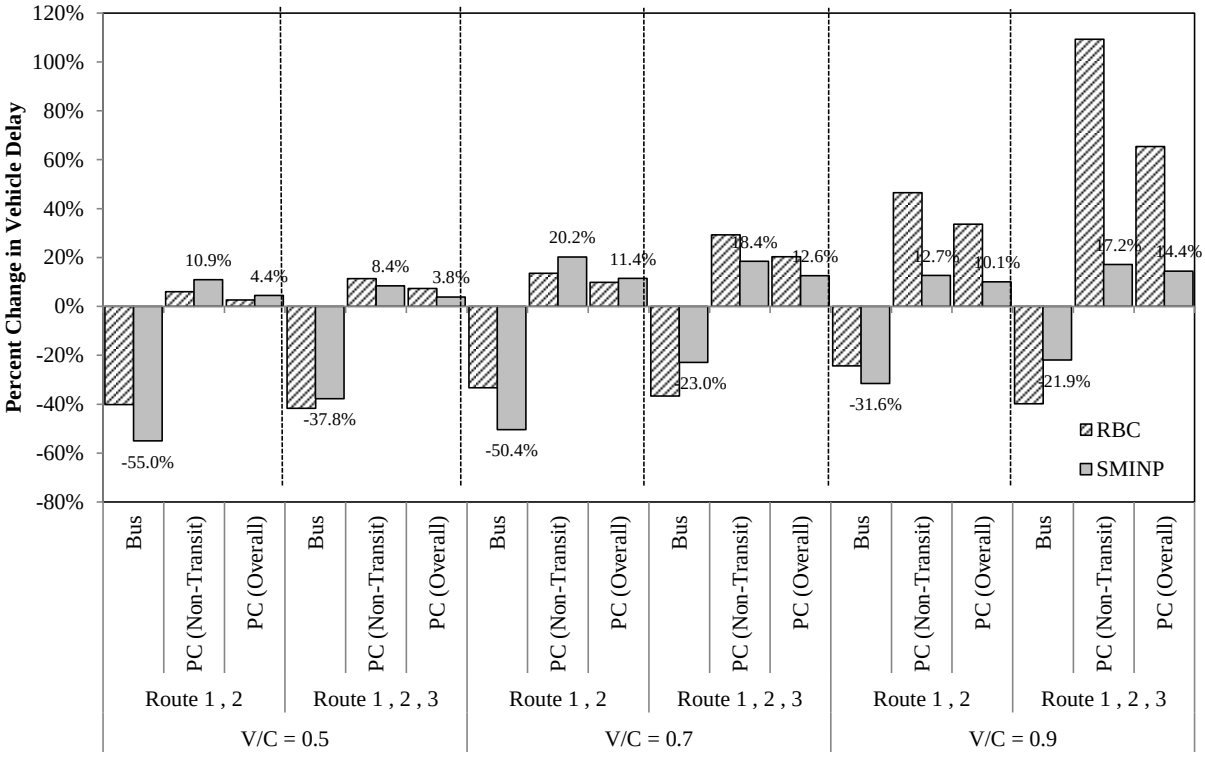


Figure 20: Percent Change in Vehicle Delays for RBC and SMINP vs Fix Time Control under Multiple Bus Arrival Scenario.

Table 10: Vehicle Delays by Control Types for Multiple Bus Lines.

Intersection Degree of Saturation	Running Bus Routes	Vehicle Delay Type	Control Model		
			Fixed	RBC	SMINP
V/C = 0.5	Route 1, 2	Bus	47.3	28.3	21.3
		PC (Overall)	34.5	36.6	38.3
		PC (Non-Transit)	34.3	35.2	35.8
		PC (Transit)	33.9	33.2	32.3
	Route 1, 2, 3	Bus	46.8	27.3	29.1
		PC (Overall)	33.8	37.7	36.6
		PC (Non-Transit)	34.3	36.8	35.6
		PC (Transit)	34.8	35.8	34.3
V/C = 0.7	Route 1, 2	Bus	46.4	31.0	23.0
		PC (Overall)	35.4	40.3	42.6
		PC (Non-Transit)	35.0	38.4	39.0
		PC (Transit)	34.3	35.8	33.9
	Route 1, 2, 3	Bus	48.0	30.4	37.0
		PC (Overall)	34.3	44.4	40.7
		PC (Non-Transit)	35.0	42.2	39.4
		PC (Transit)	35.8	39.4	37.9
V/C = 0.9	Route 1, 2	Bus	48.9	37.0	33.4
		PC (Overall)	40.8	59.8	46.0
		PC (Non-Transit)	39.2	52.4	43.2
		PC (Transit)	37.0	41.9	39.3
	Route 1, 2, 3	Bus	64.3	38.7	50.2
		PC (Overall)	38.1	79.7	44.6
		PC (Non-Transit)	40.1	66.3	45.8
		PC (Transit)	42.6	49.8	47.4

Note: PC (Overall) – All passenger cars on all approaches

PC (Non-Transit) – Passenger cars on phases conflicting with the bus requested phase

PC (Transit) – Passenger cars on phases concurrent with the bus requested phase

6. SUMMARY AND FUTURE DIRECTIONS

6.1. SUMMARY AND CONCLUSIONS

This research focused on advancing the state-of-the-art transit signal priority control system. An optimization-based real-time signal control system that can accommodate bus priority requests was developed. At the core of the system, a stochastic mixed-integer nonlinear model was proposed to optimally determine timing adjustments when receiving a bus priority request. The model used a novel approach to capture the impacts of the priority operation to other traffic by using the deviations of phase split times from optimal background split time. In addition, the stochastic formulation explicitly modeled the randomness of a bus arrival time to the stop bar that was most evident when a near-side bus stop was present. The proposed model not only captured the random dwell time of the bus at the bus stop but also accounts for the interactions of the bus with the passenger car queue, and was able to minimize the delay to the bus caused by signal timing as well as the vehicle queue.

A series of Proof-of-Concept (POC) experiments were first conducted to demonstrate some of the model's basic behaviors. The POC experiments provided insights into further refining the model to handle multiple conflicting bus lines in real-time on a continuous basis. The enhanced SMINP was implemented in a simulation evaluation test bed. The test bed was developed using a combination of one microscopic traffic simulator, VISSIM, and one commercially available optimization solver, CPLEX. A preliminary experiment was conducted on a hypothetical intersection with eight phases and running on a fixed cycle. The results demonstrated the impacts of the priority weighting factor on the delays of the bus and the general traffic. The results also showed the model has the ability to prevent accidental misuses of priority levels that are too high to cause the intersection oversaturation.

A comparison analysis was performed to compare the proposed control model SMINP with the transit signal priority strategy implemented in the RBC-TSP in VISSIM. Both control models were compared with the fix-time-do-nothing approach using the same hypothetical intersection. Two arrival headways (i.e., bus headway = 5, 10 minutes) were tested under three degrees of saturation conditions (i.e., V/C = 0.5, 0.7, 0.9). The results showed the SMINP rendered as much as a 30 percent improvement of bus delay from the TSP logic used in the RBC controller in low

to medium congestion conditions. The results also indicated that the SMINP model can recognize the level of congestion of the intersection and automatically give less priority to the bus so as to maintain a minimum impact to the traffic on conflicting phases.

A second comparison analysis was performed to investigate the performance of SMINP when multiple conflicting buses arrive at the same time. A rolling optimization scheme is developed so that optimizations can be performed not just once but multiple times in an incremental manner. Two different bus line conflict scenarios were tested under three degrees of saturation conditions (i.e., V/C = 0.5, 0.7, 0.9). One conflict scenario considered two intersecting bus lines, while the other scenario considered three intersecting bus lines. The results indicated that SMINP handles multiple bus priority much better than the RBC-TSP. Especially when V/C and the number of conflicting bus lines were high, the RBC-TSP simply failed if no-priority recovery periods were not strictly enforced. On the contrary, the SMINP automatically adjusts the relative importance of bus priority without the need to manually change the priority weighting factor, and it provides more balanced timings for both bus and general traffic. This further showed that it was reasonable and practical to use degree of saturation to approximate the impact of bus priority to other traffic.

6.2. FUTURE DIRECTIONS

Many directions can be explored based on the formulation proposed in this research. First, different real-time optimization schemes affect the optimality of the control strategies. As mentioned before, the formulation can result in an optimal timing for a two cycle planning period if all bus arrival times are known in advance even with uncertainty. However, the practical communications range between an OBU and an RSU is not enough to confidently predict all arrival times in advance for two cycles. Therefore, a rolling optimization scheme is more practical than a fixed-interval optimization scheme. Because the optimizations are conducted separately for all different buses in an incremental fashion, the global optimum solution is not guaranteed. Comprehensive numerical experiments will show how bad the rolling optimization is compared to the fixed-interval optimization method, and will give insights to future development of not just signal priority systems but also adaptive signal control systems.

Another important next step is to extend the formulation to enable bus priority along a coordinated corridor. It has been shown that when the degree of saturation and the number of

conflicting bus lines increase, the usable slack time becomes so small that it is difficult for SMINP to give priority to buses without negatively impacting general traffic. Instead of trying to squeeze out a few seconds for multiple priorities at one intersection, it may be easier to use multiple intersections to distribute the needs for priorities. Therefore, systematic planning that considers multiple intersections can be beneficial in signal priority along an arterial.

This planning has to take into account the relative importance of each bus at different intersections. For example, some intersections are naturally more congested than others, and giving absolute priority to buses in these intersections means serious disruptions to other traffic. So instead of answering the question of how to satisfy certain priority demand at one intersection, it would be better to answer how much priority of each intersection along the corridor should be provided to aggregately satisfy certain priority demand. Developing a mathematical program is a natural choice to systematically develop an optimal signal timing plan for minimizing operational costs at multiple intersections.

A necessary consequence of extending the stochastic formulation to multiple intersections is the exponential increase in the number of stochastic scenarios. The SMINP will become a large scale mathematical programming problem, and its deterministic equivalent program may not be an efficient way to solve for an optimal timing. A branch-and-cut algorithm based on disjunctive decomposition technique (*Ntaimo and Sen 2007*) may be needed to provide optimal solutions.

Another promising direction is to integrate the model with an adaptive signal control system where additional information about the development of vehicle queues at an approach can be estimated in real-time. The additional information relaxes the assumption about constant vehicle arrival and further improves the ability of the SMINP to predict the arrival time distribution of the bus to the stop bar. Theoretically, it can provide a best expected timing under uncertain conditions.

7. REFERENCES

Andrews, S. and Cops, M. (2009), *Vehicle Infrastructure Integration Proof of Concept - Vehicle*. Report FHWA-JPO-09-003, FHWA-JPO-09-017, FHWA-JPO-09-043, US DOT Research and Innovative Technology Administration, Washington, D.C.

Balke, K. (1998), Development and Laboratory Testing of an Intelligent Approach for Providing Priority to Buses at Traffic Signalized Intersections *Doctor of Philosophy*, The Texas A&M University, College Station, TX.

Beale, E. (1955), On Minimizing a Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 173-184.

Birge, J. R. and Louveaux, F. V. (1997), *Introduction to Stochastic Programming*, Springer.

Box, S. and Waterson, B. (2012), An Automated Signalized Junction Controller That Learns Strategies from a Human Expert. *Engineering Applications of Artificial Intelligence*, 25(1), pp. 107-118.

Cathey, F. W. and Dailey, D. J. (2003), A Prescription for Transit Arrival/Departure Prediction Using Automatic Vehicle Location Data. *Transportation Research Part C: Emerging Technologies*, 11(3–4), pp. 241-264.

Charnes, A., Cooper, W. W. and Symonds, G. H. (1958), Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil. *Management science*, 4(3), pp. 235-263.

Chien, S. I.-J., Ding, Y. and Wei, C. (2002), Dynamic Bus Arrival Time Prediction with Artificial Neural Networks. *Journal of Transportation Engineering*, 128(5), pp. 429-438.

Chin-Woo, T., Sungsu, P., Hongchao, L., Qing, X. and Lau, P. (2008), Prediction of Transit Vehicle Arrival Time for Signal Priority Control: Algorithm and Performance. *Intelligent Transportation Systems, IEEE Transactions on*, 9(4), pp. 688-696.

Christofa, E. and Skabardonis, A. (2011), Traffic Signal Optimization with Application of Transit Signal Priority to an Isolated Intersection. *Transportation Research Record: Journal of the Transportation Research Board*, 2259(-1), pp. 192-201.

Conrad, M., Dion, F. and Yagar, S. (1998), Real-Time Traffic Signal Optimization with Transit Priority: Recent Advances in the Signal Priority Procedure for Optimization in Real-Time

Model. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1634, TRB, Washington, D.C., pp. 100-109.

Dailey, D., Maclean, S., Cathey, F. and Wall, Z. (2001), Transit Vehicle Arrival Prediction: Algorithm and Large-Scale Implementation. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1771, TRB, Washington, D.C., pp. 46-51.

Danaher, A. R. (2010), *Tcrp Synthesis 83: Bus and Rail Transit Preferential Treatments in Mixed Traffic*. Report, Transportation Research Board, Washington, D.C.

Dantzig, G. B. (1955), Linear Programming under Uncertainty. *Management science*, 1(3-4), pp. 197-206.

Ecolite Control Products (2009), Transit Signal Priority (Tsp) User Guide for Advanced System Controller. Econolite Control Products, Inc.

Evans, H. and Skiles, G. (1970), Improving Public Transit through Bus Preemption of Traffic Signals. *Traffic Quarterly*, 24(4), pp. 531-543.

Ferguson, A. R. and Dantzig, G. B. (1956), The Allocation of Aircraft to Routes—an Example of Linear Programming under Uncertain Demand. *Management science*, 3(1), pp. 45-73.

Furth, P. G. and SanClemente, J. L. (2006), Near Side, Far Side, Uphill, Downhill: Impact of Bus Stop Location on Bus Delay. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1971, TRB, Washington, D.C., pp. 66-73.

Gartner, N. H. (1982), Prescription for Demand-Responsive Urban Traffic Control. *Transportation Research Record: Journal of the Transportation Research Board*, No. 881, TRB, Washington, D.C., pp. 73-76.

Gartner, N. H., Tarnoff, P. J. and Andrews, C. M. (1991), Evaluation of Optimized Policies for Adaptive Control Strategy. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1324, TRB, Washington, D.C., pp. 105-114.

He, Q. (2010), Robust-Intelligent Traffic Signal Control within a Vehicle-to-Infrastructure and Vehicle-to-Vehicle Communication Environment. *Doctor of Philosophy*, University of Arizona, Tucson, AZ.

He, Q., Head, K. L. and Ding, J. (2012), Pamscod: Platoon-Based Arterial Multi-Modal Signal Control with Online Data. *Transportation Research Part C: Emerging Technologies*, 20(1), pp. 164-184.

Head, K. L., Mirchandani, P. B. and Sheppard, D. (1992), Hierarchical Framework for Real-Time Traffic Control. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1360, TRB, Washington, D.C., pp. 82-88.

Head, L., Gettman, D. and Wei, Z. (2006), Decision Model for Priority Control of Traffic Signals. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1978, TRB, Washington, D.C., pp. 169-177.

Hunt, P. B., Robertson, D. I., Bretherton, R. D. and Winton, R. I. (1982), *A Traffic Responsive Method of Coordinating Signals*. Report Report No. LR1014, Transport and Road Research Laboratory, Crowthorne, Berkshire, England.

Jackson, D., W., Z., M., Peirce, S. and Baltes, M. (2008), Urban Partnership Proposals: Review of Domestic and International Deployments and Transit Impacts from Congestion Pricing. *the 87th Annual Meeting of the Transportation Research Board*. Washington, D.C.

Kittleson & Associate, KFH Group Inc, Parsons Brinckhoff Quade & Douglass and Zaworski, K. H. (2003), *Transit Capacity and Quality of Service Manual, 2nd Edition*. Report, Transportation Research Board, Washington, D.C.

Koonce, P., Rodegerdts, L., Lee, K., Quayle, S., Beaird, S., Braud, C., Bonneson, J., Tarnoff, P. and Urbanik, T. (June 2008), *Traffic Signal Timing Manual*. Report FHWA-HOP-08-024, Federal Highway Administration, http://ops.fhwa.dot.gov/arterial_mgmt/tstmanual.htm, Washington, D.C.

Li, M., Yin, Y., Zhang, W.-B., Zhou, K. and Nakamura, H. (2011), Modeling and Implementation of Adaptive Transit Signal Priority on Actuated Control Systems. *Computer-Aided Civil and Infrastructure Engineering*, 26(4), pp. 270-284.

Ling, K. and Shalaby, A. (2004), Automated Transit Headway Control Via Adaptive Signal Priority. *Journal of Advanced Transportation*, 38(1), pp. 45-67.

Ma, W., Liu, Y. and Yang, X. (2012), A Dynamic Programming Approach for Optimal Signal Priority Control Upon Multiple High-Frequency Bus Requests. *Journal of Intelligent Transportation Systems*.

Mauro, V. and Taranto, C. D. (1990), Utopia. *Proceedings, 6th IFAC/IFIP/IFORS S*yposium on *Control, Computers, Communications in Transportation*., Paris, September 1989, pp.245-252.

Ntaimo, L. and Sen, S. (2007), A Branch-and-Cut Algorithm for Two-Stage Stochastic Mixed-Binary Programs with Continuous First-Stage Variables. *International Journal of Computational Science and Engineering*, 3(3), pp. 232-241.

PTV America. (2010), Ring Barrier Controller User Manual. PTV America, p. 77.

Qiwu, R. and Jianguo, Y. (2012), A Novel Closed-Loop Feedback Traffic Signal Control Strategy at an Isolated Intersection. *Information Science and Technology (ICIST), 2012 International Conference on*, pp. 96-100.

Sen, S. and Head, K. L. (1997), Controlled Optimization of Phases at an Intersection. *Transportation Science*, 31(1), pp. p. 5-17.

Sims, A. G. and Dobinson, K. W. (1980), The Sydney Coordinated Adaptive Traffic (Scat) System Philosophy and Benefits. *IEEE Transactions on Vehicular Technology*, VT-29(2), pp. 130-137.

Smith, B. L., Venkatanarayana, R., Park, H., Goodall, N., Datesh, J. and Skerri, C. (2010), *Intellidrive^{sm} Traffic Signal Control Algorithms: Task 2: Development of New Traffic Control Signal Algorithms under Intellidrive^{sm}* Report, Charlottesville, Virginia.

Smith, H. R., Hemily, B. and Ivanovic, M. (2005), *Transit Signal Priority (Tsp): A Planning and Implementation Handbook*. Report, United States Department of Transportation, Washington, D.C.

Society of Automobile Engineers. *Dedicated Short Range Communications (Dsrc) Message Set Dictionary*, 2009. http://standards.sae.org/j2735_200911.

Stevanovic, J., Stevanovic, A., Martin, P. T. and Bauer, T. (2008), Stochastic Optimization of Traffic Control and Transit Priority Settings in Vissim. *Transportation Research Part C: Emerging Technologies*, 16(3), pp. 332-349.

Tlig, M. and Bhouri, N. (2011), A Multi-Agent System for Urban Traffic and Buses Regularity Control. *Procedia - Social and Behavioral Sciences*, 20(0), pp. 896-905.

Vasudevan, M. (2005), Robust Optimization Model for Bus Priority under Arterial Progression. *Ph.D.*, College Park, MD.

Yagar, S. and Han, B. (1994), A Procedure for Real-Time Signal Control That Considers Transit Interference and Priority. *Transportation Research Part B: Methodological*, 28(4), pp. p. 315-331.

Yin, K., Zhang, Y. and Wang, B. X. (2010), Analytical Models for Protected Plus Permitted Left-Turn Capacity at Signalized Intersection with Heavy Traffic. *Transportation Research Record: Journal of the Transportation Research Board*, 2192(1), pp. 177-184.

Zlatkovic, M., Stevanovic, A. and Martin, P. T. (2012), Development and Evaluation of Algorithm for Resolution of Conflicting Transit Signal Priority Requests. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2311, TRB, Washington, D.C., pp. 167–175.