ABSTRACT

KIM, SANGKEY. Dynamic Bandwidth Optimization for Coordinated Arterial. (Under the direction of Dr. Billy M. Williams and Dr. Nagui M. Rouphail.)

Urban traffic congestion problems continue to be an important focus of fundamental and applied research. Economic vitality and environmental sustainability will be directly impacted by the quality of the innovative solutions brought forth to address issues of travel delay and wasted fuel consumption. Signalized urban arterials are a key component of overtaxed urban transportation networks.

A commonly used strategy for improving mobility along signalized arterials is coordination of neighboring intersections to minimize stops by maximizing the duration of green bands, otherwise known as arterial bandwidth. Signal coordination has been researched, developed, and refined for five decades. In contrast to traditional methods based on analysis of programmed green times, a dynamic bandwidth analysis method is presented that reproduces actual dynamic bandwidth durations using closed loop signal data to assess the performance of semi-actuated coordinated arterial streets.

In this study, the author introduces 1) an enhanced MAXBAND formulation, 2) dynamic bandwidth characteristics, and 3) dynamic bandwidth optimization method.

An enhanced MAXBAND formulation is introduced to overcome the potential for the MAXBAND family of bandwidth optimization mixed-integer linear programs (MILPs) to yield suboptimal solutions under certain conditions. Using hypothetical network scenarios, the MAXBAND over constraint problem is revealed, and the proposed enhanced formulation is tested. Optimization performance of the original and enhanced formulations are compared using two hypothetical test network scenarios and the real world arterial used in one of the

seminal MAXBAND papers. The test result supports that the new formulation is consistent with the original MAXBAND formulation in cases where over constraint and suboptimality are not an issue and that the new formulation provides an optimal solution in cases where the original formulation performs suboptimality due to over constraint of the bandwidth ratio.

Although the newer MILP formulations, MULTIBAND, MULTIBAND-96, and AM-BAND, include increasingly innovative objective functions and constraints compared to the original MAXBAND MILP, all the MAXBAND-related models have similar constraints incorporating the directional bandwidth target value. Therefore, the modified objective function and constraints in the new formulation can be used to revise the formulations for all of the MAXBAND family of optimization models such as MULTIBAND, MULTIBAND-96, and AM-BAND.

In addition, this study shows real world dynamic bandwidth characteristics. Detailed analysis at three arterial sites revealed that coordinated green time distributions are complex and multi-modal and cannot be represented by a single statistic. Dynamic bandwidth analysis confirmed that programmed green bandwidth consistently underestimates the size of the actual dynamic bandwidth, and exhaustive search results highlighted the potential for further improvements in coordination.

Finally, the author provides dynamic bandwidth optimization methods for improving arterial coordination quality. The provided methods are 1) exhaustive search method by developed LP and 2) linear programming. The exhaustive search method can provide the complete feasible solution space so it is possible to easily understand each offsets coordination quality. LP formulation is developed to enable dynamic bandwidth maximization on semi-actuated arterial streets. The methodology relies on the use of

archived signal log data regarding phase start and end times in each cycle in both directions. Those data are used to optimize the offsets that maximize the variable system bandwidth across multiple cycles constituting a coordination plan period. The formulation can also be used to optimize signal offsets using programmed (fixed) green durations at each intersection. The proposed formulation offers five significant enhancements compared to traditional methods. The optimal offset solutions are tested in the field, and the results support the effectiveness of provided dynamic bandwidth optimization method.

Dynamic Bandwidth Optimization for Coordinated Arterial

by
Sangkey Kim

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Civil Engineering

Raleigh, North Carolina

2014

APPROVED BY:

_____	_____
Billy M. Williams, Associate Professor Committee Co-Chair	Nagui M. Rouphail, Professor Committee Co-Chair
_____	_____
George List, Professor	Yahya Fathi, Professor

DEDICATION

To my family, my lovely wife, Mijoung, and my two children, Dana and Kelly.

Without their support and love, none of this would be possible. God blesses my family.

BIOGRAPHY

Sangkey Kim was born in Andong, South Korea on February 19, 1977. He entered Chung-Ang University in 1997 and earned a bachelor's degree in civil engineering. Mr. Kim studied urban engineering as a master's student. He focused on the transportation planning area especially future demand forecasting and traffic assignment methods. After earning his master's degree, he worked for the Korea Transport Institute (KOTI) for two years. As a transportation planning researcher, Mr. Kim conducted several of pre-feasibility studies for freeway, railroad and subway system construction plan evaluation for the Korean government. After two years of study and experience, in July 2010 he began the journey to pursue his doctorate of philosophy under the guidance of Dr. Rouphail and Dr. Williams at North Carolina State University in Raleigh, North Carolina.

ACKNOWLEDGMENTS

First and foremost, I would like to express gratitude to my advisors, Dr. Nagui Rouphail and Dr. Billy Williams, for their guidance and counsel throughout my research and preparation for this dissertation. They gave me invaluable lessons in life and research as well as continuous support for my research during past four and one half years.

I would also like to express my appreciation to Dr George List and Dr Yahya Fathi who were willing to be my committee member and gave precious advice for my dissertation.

I would also like to thank my friend Dr. Ali Hajbabaie who provided me with advice and helped guide my research. His helping attitude and devotion supported me all the time.

I also appreciate my friend, Chang Beak, who is a signal engineer with NCDOT. He taught me from A to Z about the NCDOT signal system. Without his help, I could not have accomplished this dissertation.

Last but not least, my work could not have been done without the support from the NCDOT under Project NCDOT RP-2012-12.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Background

According to the Texas Transportation Institute's 2012 *Annual Urban Mobility Report*, 1982 congestion costs were an estimated $24 billion (2011 dollars) resulting from 1.1 billion delay hours and 0.5 billion gallons of wasted fuel. The report further illustrates that these costs increased dramatically through 2011 when total congestion costs were an estimated $124 billion resulting from 5.5 billion delay hours and 2.9 billion gallons of wasted fuel (Shrank 2007). Furthermore, the 2007 Traffic Signal Operation Self-Assessment Survey reported that *Signal Operation in Coordinated Systems* was given a "D-" grade, indicating that many signalized urban streets experience heavy congestion in the peak periods (National Traffic Signal Report Card 2007), and while the 2012 survey did not report a separate grade for operation in coordinated systems, the grade for Signal Timing Practices only improved from a "C-" to a "C" between the 2007 and 2012 surveys (National Traffic Signal Report Card 2012).

Unlike freeway facilities, where delay results primarily from capacity constrained bottlenecks, delay along signalized arterial whose demand does not exceed capacity results from deceleration and stops as vehicles interact with signal control. Therefore mitigating delay along signalized arterials through improved signal timing is a cost effective congestion management approach that can be accomplished without adding physical roadway capacity.

For more than a decade, the United State Department of Transportation has supported the development of adaptive and traffic responsive control systems, and new traffic adaptive

algorithms such as Optimized policies for Adaptive Control (OPAC) and Real-Time Hierarchical Optimized Distributed and Effective System (RHODES) have emerged (Gartner 1983, Head et al. 1992). Also, the United Kingdom and Australia have developed Split Cycle Offset Optimized Technique (SCOOT) and Sydney Coordinated Adaptive Traffic System (SCATS) (Hunt et al. 1981, Lowrie et al. 1992). The objective of developing these algorithms and systems is to help signals operate more efficiently.

However, there are several realistic limits to implementing adaptive signal control. Adaptive signal control operations require deploying several detectors to collect movement-specific traffic data. From that data, parameters are generated which require calibration by each system to estimate individual vehicle movements (travel time). Therefore, adaptive control systems typically require more installation and maintenance cost compared to traditional signal systems such as a closed loop system. In addition, it is necessary to train traffic signal engineers to operate adaptive systems which require extensive amounts of time and money.

One of the most common types of signal systems is the closed loop system which is a distributed traffic control system. This system consists of three components: the local controller, the on-street master controller, and the office computer. The systems have an actuated control capability with three different modes: “time of day”, “manual”, and “traffic responsive”. The limitation of the closed loop system is that signal timing, especially offset setting, relies on the operator’s engineering experience and judgment. Even if these systems have the capability to archive signal and detector log files (which may contain important

information for improving signal operation), still the offsets have to be entered into the system.

Well-designed signal coordination along arterial streets minimizes the number of stops and, consequently, travel delay. Synchronizing the onset of green indication for the intersections along an arterial street is one of the key steps in improving coordination and is known as offset optimization. Several studies, as will be discussed in the literature review section below, have addressed offset optimization assuming fixed-time signal-timing parameters. However, coordinated phase green times at intersections of an arterial under semi-actuated control have dynamic rather than static durations. This is true because semi-actuated control allocates green time to non-coordinated phases only as needed based on detector calls, thereby reserving any unused green from non-coordinated phases for additional green time for the coordinated movements. In other words, the controller can skip or terminate specific non-coordinated phases based on demand and allocate the unused green time to the major street coordinated movements. This phenomenon in semi-actuated, coordinated control is called “early return to green”. Also, if all side-streets and major street left turn phases are skipped, then the major street green will be extended throughout the cycle. Therefore, since the duration of the coordinated phase green indication is dynamic, the resulting coordinated bandwidth is also dynamic, varying from cycle to cycle.

In addition, each intersection’s early return to green and green extension is independent of upstream and down-stream intersections, depending only on the local demand. Furthermore, vehicle demands for non-arterial cross streets typically exhibit random arrival patterns. Therefore, the resulting dynamically varying arterial bandwidth must be

analyzed by observing cycle by cycle phase durations. These characteristics of semi-actuated signal control make the coordination of signals along an arterial a challenging task if dynamic bandwidth is to be taken into consideration.

However, this challenge brings the opportunity of improved understanding and improved arterial coordination. Analysis of dynamic bandwidth for current timing plans can provide valuable information to signal system managers and operators by assessing the impact of dynamic, coordinated phase green on dynamic arterial bandwidth. Furthermore, methodologies to optimize offsets along semi-actuated arterials based on observed phase durations hold the promise of improving signal coordination compared to optimization methods based on static programmed phase times. Improved arterial coordination will reduce traffic congestion, travel time, and the number of stops on arterial streets, thereby playing a role in addressing the chronic urban congestion problem highlighted above.

1.2 Problem Statement

Bandwidth is defined as the total amount of time per cycle available for vehicles to travel through a system of coordinated intersections at the progression speed, i.e. the time difference between the first and last hypothetical trajectory that can travel through the entire arterial at the progression speed without stopping.

Intersection offsets have a direct impact on the bandwidth magnitude. Coordinated intersection offsets are the decision variables that are typically optimized to maximize bandwidth. The majority of bandwidth optimization studies published in the signal

coordination literature were conducted using programed (fixed) green times. However, most coordinated arterial signal systems operate in semi-actuated mode.

One approach to dealing with varying green time would be to use the observed average coordinated green durations. However, field studies show that the distribution of coordinated green times are typically asymmetrical. Therefore, using average green durations for developing a semi-actuated control strategy will not guarantee finding a near optimal set of offsets. In practice, signal engineers traditionally conduct field visits to observe the early return to green and the initial queue to fine tune the offsets and improve arterial performance based on engineering judgment.

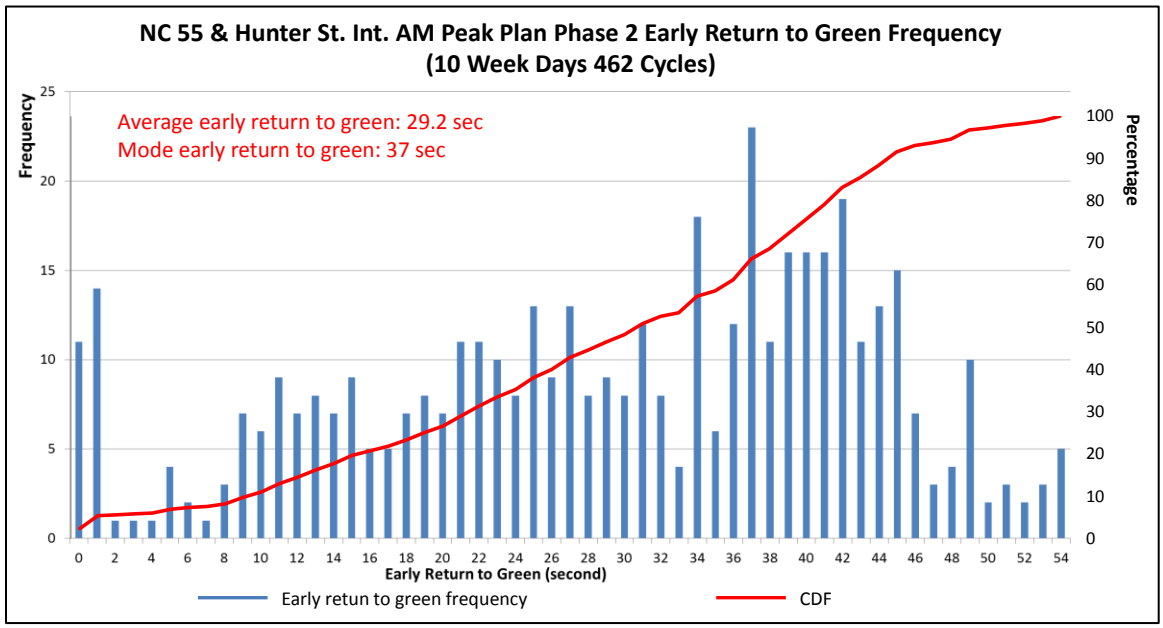


Figure 1-1 Intersection Early Return to Green Distribution

Figure 1-1 shows the frequency of early returns to green for an intersection on a coordinated system along NC 55 in Apex, NC. Only 2.38% of 462 cycles have no early return to green. The average additional green by cycle is 29.2 seconds with a primary mode of 37 seconds. It is also apparent from Figure 1-1 that the early return to green has a multi-modal distribution. The research to date on early return to green distribution has focused on using a fixed number to represent the distribution in the offset optimization process. The limitation of this approach is that a fixed value such as average, median, or mode cannot fully represent the entire early return to green given its characteristically asymmetrical, multi-modal distribution.

As mentioned above, bandwidth is traditionally calculated based on allotted green time with this "worst case" bandwidth used to evaluate arterial performance. Using allotted green carries an implicit assumption that either there is no early return to green or that it can be effectively ignored. However, the observed early return to green shown in Figure 1-1 makes it clear that arterial performance evaluation based on a static bandwidth method may be significantly different from the actual experienced bandwidth, especially when demand for non-coordinated movements is low. As such, it is important to calculate arterial bandwidth using actual dynamic green times from a representative sample of cycles "over time." In fact, since the actual displayed green is longer than or equal to the allotted green for coordinated phases, the dynamic bandwidth is always greater than or equal to the static bandwidth. Therefore, using static bandwidth will in most cases underestimate arterial street green band capacity.

Furthermore, signal plan development traditionally relies on field manual fine-tuning. If the cycles observed during fine-tuning are not representative of normal conditions, this heavy reliance on limited observation fine-tuning could result in sub-optimal offsets. Finally, ongoing signal plan evaluation traditionally relies on costly field observations including before and after travel time runs. While this traditional approach can yield acceptable results, the process is time consuming, includes a relatively high degree of subjectivity, and by necessity yields decisions that are founded on very a limited sample of operational conditions.

In summary, no methodology currently exists to determine dynamic bandwidth on arterial streets using the signal operation data resident in deployed signal control systems, the primary surface street component of Advanced Transportation Management Systems (ATMS). This study aims to bridge this gap by developing a methodology to determine dynamic bandwidths for semi-actuated coordinated arterial streets.

1.3 Objective

The primary objective of this study is to develop a methodology to determine dynamic bandwidth using closed loop signal data to assess the performance of semi-actuated coordinated arterial streets. In addition, the study will show the benefits of optimizing dynamic bandwidth in arterial streets. This research is expected to 1) provide a dynamic bandwidth analysis algorithm and tool which can allow analysis of real bandwidth series from control system ATMS data sets and 2) suggest feasible methodologies to find the global optimal offset point. Overall, the aim of this study is aimed to contribute to the important

goal of continually improving signalized arterial performance. This research will focus on the following topics:

- Developing a dynamic bandwidth analysis algorithm and tool ;
- Exploring the difference between traditional bandwidth and dynamic bandwidth;
- Improving traditional bandwidth optimization method;
- Developing a new formulation for dynamic bandwidth optimization.

This research is expected to develop a dynamic bandwidth analysis algorithm and tool which can allow analysis of real bandwidth series from ATMS data sets and suggest feasible methodologies to find the global optimal offset point. Finally, this study will help to improve arterial performance, arterial signal management efficiency and give some important information to traffic engineers that can support better engineering judgment.

1.4 Contribution

This thesis presents enhanced MAXBAND formulation is introduced to overcome the potential for the MAXBAND family of bandwidth optimization mixed-integer linear programs (MILPs) to yield suboptimal solutions under certain conditions. Although the newer MILP formulations, MULTIBAND, MULTIBAND-96, and AM-BAND, include increasingly innovative objective functions and constraints compared to the original MAXBAND MILP, all the MAXBAND-related models have similar constraints incorporating the directional bandwidth target value. Therefore, the modified objective

function and constraints in the new formulation can be used to revise the formulations for all of the MAXBAND family of optimization models such as MULTIBAND, MULTIBAND-96, and AM-BAND.

In addition, this thesis presents a linear programming (LP) formulation to enable dynamic bandwidth maximization on semi-actuated arterial streets. The methodology relies on the use of archived signal log data regarding phase start and end times in each cycle in both directions. Those data are used to optimize the offsets that maximize the variable system bandwidth across multiple cycles constituting a coordination plan period. The formulation is also flexible to be used to optimize signal offsets using programmed (fixed) green durations at each intersection. The proposed formulation offers five significant enhancements compared to traditional methods.

- The formulation is strictly linear (complexity of class P) as opposed to the traditional mixed integer programming formulations (complexity of class NP-hard).
- It can work with either static or dynamic (cycle varying) green durations.
- Traditional bandwidth optimization methods have explicit constraints to enforce bandwidth allocation to be proportional to directional demand. However, as longs as the minimum required bandwidth is allocated to each direction, a proportional allocation of bandwidth is not necessary and in fact, can result in reporting sub-optimal solutions. The proposed formulation overcomes this shortcoming.

- The LP formulation enables the analyst to carry out a post-processing step which reports the range of offsets at non-critical intersections within which the optimal objective function value will be unaffected. This capability provides engineers with multiple solutions with identical bandwidths, enabling the consideration of other performance measures such as stops, delay, or fuel consumption and emissions. The proposed formulation predicts the maximum proportion of traffic demand that can be served in the bandwidth, a unique attribute absent from other formulations found in the literature.

1.5 Organization

This thesis consists of seven chapters. In the next chapter, a literature review on related topics is provided. In Chapter three, an enhanced MAXBAND formulation is described to overcome the potential for the MAXBAND family of bandwidth optimization mixed-integer linear programs (MILPs) to yield suboptimal solution. Chapter four describes field dynamic green data characteristics, selected study sites and dynamic bandwidth analysis tool (DBAT). Chapter five introduces dynamic bandwidth optimization method. In Chapter six, dynamic bandwidth optimization results on the study site is provided to compare before and after arterial travel time. Finally a summary of the research results, including findings, conclusions, and future recommendations, is provided in Chapter Seven.

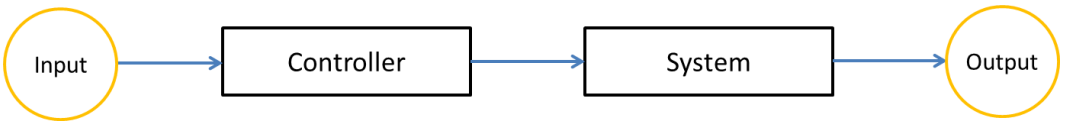
CHAPTER 2 LITERATURE REVIEW

For more than 100 years, traffic signals have served urban and suburban streets. Before traffic signals were developed, public officers such as policeman manually managed traffic movements. During past decades, many different computer based signal operation systems were developed for isolated intersection and coordinated arterials. This chapter reviews traditional offset optimization methods as well as various signal control system with arterial performance measures.

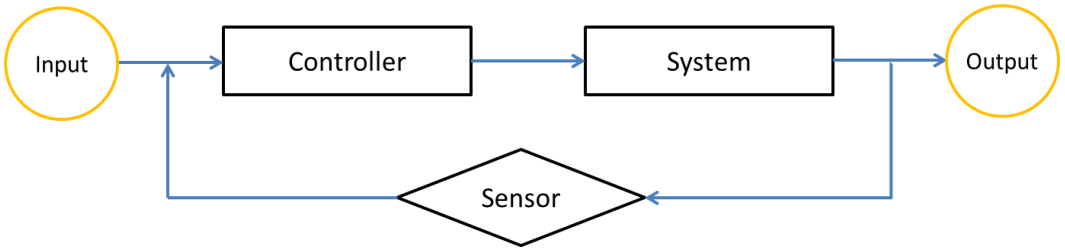
2.1 Signal System

2.1.1 Closed Loop System

The difference between open and closed loop control systems is that an open loop system refers to a system where the communication between the controller system and the output is one way. A closed loop system has a feedback system to monitor the output of system and it corrects the errors. An example would be as clock time drift between master intersection clock and local intersection clock. Figure 2-1shows simplified flow of two systems.



1.a Open Loop System



1.b Closed Loop System

Figure 2-1 Open and Closed Loop System Process

Old forms of signal controllers are called electro-mechanical signal controllers, which are mainly composed of movable parts (cams, dials, and shafts) that control the passage of green, yellow and red lights in a predetermined sequence. Timings were controlled by mechanical tabs on a dial that were manually adjusted in the field by traffic engineers. These old systems do not provide any feedback loop in the system. They also do not allow communication between the master signal and local signal controller. These old types of signal controllers are called “open loop systems”.

The closed-loop system is a distributed processor traffic-control system with control logic distributed among three levels (Traffic Signal Control System).

- The local controller
- The on-street master controller
- The office computer

These systems provide two-way communication between the local controllers, on-street master, and between the on-street master and the office computer. Typically, the local controller receives information from field detectors. The master controller receives information such as the status, time, and traffic volume from the local controllers. The office computer enables the system operator to monitor and control the system's operations.

Three control modes are typically found with most closed-loop systems (Traffic Signal Control System).

- Manual mode
- Time of day mode
- Traffic responsive mode.

Under the Manual Mode, the operator specifies the pattern number of the desired traffic-signal timing plans and sequence via computer console. The time of day mode allows the controller unit to automatically select and implement a predetermined traffic-signal timing plan such as cycle, offset and split based on the time of day. With the traffic responsive mode, the computer automatically selects the predefined traffic-signal timing plan. This is the best fit to accommodate the current traffic flow conditions in the signal network. The pattern selection and implementation is accomplished through a traffic flow data matching technique executed every five minutes.

There are four traffic signal controller types for operating traffic signal systems. These consist of pre-time (fixed time), fully actuated with non-coordinated controllers, semi-actuated, and fully-actuated with coordinated controllers. Fixed-time signal controllers are

still widely used in CBD areas or similar environments where demand is over capacity. This type of controller provides exactly the same amount of green time to phases during each cycle. Fully actuated control is most often applied to non-coordinated signal controllers at isolated intersections. All phases are actuated so that the green duration of each phases is decided by minimum and maximum green and passage time, and, if necessary, pedestrian phase settings. Each phase can be extended, gapped out, or even skipped depending on demand and system settings. Therefore, in fully-actuated control there is no fixed cycle length. Volume-density represents the most complex legacy implementation of fully-actuated control with variable initial green based on arrivals on red and passage (gap) time that is shortened during the phase extension.

In semi-actuated signal control, only non-coordinated phases are actuated. All of the unused green time for non-coordinated phases reverts to the coordinated phases. This signal control scheme guarantees a programmed bandwidth (minimum bandwidth) for coordinated movements. Fully-actuated coordinated systems are similar to semi-actuated signal controllers. They can allocate a portion of coordinated phases to non-coordinated phases. This means a portion of the coordinated phases are actuated (Day et al 2008).

A closed loop system coordinated system consists of six main components.

- System detectors
- Local control equipment
- Controller master communications
- On street master
- Master central communications

- Office computer.

Among each component, the office computer allows the operator (traffic engineer) to set the time and date, display intersection, modify the master database, modify controller and coordination settings, modify system parameters, and monitor the system.

2.1.2 Traffic Responsive System and Adaptive Traffic Control System

Traffic responsive systems manage local controllers by updating cycles, offsets, and splits based on network level sensing. For example, traffic responsive plan selection (TRPS) involve matching defined plans to current traffic levels with this evaluation and selection normally taking place either in a field master controller or a central computer (Traffic Signal Timing Manual 2013). When the master or computer selects a new timing plan due to demand changes, it sends a command to all local signal controllers in a coordinated group instructing them to change to the new plan simultaneously. The master or central computer monitors multiple traffic condition data from an array of detectors, including data such as volume and occupancy. The detector data is processed to calculate values for a few key parameters that are compared to predetermined thresholds. When the thresholds are crossed, the most applicable timing plans from within the predetermined plans is implemented for the conditions represented by the threshold categories selected.

In contrast to traffic responsive plan selection, which is by design reactive to measured traffic conditions, real-time adaptive traffic signal control system is a concept where vehicular traffic in a network is detected at system detectors, and then various algorithms are used to predict when and where the traffic will be in the future. This predictive feature allows the control system to make signal adjustments at appropriate

intersections to serve the predicted traffic flows. The signal controller utilizes prediction algorithms to compute optimal signal timings based on detected traffic volume and simultaneously implement the timings in real-time.

Miller introduced the principle of adaptive control for an online traffic modeling strategy (Miller 1963). The model calculates time wins and losses and combines these criteria for the different stages in the performance measures to be optimized. The first adaptive control system (PLIDENT), was implemented in Glasgow, United Kingdom (UK) in the 1960s. However, the system did not operated effectively. (Holroyd and Hillier 1971). The second adaptive control system field trial was in Canada (Corporation of Metropolitan Toronto 1976), but that trial also failed due to an inaccurate demand prediction algorithm for a longer time period, slowing the reaction of transition programs.

In the 1967, US Federal Highway Administration (FHWA) launched the Urban Traffic Control System (UTCS) project (Macgowan and Fullerton 1979). The stated objectives of the UTCS projects were (Stockfish 1984):

- To develop and test, in the real world, new computer based control strategies that would improve traffic flow.
- To document System planning, design, installation, operation, and maintenance to assist traffic engineers with installing their own systems.
- To stimulate modernization of traffic control equipment.

The UTCS project identified three generations of adaptive control systems and the plan was to demonstrate and evaluate each generation of controls. The three generations are (Stockfish 1984):

- The first generation (UTCS-1) uses a library of predetermined timing plan, each developed with off-line optimization programs. The plan selected can be based on time of day, measured traffic pattern, or operator specification. The update period is 15 minute intervals. First generation allows critical intersection control and has a bus priority system (Raus 1975, Macgowan and Fullerton 1979).
- The second generation (UTCS-2) uses timing plans computed in real time, based on predicted traffic conditions, using detector observations input into a prediction algorithm.
- The third generation was conceived as a highly responsive control with a much shorter control period than second generation and without the restriction of a cycle based system. Third generation system included a queue management control at critical intersections.

Table 2.1. Comparison of Key Features of Three Generation of Control System

Feature	First Generation	Second Generation	Third Generation
Optimization Frequency of Update	Off-line 15 minutes	On-line 5 minutes	On-line 3-6 minutes
No. of Timing Pattern	Up to 40	Unlimited	Unlimited
Traffic Prediction Critical Intersection Control	No Adjusts split	Adjusts split and offset	Adjusts split, offset and cycle
Hierarchies of Control	Pattern selection	Pattern computation	Congested and medium flow
Fixed Cycle Length	Within each section	Within variable groups of intersection	No fixed cycle length

Source: Traffic Engineering (Mcshane et al. 1990)

The UTCS-1 and UTCS-2 systems were installed in Washington, D.C. to develop, test and evaluate advanced traffic control strategies (The Urban Traffic Control System in Washington 1974). The first generation UTCS-1 system was applied in New Orleans, controlling 60 intersections in an arterial environment. A time of day plan test resulted in an 8.8% reduction in travel time while the second generation UTCS-2 system test resulted in a comparable 8.5% reduction in travel time (FHWA RD-73-3 1978). In 1979, UTCS-1.5 was developed as an upgrade version of UTCS-1 (performance was not better than UTCS-1) (CCSAG Newsletter 1985). Like UTCS-1, timing plans were implemented from a library of pre-determined plans. However, UTCS-1.5 including the capability to generate new plans based on observed traffic data. These automatically generated plans would need to be reviewed and approved by a system operator before they could be added to the plan library. The UTCS-1.5 was tested in Broward County, Florida (1982), and Birmingham, Alabama

(1984). In May of 1985, The UTCS project concluded and the policy statements were distributed on the support for the UTCS-1.5. The UTCS policy statement indicated that FHWA would not further enhance the software or documentation and that the private sector will likely develop and maintain their own system.

NCHRP 403 describes readily available adaptive control systems (NCHRP 403 2010). Several adaptive traffic control systems (ATCS) are currently deployed in the United States.

- Sydney Coordinated Adaptive Traffic System (SCATS)
- Split, Cycle, Offset Optimization Technique (SCOOT)
- Automatic Traffic Surveillance and Control (ATSAC)
- Optimized Policies for Adaptive Control (OPAC)
- Real-Time Hierarchical Optimization Distributed Effective System (RHODES)
- Adaptive Control Software-Lite (ACS-Lite)
- InSync

2.1.2.1 SCATS

In the late 1970s, the Road and Traffic Authority in New South Wales, Australia developed SCATS (Sims et al 1980). SCATS generate cycles, offsets and splits in three separate heuristic processes using calculated Degrees of Saturation (DSs) and link flows (LFs) from detector data (Lowrie , 1982, Stevanovic et al. 2009). Cycle length generated depends on two scenarios called low volume scenarios and high volume scenarios. Under the low volume scenarios, cycle length is determined from LFs. The high volume scenarios provided cycle length is computed using DSs. SCATS does not use common cycle lengths for coordinated signals. Instead of using common cycle length, it considers quality of

progression and selectively joins together intersections that have good progression. For offset adjustments, SCATS uses a number of predetermined offset plans and seeks the best offset for particular flow patterns. However, SCATS uses only stop bar detectors, so it cannot create real-time volume (demand) profiles. On the other hand, SCATS' offset would not be sensitive against volume (demand) fluctuation.

2.1.2.2 SCOOT

The SCOOT is the most widely deployed adaptive system in existence. Figure 2-2 provides the SCOOT operation diagram. The system was developed in the United Kingdom (U.K.). SCOOT uses upstream system detectors to collect and create real-time "Cycle flow profiles" for each link. The SCOOT system detector location is different than the detector location for typical actuated control detection. Typical actuated detectors are located nearby the associated intersections. However, SCOOT system detectors are located at the upstream end of system approach links (Transportation Research Laboratory 2009).

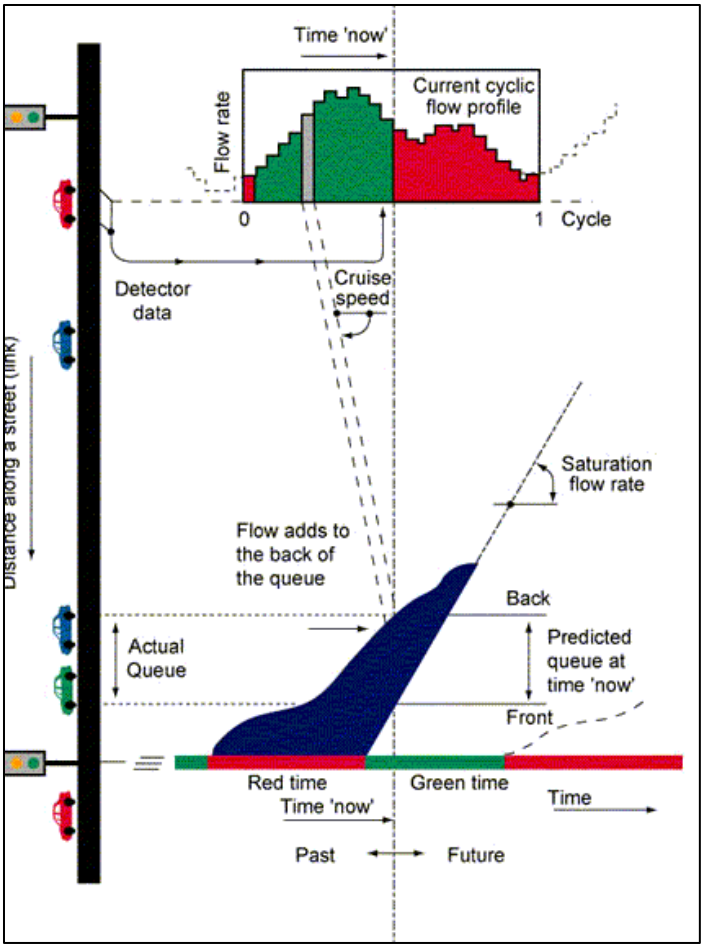


Figure 2-2 SCOOT Operation Diagram
Source: http://www.scoot-utc.com/DetailedHowSCOOTWorks.php

The upstream system detector data are processed by the SCOOT system predictive algorithms to create arrival profiles at the system intersections. SCOOT has three embedded optimization algorithms, one each for splits, offsets and cycles. SCOOT uses a common cycle length for coordinated arterial systems. These system cycle lengths are set at a length that maintains a degree of saturation less than predetermined value for all intersection within a given upper and lower boundary.

2.1.2.3 ATSAC

ATSAC was developed by Los Angeles Department of Transportation (LA DOT). ATSAC does not have any formal optimization logic (algorithms) for adjusting signal timing. It instead applies heuristic formulas based on extensive systems operation experience (Rowe 1991). The adaptive adjustment of signal timings are based on second-by-second fluctuation of volumes and occupancies measured at system detectors. Cycle lengths are adaptively updated within predetermined upper and low boundaries. For a given system, splits are adjusted under minimum green time consideration after the current cycle length is set. ATSAC does not provide alternate phase sequences.

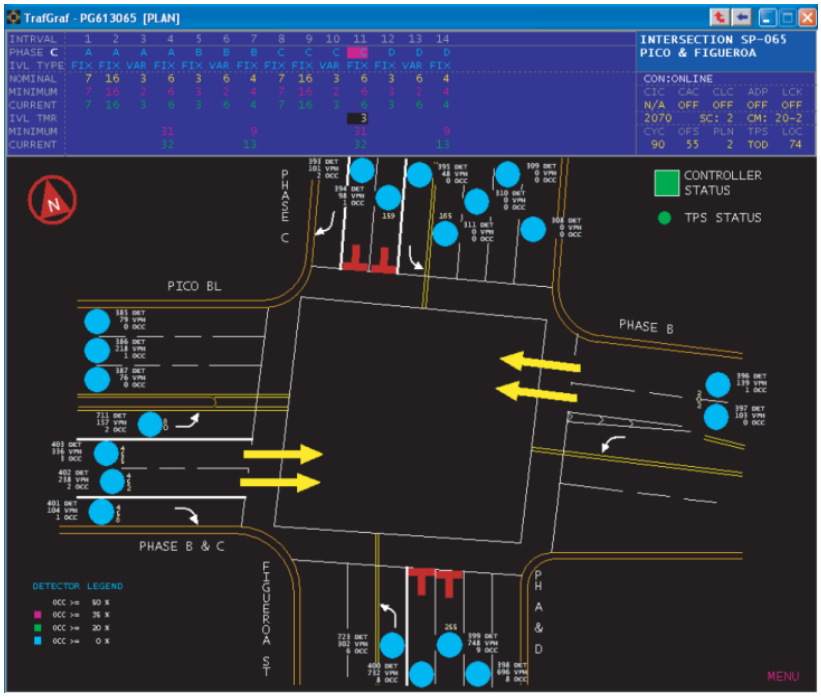


Figure 2-3 Dynamic Map Function in ATSAC
Source: NCHRP 403

2.1.2.4 OPAC

The Optimized Policies for Adaptive Control (OPAC) strategy utilizes a real-time signal timing optimization algorithm developed at the University of Massachusetts at Lowell (Gartner 2001). OPAC is a fully-adaptive, proactive, and distributed real time traffic control system. The system was developed as part of the FHWA Real-Time Traffic Adaptive Control System (RT-TRACS) program (Andrews and Elahi 1997). The fundamental features of OPAC system are:

- Optimization of any or all phases splits designed to minimize total intersection delay and/or stops
- Support for phase skipping in the absence of demand
- Multiple sets of configuration parameters for customizing the resulting timing to weight certain movements for special circumstance or by time of day
- Configurable to respond to changes in left turn lead/lag phasing by time of day
- Special considerations for phase timing in the presence of congestion

OPAC is different from traditional cycle-split signal control strategies in that it drops the concept of system cycle lengths (Liao 1998). In OPAC, the signal control algorithm consists of a sequence of switching decisions made at fixed time intervals. A decision is made at each decision point on whether to extend or terminate a current phase. Dynamic programming techniques are used to calculate optimal solutions.

2.1.2.5 RHODES

In 1992, the Real-Time Hierarchical Optimization Distributed Effective System (RHODES) was developed by the University of Arizona (Mirchanani and Head 2001). RHODES is a real-time traffic adaptive control system. It has a three-level hierarchical structure for characterizing and managing traffic and predicts traffic at these levels utilizing detector and other sensor information (Head et al. 1992). RHODES can receive and consider input from different types of detectors. Based on predicted future traffic conditions, RHODES generates optimized signal control plans. Figure 2-4 illustrates the hierarchy of the RHODES system.

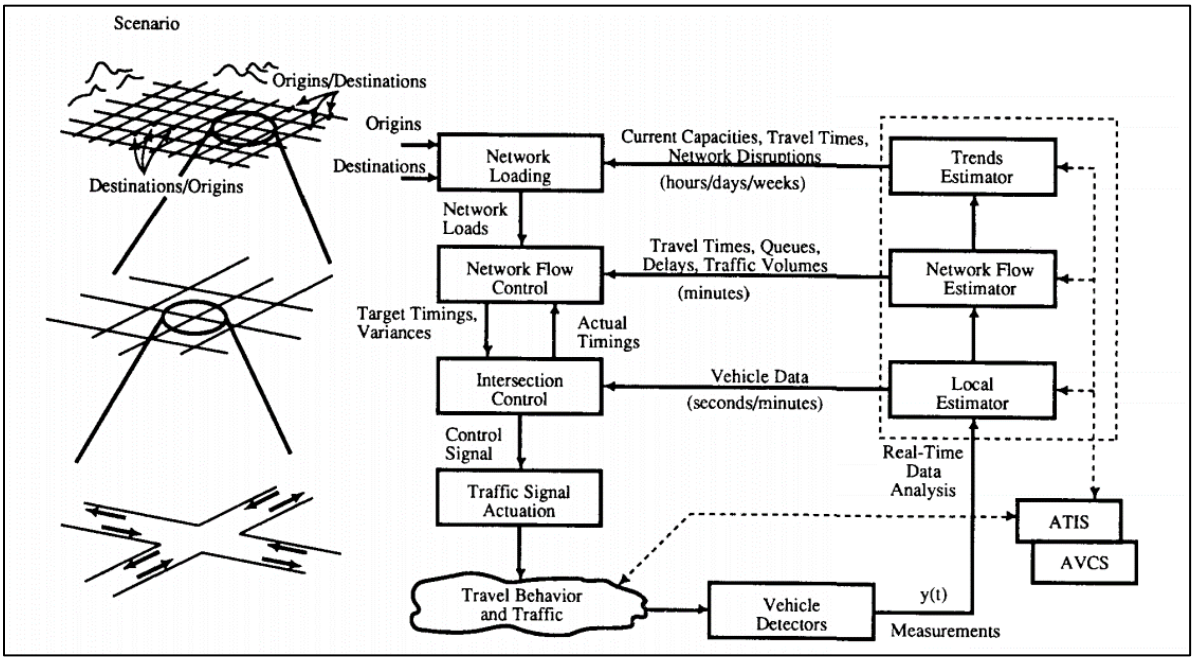


Figure 2-4 Functional Diagram of the RHODES Real-Time Traffic Control System (Head et al 2001)

RHODES uses a dynamic programming (DP) based real-time signal control systems similar to OPAC. However, RHODES uses signal phases as stages, the amount of green-time as decision variables, and the total number of time-steps as state variables. The RHODES DP formulation requires a fixed sequence of phases and a longer forecast horizon. Since the RHODES DP formulation requires a fixed sequence of phases, it cannot optimize phase sequences. RHODES uses the REALBAND algorithm for its signal coordination (Dell'Olmo and Mirchandani 1995). REALBAND constructs a decision tree which contains all the possible decisions from the identified conflict movements. Each path in the decision tree represents a set of conflict resolutions that can be made within the system. The system calculates each path's performance such as delay and uses the calculated performance with path combination as constraints in the optimization algorithm.

2.1.2.6 ACS-Lite

ACS-Lite was designed for closed loop system's operation in the late 1990s (Gartner et al. 2002). The system was developed by the University of Arizona, Purdue University, and private vendors such as Siemens and Econolite (Luyanda et al. 2003). This system is a reduced-scale version of the FHWA's adaptive control software. It offers small and medium-size communities a low-cost traffic control system that operates in real time, adjusting signal timing to accommodate changing traffic patterns and ease traffic congestion. Changes to cycle time are handled on a time of day plan like traditional traffic control systems. At each optimization step (which occurs approximately every 10 minutes), the system changes the splits and offsets a small amount to react traffic flow fluctuation.

ACS-Lite provides adaptive control within the industry standard context of cycle, splits, and offsets utilizing three control algorithms. Figure 2-5 shows that how the algorithms work in tandem to update traffic signal timing.

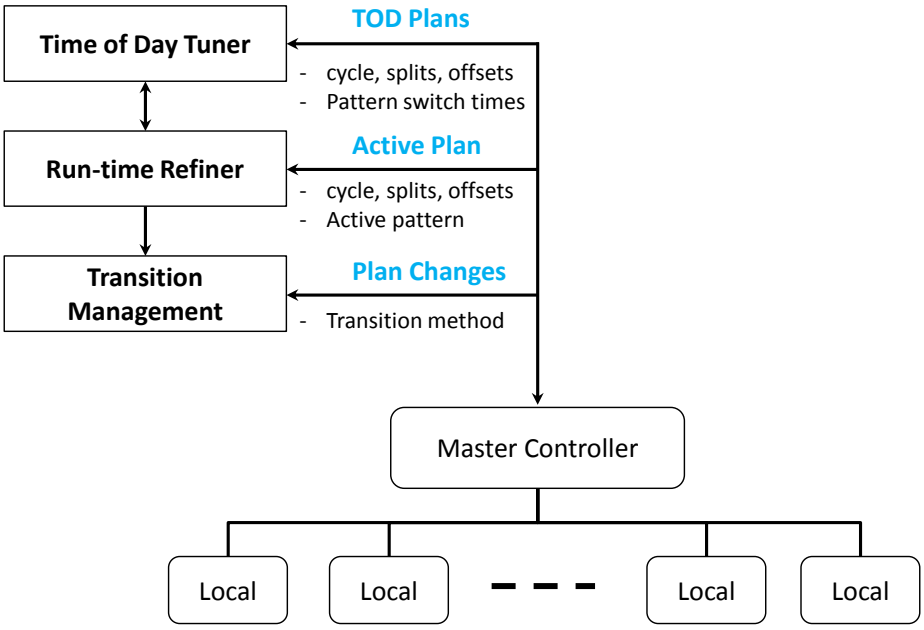


Figure 2-5 ACS-Lite Architecture (31)

2.1.2.7 InSync

The InSync system is an adaptive traffic signal system developed by Rhythm Engineering that uses advanced sensor technology, image processing, and artificial intelligence. The system uses a fundamentally different system of controlling and optimizing

signal phases and timings in real-time (no cycle length and phase sequence). InSync signal timing methodology includes three major components (NCHRP 403, 2010).

- Digital architecture
- Global optimization
- Local optimization

The "digital architecture" term refers to the concept of a "finite state machine". In other words, InSync considers all possible non-conflict movement pairs and creates a maximum of "x" possible sequences of phase pairs at all intersections. Through the finite state machine framework, the InSync system can call any non-conflicting movement pair at any time. Thus, there is no predetermined phase sequence. The controller transitions signal indications from one state to the next based on the InSync logic, encapsulated by the local and global optimization algorithms.

This local and global optimization framework defines a two level optimization process. For global optimization, the InSync system focuses on time dependent platoon movements. The global optimizer in the system works to group platoons and optimizes their progression by maximizes the likelihood that each intersection's coordinated phase will be green at that time each "time tunnel" (which has similar concept to "green band") reaches the intersection. Conventional arterial coordination requires plan-based system cycle lengths for all coordinated signals. However, Insync does not require common cycle length for coordination. There are also no intersection timing plans for phase sequencing, and therefore there is no transition time required between the "time tunnels." In essence, outside of the

time tunnels, each intersection runs its own local optimization (i.e. the “local optimizer”). The local optimizer allows each signal in the arterial to operate in an intelligent, fully actuated mode.

2.2 Performance Monitoring System

2.2.1 SMART-SIGNAL

In 2007, the University of Minnesota developed the Systematic Monitoring of Arterial Road Traffic and Signals (SMART-SIGNAL) system for monitoring arterial signal operation performance (Sharma et al. 2006, Sharma et al. 2008). Figure 2-6 shows overall architecture of SMART-SIGNAL.

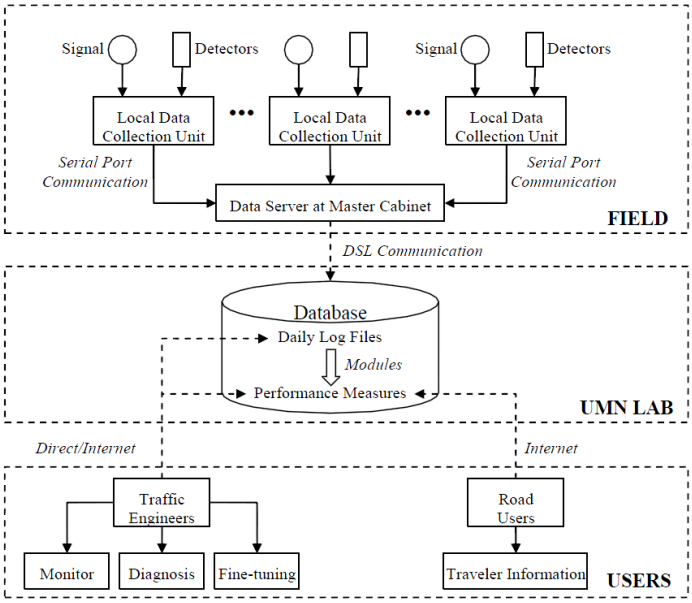


Figure 2-6 SMART-SIGNAL System Architecture (Sharma et al. 2008)

The SMART-SIGNAL system collects two kinds of event signal data from “DATA Collection System”, which are vehicle actuation events and signal phase change events. Collected high resolution vehicle events data are used for estimating turning movement percentages and queue length. The results of dynamic queue length estimation and signal status date are processed for measuring each intersection's performance. Furthermore, the system generates “virtual prove car” to estimate arterial travel time to measure arterial performance.

2.2.1.1 Data Collection System

General actuated signal control systems are operated by detector call and the operation results are displayed as signal phases. SMART-SIGNAL archives these two event data which are vehicle actuation events and signal phase change events. Those data sets are acquired separately from the data collection unit located in the traffic signal cabinet. SMART-SIGNAL uses Traffic Signal Timing Performance Measurement System (TSPMS) developed by TTI, as the data collection component at signal intersection (Balke et al. 2005). An industrial PC and a data acquisition card are deployed in each intersection's traffic signal cabinet to archive both events data. Collected data is transmitted to the data server in the master controller cabinet through the existing communication line. Figure 2-7shows the structure of data collection flow in SMART-SIGNAL. The SMART-SIGNAL traffic data collection flow shows two data categories which are the existing traffic signal data and the additional data collection process. Detector calls and signal status data are existing data groups and data is processed through a Traffic Controller Interface Device (CID) and a

Traffic Event Recorder software program. The processed data is archived in Traffic Log Data Server.

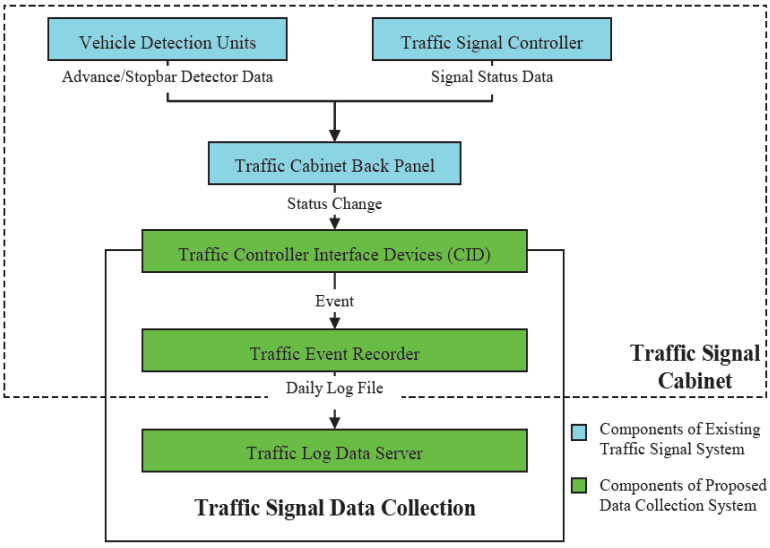


Figure 2-7 Traffic Data Collection Flow in SMART-SIGNAL (Balke et al. 2005)

2.2.1.2 Data Processing

The archived raw data is processed and converted to an easy to read format for measuring intersection and arterial performance. The data processing procedure provides high resolution detector actuation data (second-by-second) such as volume and occupancy. It also provides cycle-by-cycle signal timing with status data, indicating each phases green start and end time. Figure 2-8 shows the SMART-SIGNAL data processing flow chart. The data process begins after the raw data is transmitted back to server. The data processing flow includes four steps.

- Data Verification

- Preprocessing module
- Performance measure calculation
- Visualization

The data verification step tests the collected data quality and filters out wrong data. The preprocessing module generates some basic measures from the raw data. The performance measure calculation creates aggregated volume, delay, queue size, queue length, travel time etc. Finally the visualization step shows the results of different types of performance measures and diagnosis' fine tuning of traffics signal system.

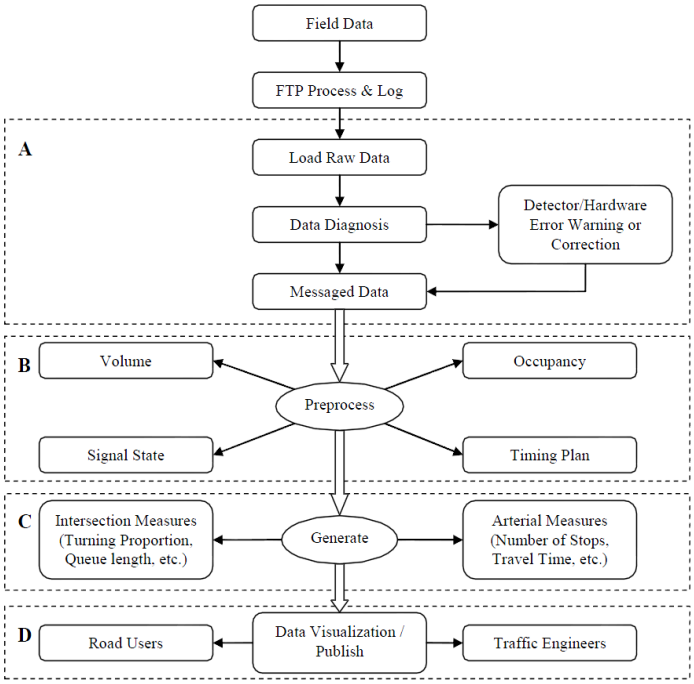


Figure 2-8 SMART-SIGNAL Data Process Flow Chart (Balke et al. 2005)

2.2.1.3 Intersection Performance Measurement

Many adaptive control systems use intersection delay and level of service as intersection performance measurements. In the SMART-SIGNAL system, queue length and turning movement proportions (TMP) are used for measuring intersection performance.

For the queue estimation, SMART-SIGNAL uses a dynamic queue length estimation method. The model calculates difference of the arrival and departure rate and provides a queue length over time. The SMART-SIGNAL queuing model defines a number of event times that describe the dynamics of queue interaction with the signal status. The queuing model includes two separate estimation models which include “Short Queue Estimation Model” and “Long Queue Estimation Model”. If vehicle arrivals can be measured from advance detectors and queue length is less than distance between stop line and detector location, it is defined as “short queue”. Otherwise, it is defined as “long queue”. Short queue can be estimated according to the queue development using advance detector calls, which can provide vehicle headway. When queue spills over the advance detector, the advance detector will be occupied by a car and it will provide hi-occupancy. SMART-SIGNAL developed the relationship of queue development and occupancy profile at a signalized intersection. Using that relationship, maximum queue length and time-dependent queue length curve are estimated. SMART-SIGNAL’s queue estimation procedure was compared to field data for 70 samples and was able to predict actual queue lengths with an average error of 7.5%, and queue sizes with an error of approximately 9.4%.

Turning movements’ proportion or counts are important information to analyze an intersection's performance, so it is often used as input data for simulation. However, it is

difficult to measure directly from detector calls since full-set detector configuration is rare in the field. Right-turn detectors are usually not deployed because a protect phase is absent at the majority of the intersections in the United States. Furthermore, shared lanes (through and right, or through and left) and exclusive left turn lanes with long loop detectors also make it difficult to measure turning movement proportions directly from detector calls.

SMART-SIGNAL proposed a simple turning movement proportion estimation model using advance detectors and left turn stop bar detectors for the major approach. Left-turn stop bar detectors were also used for the minor approach. Figure 2-9 shows SMART-SIGNAL detector configuration.

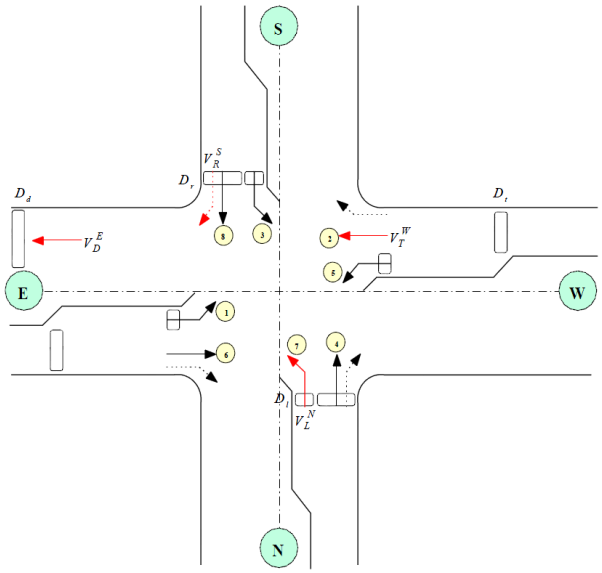


Figure 2-9 SMART-SIGNAL Detector Location Configuration (Balke et al. 2005)

To solve the turning movement proportion, two assumptions are made.

1. The travel time between each detector is known value and stable.

2. The right-turn movement traffic in a cycle is continuous and uniform.

Based on two assumptions, SMART-SIGNAL estimates the short time intersection turning movement proportion. The suggested model was tested, a total of 56 sample cycles and 85 percentiles have errors of less than 15% and the average error was 8.9%. However, the error rate can vary site-by-site due to the second assumption.

2.2.1.4 Arterial Performance Monitoring Using Virtual Vehicle

SMART-SIGNAL uses a virtual probe vehicle to estimate time-dependent arterial travel time, utilizing high resolution vehicle actuation data and signal status data. The manually generated virtual prove vehicle has three possible maneuvers including acceleration, deceleration and no-speed-change.

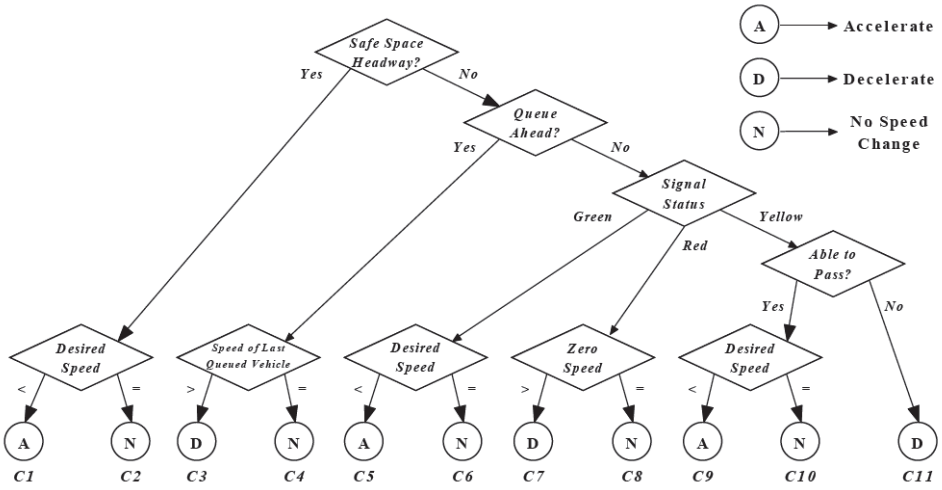


Figure 2-10 SMART-SIGNAL Virtual Vehicle Maneuver Decision Tree (Liu and Ma. 2009)

Virtual probe decides its acceleration dependent on given traffic states, such as queue and signal status step-by-step. The step-by-step maneuver selection will be continued until its destination and the time different between start and end time will be an arterial travel time. Figure 2-10 shows step-by-step virtual prove maneuver decision tree.

The suggested model is tested on a 1.83 mile long major arterial on France Avenue in Minneapolis, MN. This arterial includes 11 signalized intersections with a coordinated actuation signal controller. Figure 2-11 shows the virtual probe vehicle's trajectory with real floating car location. The study uses Root Mean Squared Percent Error (RMSP) as degree of model fitness. The reported estimation RMSP error is 0.0325, but the report did not mention test sample sizes.

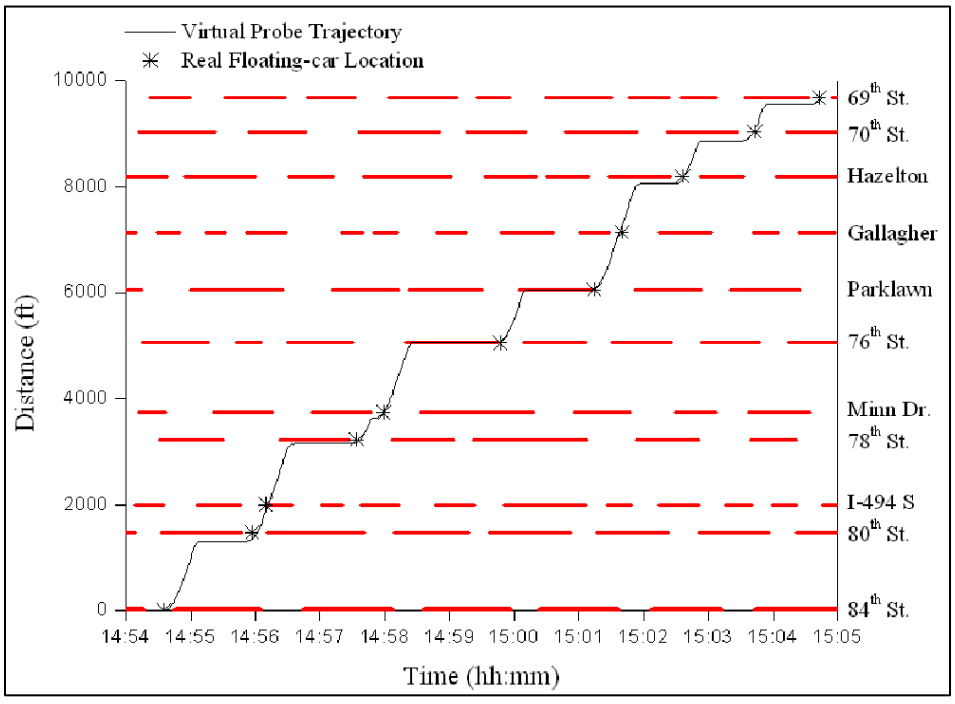


Figure 2-11 Virtual Vehicle Trajectory (Liu and Ma 2009)

2.2.2 Purdue Arterial Monitoring System (Day et al. 2010)

The Purdue Coordination Diagram (PCD) was developed by Purdue University and Indiana DOT. It uses phase status log and high resolution detector data for monitoring an intersection's level of performance and system level (arterial) performance. Figure 2-12 shows the Purdue monitoring systems flow.

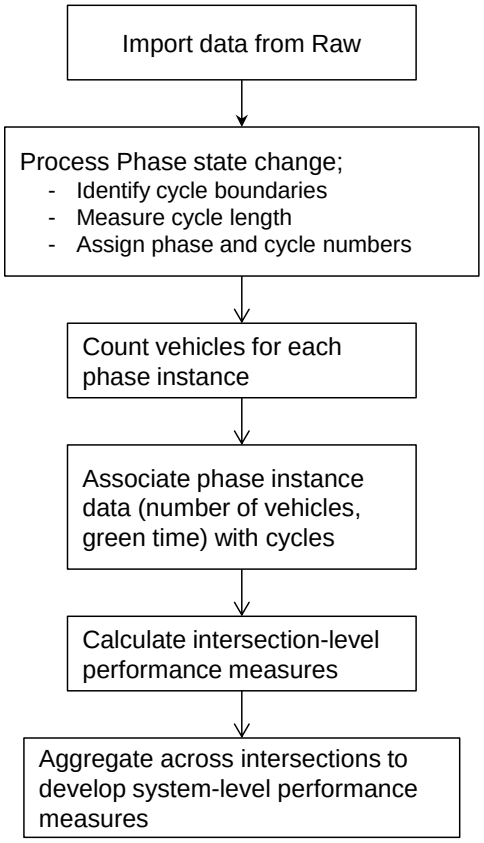


Figure 2-12 Flowchart for Purdue Signal Monitoring System
Source: NCHRP Project 3-79a

The Purdue system collects signal status data for monitoring cycle-by-cycle signal status. They developed a new analytical method to define dynamic cycle length. The created

dynamic cycle length and each phase will have a unique ID. In addition, each intersections detector high resolution data is also archived in the system for identifying vehicle arriving and vehicle location under the signal status. The Purdue signal monitoring system creates both intersection and arterial level performance measured by combining signal status data and high resolution detector data. Figure 2-13 shows controller log data sample.

The system log event data has three elements:

- A timestamp containing the data and time of the event, with resolution of 0.1 seconds.
- A number representing event type (Phase green, Phase yellow, Detector on, Detector off, etc.)
- A number representing the event channel. For phase information, this was the number of the phase for which the event was relevant.

Timestamp	Parameter	Channel	Explanation
04/08/09 14:10:49.6	8	22	Detector 22 off
04/08/09 14:10:49.9	9	7	Detector 7 on
04/08/09 14:10:50.1	8	7	Detector 7 on
04/08/09 14:10:51.1	63	2	**Phase 2 yield point**
04/08/09 14:10:51.3	2	2	**Phase 2 yellow state**
04/08/09 14:10:51.3	33	2	**Phase 2 termination: gap out**
04/08/09 14:10:51.8	9	10	Detector 10 off
04/08/09 14:10:51.9	9	28	Detector 28 off
04/08/09 14:10:52.0	8	10	Detector 10 on
04/08/09 14:10:52.0	9	12	Detector 12 off
04/08/09 14:10:52.0	8	28	Detector 28 on
04/08/09 14:10:52.1	8	12	Detector 12 on
04/08/09 14:10:52.4	9	17	Detector 17 off
04/08/09 14:10:52.8	9	19	Detector 19 off
04/08/09 14:10:53.0	8	19	Detector 19 on
04/08/09 14:10:56.1	9	21	Detector 21 off
04/08/09 14:10:56.4	3	2	**Phase 2 red clearance state**

Figure 2-13 System Log Data Sample
Source: NCHRP Project 3-79a

2.2.2.1 Split Failure Monitoring

The Purdue monitoring system archives signal status log. The total green time for a phase in a defined cycle is found by summing over all instances of the phase that occur within cycle. Therefore, capacity can be estimated under assumed, observed or estimated saturation flow rates, such as the following equation.

$$C_{\emptyset,a} = g_{\emptyset,a} * \frac{S_{\emptyset}}{3600}$$

Where, $C_{\emptyset,a}$: capacity provide to phase $\emptyset$ during cycle "a"

$g_{\emptyset,a}$: the amount of effective green time for cycle "a"

$S_{\emptyset}$: saturation flow rate of phase $\emptyset$

The equation represents the total number of vehicles that can be expected to be served at the saturation flow rate. However, cycle lengths will be changed by time of day plan, so capacity per cycle in units of vehicles becomes difficult to compare between different cycle lengths. Therefore, the equation needs to be normalized as:

$$C_{\emptyset,a} = \frac{g_{\emptyset,a}}{C_a} * S_{\emptyset}$$

Where, $C_{\emptyset,a}$: capacity provide to phase $\emptyset$ during cycle "a"

$g_{\emptyset,a}$: the amount of effective green time for cycle "a"

$S_{\emptyset}$: saturation flow rate of phase $\emptyset$

C_a : cycle length of "a"

Figure 2-14 shows a cycle-by-cycle effective green plot.

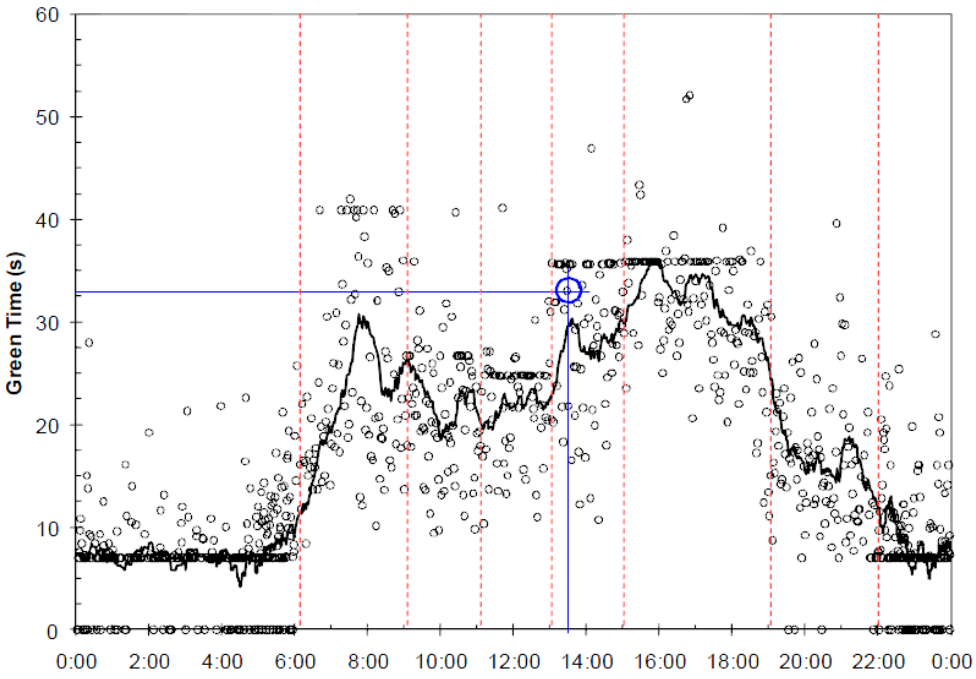


Figure 2-14 Observed Green Time
Source: NCHRP Project 3-79a

High resolution detector data is archived in the system and provides a vehicle count during each cycle. The cycle-by-cycle vehicle counts are normalized by the following equation and it can be directly compared with estimated capacity.

$$V_{\emptyset,a} = 3600 \frac{N_{\emptyset,a}}{C_a}$$

Where, $V_{\emptyset,a}$: hourly flow rate for phase $\emptyset$ during cycle "a"

$N_{\emptyset,a}$: the number of vehicle arriving during phase $\emptyset$ in cycle "a"

C_a : cycle length of "a"

Combing the normalized hourly flow rate and capacity allows the degree of saturation or volume to capacity ratio.

$$X_{\emptyset,a} = \frac{V_{\emptyset,a}}{C_{\emptyset,a}}$$

Where, $X_{\emptyset,a}$: normalized degree of saturation of phase ∅ during cycle "a"

$V_{\emptyset,a}$: hourly flow rate for phase ∅ during cycle "a"

$C_{\emptyset,a}$: capacity provide to phase ∅ during cycle "a"

The degree of saturation gives a measure that quantifies how much the provided green time is utilized by vehicles. Figure 2-15 shows the volume to capacity ratio monitoring results. The number of dots above the red line indicates signal failure. This method allows monitoring the frequency of signal failure per time of day plan for each phase.

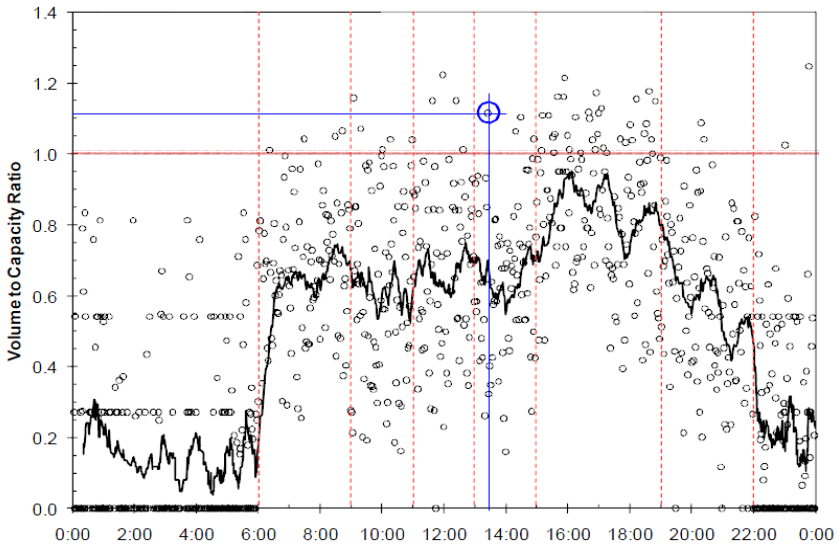


Figure 2-15 Volume to Capacity Ratio Monitoring Results
Source: NCHRP 409

2.2.2.2 Purdue Coordination Diagram

The Purdue Coordination Diagram (PCD) is a visualization tool for evaluating the quality of progression. Figure 2-16 shows the result of high resolution detector data and phase status data for over several cycles. The green and orange lines indicate start and end time of green for each cycle. The black dots represent each vehicle's location. The PCD directly provides the arrival on green percentage for each cycle.

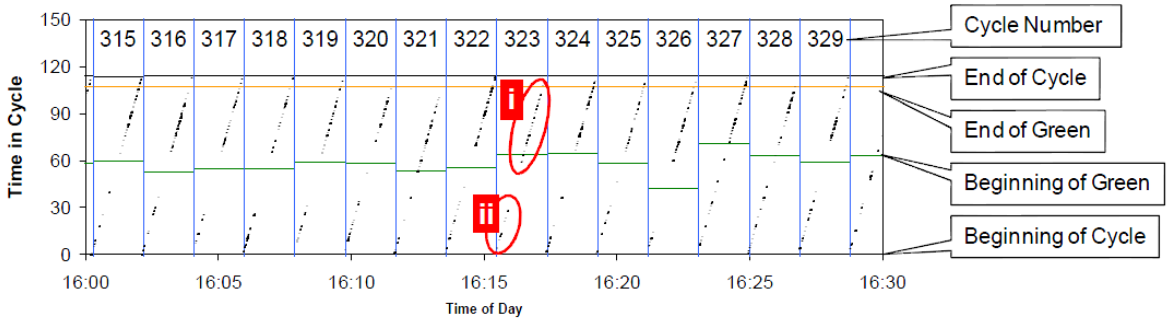


Figure 2-16 PCD over Several Cycles
Source: NCHRP Project 3-79a

The black dots (vehicle location) are derived from advance detectors so the PCDs reflect actual vehicle behavior on the corridor. Figure 2-17 shows the 24 hour extended PCD. The PCD describes vehicle arrival with coordinated phase status, so this plot gives information of the arterial coordination performance and provides a qualitative picture.

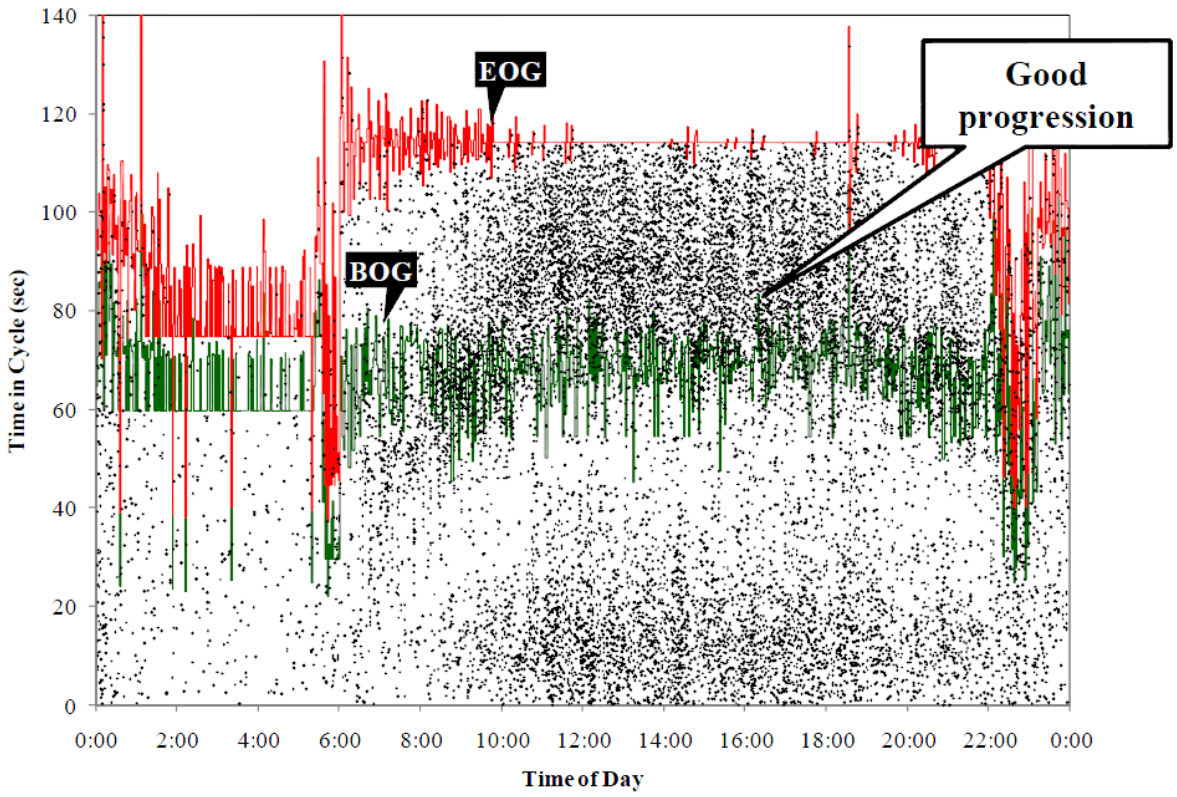


Figure 2-17 PCD Over 24 hours
Source: NCHRP Project 3-79a

2.3 State of Practice for Signal Timing Plan Development Process

Signal coordination to support platoon progression is a key focus of arterial signal timing plans. The decision of whether or not to coordinate adjacent signals is evaluated in different ways. The general consideration is the space in between consecutive intersections. Manual on Uniform Traffic Control Devices (MUTCD 2009) provides the guidance that traffic signals within 800 meters (0.5 miles) of each other along a corridor should be coordinated unless operating on different cycle lengths.

The purpose of signal coordination is to provide smooth flow of traffic along streets and highways in order to reduce travel times, number of stops, and delays. A well-designed signal system allows platoons to travel along an arterial or throughout a network of major streets with minimum stops and delays. Designating traffic movement with the high peak hour demand as the coordinated phase is the most common practice to achieve these goals. The coordination logic (semi-actuated) reserves unused green time for the coordinated phase when non-coordinated phases have low demand. In general, this logic more stable capacity on coordinated phases and results in fewer stops for the high demand arterial traffic movements.

Figure 2-18 provides a conceptual flow chart for the current state of practice in arterial traffic signal timing development. Most of US signalized arterials have existing signal timing plan so a majority of signal timing development efforts are involved with signal re-timing process. Arterial signal re-timing can be conducted by regular schedule or on a more ad hoc basis in response to complaint calls or known traffic environment changes.

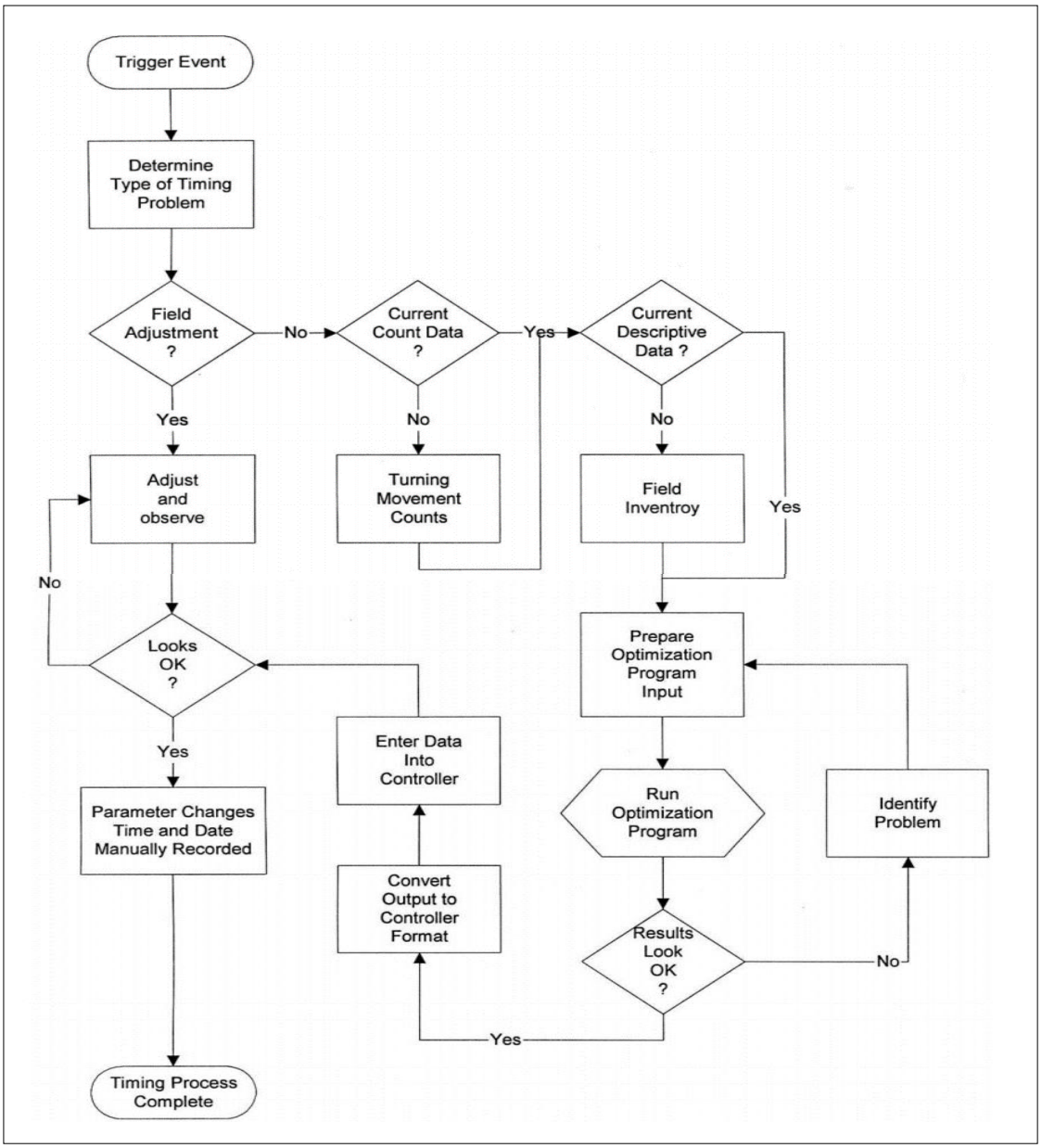


Figure 2-18 Classical Approach to Signal Timing
Source: NCHRP 409

Signal re-timing is a process that optimizes the operation of signalized intersections through a variety of low-cost improvements, including the development and implementation of new signal timing parameters, phasing sequences, and improved control strategies. When

arterial signal re-timing is decided to be necessary, the signal timing engineer conducts a travel time study to evaluating current conditions and progression quality. Many of DOTs in U.S. use the Tru-Traffic software for analyzing arterial travel time. Tru-Traffic allows the field travel time runs to be analyzed in relation to current timing plans. After conducting field travel time surveys, intersection turning movement data are collected. Computer based simulation and traffic analysis tools such as Synchro and Vistro are normally used for signal timing development. The collected turning movement counts for each intersection are essential input data for analysis and simulation. The computer analysis provides various signal timing options with an expected performance for each option. The signal timing engineer usually selects one of these near optimal timing plan options based on experience, expected performance, and engineering judgment for each of the time of day, day of week periods that will be served by a unique timing plan.

The next step is implementation of the selected signal timing plans in the field. After implementing the selected timing plan, the signal plan engineer conducts field fine-tuning of the coordination offsets. For the offset fine-tuning, the engineer visits the site to observe existing traffic conditions, paying special attention to operational characteristics such as each intersection's initial queue length, early return to green, queue spill back, etc. The signal plan engineer must rely solely on experience and his/her engineering judgment for field offset fine-tuning. After completion of field fine-tuning, another travel time survey will be conducted for before and after comparison.

During signal timing development, the signal plan engineer must use his/her experience and engineering judgment when making decisions because computer based

analysis tool can never provide an exact representation of the prevailing traffic conditions. In reality, many of the signal timing variables, such as minimum and maximum green time, green extension and etc., are given by or determined directly from policy. For example, NCDOT Traffic Management & Signal Systems Unit Design Manual provides guidelines for typical minimum green value (7 seconds), extension value (2 second for stretch detection, 3 seconds for low speed detection).

NCHRP 409 summarizes the current signal timing state of the practice by stating that:

- Many agencies do not review field performance data to determine the adequacy of signal timing at intervals less than three years.
- Many agencies do not review signal design, operations, maintenance, and training practices annually.
- Many agencies do not have precise and clearly stated policies that support detailed objectives.

2.4 Arterial Performance Measures

Performance measurement is the process of collecting, analyzing and reporting information regarding the performance of an individual or system. When considering signal timing among a series of signalized intersections (as for coordinated signal operation), performance measures that account for the relative intersection of adjacent intersections becomes important. In case of arterial performance measures at each intersection along an arterial or within a network, a number of performance measures are used to assess how well

the intersections fit together in terms of signal timing. The performance measures include number of stops, travel speed, and bandwidth.

2.4.1 Number of Stops

The number of stops is used frequently to measure an arterial or network's signal system effectiveness. Motor vehicle stops can often play a larger role than delay in the perception of the effectiveness of a signal timing plan a long an arterial street or a network. The number of stops or average numbers of stops per vehicle tends to be used more frequently in arterial applications where progression between intersections is a desired objective. The number of stops has not been identified as a candidate for standardization nationally. In addition, this measure is difficult to collect directly on the field. Therefore, computer based simulation tools are normally used for estimating and optimizing the number of stops. Stops are highly correlated with amount of emission and the quality of progression along arterials. FHWA's Traffic Signal Timing Manual states that the number of stops is an important measure because acceleration from stops is a major source of vehicle pollutants and surveys reveal that multiple stops along an arterial is more highly correlated with driver frustration than is delay.

2.4.2 Travel Speed

Travel speed (time) is one of the most popular measures used to assess for arterial progression. Arterial travel speeds account for both the delay at intersections and the travel time in between intersections. In the *HCM 2010*, the through vehicle travel speed in between two adjacent intersections is used for measuring an arterial's level of service (LOS). The

HCM defines arterial LOS as a function of the class of arterial under the study and the travel speed along the arterial. This speed is based on intersection spacing, the running time between intersections, and the control delay to through vehicles at each signalized intersection. Since arterial travel time (space mean speed) in HCM method is calculated segment by segment regardless of origin or destination, the resulting speed estimates may be different corresponding to speed (travel time) measurements made from end to end travel time runs. Data was collected by a GPS enabled floating car that measured a small subset of the possible origin-destination combinations along an arterial.

In *HCM 2010*, two performance measures are used to characterize LOS for a given direction of travel along an urban street segment. One measure is travel speed for through vehicles and the other measure is the volume-to-capacity ratio for the through movement at the downstream boundary intersection. Table 2.2shows the *HCM 2010* level of service thresholds established for the automobile mode on urban streets.

Table 2.2 HCM 2010 LOS Criteria

Travel Speed as a percentage of Base Free Flow Speed (%)	LOS by Volume-to-Capacity Ratio	
	≤ 1.0	> 1.0
> 85	A	F
> 67 – 85	B	F
> 50 -67	C	F
> 40 – 50	D	F
> 30 – 40	E	F
≤ 30	F	F

2.4.3 Bandwidth

The Federal Highway Administration Traffic Signal Timing Manual defines Bandwidth as the total amount of time available per cycle for vehicles to travel through a system of coordinated intersections at the progression speed, i.e. the time difference between the first and last hypothetical trajectory that can travel through the entire arterial at the progression speed without stopping. Bandwidth is an outcome of the signal timing that is determined by the offsets between intersections and the allotted green time for the coordinated phase at each intersection. Bandwidth is a parameter that is commonly used to describe capacity or maximized vehicle throughput. Bandwidth can be confirmed by the time-space diagram which is visual toll for engineers to analyze a coordination strategy and modify timing plans. Bandwidth, along with its associated measures of efficiency and attainability, are measures that are sometimes used to assess the effectiveness of a coordinated signal timing plan.

Bandwidth efficiency is calculated by:

$$B_e = \frac{(BW_1 + BW_2)}{2C}$$

Where, B_e: Two-way bandwidth efficiency

BW_1: Outbound bandwidth

BW_2: Inbound Bandwidth

Bandwidth attainability is calculated by:

$$B_a = \frac{(BW_1 + BW_2)}{(g_{min1} + g_{min2})}$$

Where, B_a: Two-way bandwidth attainability

g_{min1}: Outbound direction minimum green along the arterial

g_{min2}: Inbound direction minimum green along the arterial

Table 2.3 and Table 2.4 show the FHWA *Traffic Signal Timing Manual*'s two Guidelines for bandwidth efficiency and bandwidth attainability.

Table 2.3 Guidelines for Bandwidth Efficiency

Efficiency Range	Passer II Assessment
0.00 - 0.12	Poor Progression
0.13- 0.24	Fair Progression
0.25 – 0.36	Good Progression
0.37- 1.00	Great Progression

Table 2.4 Guidelines for Bandwidth Attainability

Attainability Range	Passer II Guidance
1.00 – 0.99	Increase minimum through phase
0.99 – 0.70	Fine-tuning needed
0.69 – 0.00	Major changes needed

2.5 Bandwidth Optimization

There are two distinct approaches to arterial bandwidth optimization. The first approach produces uniform bandwidths, while the second provides variable bandwidths. Two well-known software tools that implement the first approach are MAXBAND (Little et al. 1981) and PASSER II (Messer et al. 1973).

Messer et al. (*5*) developed Progression Analysis and Signal System Evaluation Routine (PASSER II) in 1973. PASSER II is a macroscopic deterministic optimization model. An iterative gradient search method is used to determine the phase sequence and cycle length and can provide the maximum two-way progression for a specified arterial signal system. Brook's Interference Algorithm (Brooks 1965) and Little's Optimized Unequal Bandwidth Equations (Morgan and Little 1964) are used in PASSER II. In PASSER II, cycle length, phase sequences, and offset are the decision variables and the objective is to minimize the total interference to progression.

Morgan and Little (1964) introduced a computation method to maximize arterial signal bandwidth. The widely used program of Little et al. (1966) efficiently finds offsets for maximum bandwidth given cycle time, red times, intersection spacing, and progression speed. Directional bandwidth weighting are set using a target value representing the directional flow rate ratio. Using a mixed-integer linear programming (MILP) approach, Little, Kelson and Gartner (1981) developed MAXBAND. The purpose of MAXBAND is to provide the maximum bandwidth setting for coordinated arterials. This program has the following capabilities:

- Identifies the best system cycle length within a given range

- Allows for different progression speeds for each link within a given range
- Provides optimal major street left turn phase sequencing
- Considers user specified queue clearance times
- Enables directional factors for two-way weighted bandwidth optimization

MAXBAND considers a two-way arterial with “n” intersections and specifies the corresponding offsets so as to maximize the number of vehicles that can travel within a given speed range without stopping throughout the arterial. Figure 2-19 shows the basic geometry defined for the MAXBAND mixed-integer linear program.

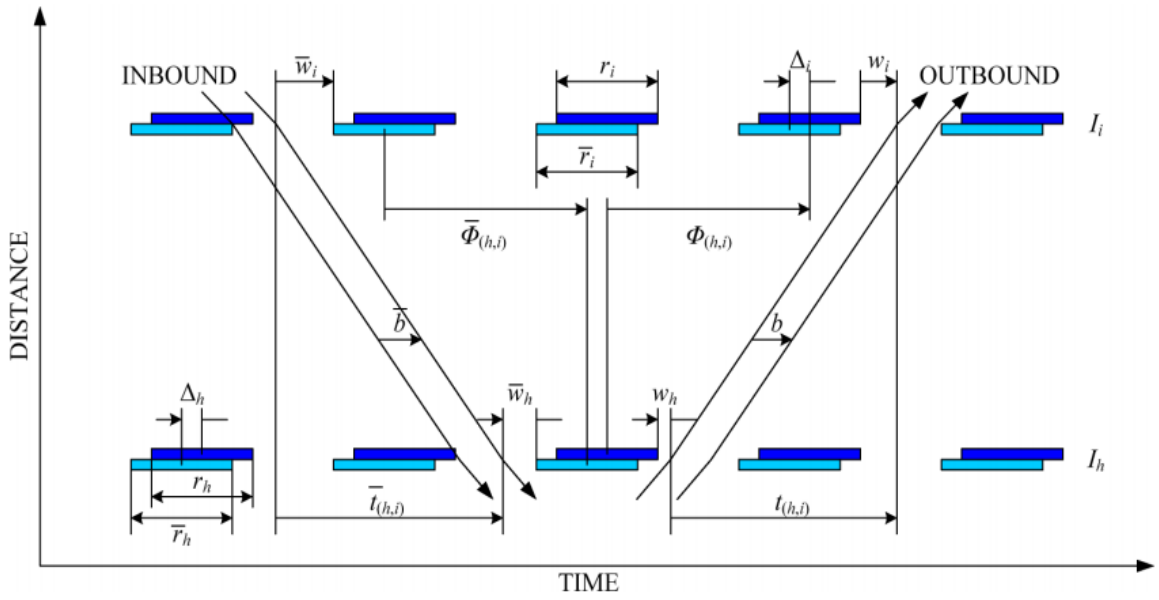


Figure 2-19 Time-Space Diagram for MAXBAND Model
(Source: Kai Lu, 2012)

In MAXBAND, phase splits at each intersection are assigned according to Webster's theory, and all signals are constrained to share a common cycle length. MAXBAND includes a queue clearance time in order to allow secondary flows which have accumulated during the red time to discharge before the platoon arrives.

The MILP form of MAXBAND is:

$$Max\ b + K\bar{b}$$

subject to

$$\begin{Bmatrix} \bar{b} \geq Kb\ if\ K < 1 \\ \bar{b} \leq Kb\ if\ K > 1 \\ \bar{b} = b \quad if \quad K = 1 \end{Bmatrix}$$

$$w_i + b \leq 1 - r_i \qquad \forall\ i = 1,2,\cdots,n$$

$$\bar{w}_i + \bar{b} \leq 1 - \bar{r}_i \qquad \forall\ i = 1,2,\cdots,n$$

$$\left(t_{i,i+1} + \bar{t}_{i,i+1}\right) + (w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (\Delta_i - \Delta_{i+1}) =$$

$$-\frac{1}{2} * (r_i + \bar{r}_i) + \frac{1}{2} * (r_{i+1} + \bar{r}_{i+1}) + (\bar{\tau}_i + \bar{\tau}_{i+1}) + m_{i,i+1} \qquad \forall\ i = 1,2,\cdots,n$$

$$\Delta_i = \left(\frac{1}{2}\right)[(2\delta_i - 1)l_i - (2\bar{\delta}_i - 1)\bar{l}_i$$

$$m_{i,i+1} = integer \qquad \forall\ i = 1,2,\cdots,n-1$$

$$b, \bar{b}, w_i, \bar{w}_i > 0 \qquad \forall\ i = 1,2,\cdots,n$$

$$\delta_i(\bar{\delta}_i) = binary$$

Where, $b\left(\bar{b}\right) = outbound\ (inbound) bandwidth$;

$K = target\ value\ (\frac{\bar{b}}{b})$;

$n = number\ of\ intersection$;

$I_i = signal\ intersection\ i;$

$r_i(\bar{r}_i) = outbound(inbound) red\ time\ at\ I_i;$

$w_i(\bar{w}_i) = time\ from\ right\ (left)\ side\ of\ red\ phase\ at\ I_i\ to$

$left\ (right) edge\ of\ outbound\ (inbound) green\ band;$

$t_{(h,i)}\left[\bar{t}_{(h,i)}\right] = travel\ time\ from\ I_h\ to\ I_i\ outbound(inbound);$

$\Delta_i =$ time from center of $\bar{r}_i$ $to\ nearest\ center\ of\ r_i;$

$m_{i,i+1} = loop\ interger\ variable\ related\ with\ I_i and\ I_{i+1};$

$l_i\left(\bar{l}_i\right) = time\ allocated\ for\ outbound(inbound)\ left\ turn\ green\ at\ I_i;$

$\delta_i(\bar{\delta}_i) = left\ turn\ phase\ sequence;$

The second class of optimization methods builds on the uniform bandwidth approach by providing bandwidths that vary by link. Tsay and Lin (1998) introduced BANDTOP, which generates a saw-toothed bandwidth pattern, rather than parallel and uniform patterns. In their paper, they showed that MAXBAND might not find the maximum bandwidth for certain signal-timing parameters. Gartner et al. (1991) proposed MULTIBAND, which can consider a variable bandwidth arterial progression. MULTIBAND used a multi-band/multi-weight concept to consider actual arterial link flow, thereby providing a capability to adapt the progression scheme to the specific traffic flow pattern that exists on the links of the arterial. MULTIBAND generates a global optimal solution that determines cycle length, offsets, progression speeds and phase sequences to maximize a combination of the individually weighted bandwidths in each directional arterial segment. Stamatiadis and Gartner (1996) developed the MULTIBAND-96 program which optimizes all the signal

control variables and generates variable bandwidth progressions on each arterial in the network. The MINOS mathematical programming package was used for the optimization. Gartner and Stamatiadis (2002) applied the MULTIBAND method to solve for an urban grid network. In this paper, the efficiency of MILP was improved by a heuristic network decomposition procedure.

Tian et al. (2003) proposed a heuristic approach to a bandwidth oriented signal timing method based on a system partitioning technique. A large signalized arterial was divided into subsystems of three to five intersections and then each subsystem is optimized to achieve the maximum bandwidth efficiency. From each subsystem solution, a large system offset was adjusted. This method provides maximum progression for the peak direction while maintaining partial progression for the off-peak direction. Lin et al. (2010) proposed a new mixed-integer nonlinear programming model for an optimal arterial-based progression algorithm. This model was designed to optimize the bandwidth while maximizing the number of non-stopping vehicles through downstream intersections. Lu et al. (2012) introduced a two-way bandwidth maximization model with a proration impact factor. Under a certain weighting factor, MAXBAND and MULTIBAND may not find the maximum bandwidth solution. Therefore, in this paper, authors introduced bandwidth proration impact factors, which indicate the target bandwidth demand ratio. Girianna and Benekogal (2004) used Genetic Algorithms (GA) to dynamically optimize signal-timing parameters in a discrete-time signal coordination problem. They applied their model to a one-way arterial network with 20 signalized intersections and reported promising results.

For minimizing the early return to green effect on coordinated arterials, Skabardonis (1996) and Chang (1996) proposed methods that consider semi-actuated arterial coordination. However, both methods used fixed green times. Skabardonis used an analytical estimate of average coordinated green as the fixed time split for offset optimization, and Chang used the average of field observed green times. Skabardonis and Geroliminis (2008) introduced an analytical model to estimate arterial travel time based on spatial and temporal queuing at the signalized intersection. They also introduced an archival data management system called APeMS. APeMS receives data from system and archives and analyzes the performance of arterial to provide performance measures such as travel time, number of stop, cycle failures, and travel time reliability in real-time. Yin et al. (2007) developed a methodology to fine-tune the offsets in arterial streets to improve the progression. Their approach, Offset Refiner, creates stochastic distribution of coordinated phases' green start and end times at each intersection using archived signal data. Then they estimate maximum expected bandwidth and the corresponding offsets based on the stochastic distributions.

Pengfei et al. (2011) collected signal ATMS data and created distributions of coordinated phase green. Based on the created intersection green time distributions, the optimal offsets were derived using a Monte Carlo simulation method. However, this method hardly provides real optimal offsets since average or mode of early return to green for each intersection combination has very low chance to happen on the field. Early return to green and green extension phenomenon depends on non-coordinated approaches' vehicle arrival pattern and theses values are time dependent. Therefore, using one statistical value for adjusting offset in one set of time of day plan may not improve arterial progression quality.

Recently, Purdue University and the Indiana DOT (Day et al. 2010) developed a new system, which provides "Purdue Coordination Diagram." This innovate system uses hi-resolution vehicle event data to optimize cycle length and vehicle arrival percentage during green time. In addition, this system uses first vehicle estimated trajectory to adjust offset to improve percentage of vehicle arriving on green. University of Minnesota (Liu and Ma 2009) also developed real-time arterial performance monitoring system called SMART-SIGNAL. SMART-SIGNAL also uses hi-resolution data to calculate vehicle arrival type and the proportion of vehicle arriving during green phase. This system can adjust offsets using virtual probe trajectories. Both new signal systems require hi-resolution data, which can provide event base detector data. Both approaches use similar offset adjustment method to improve percentage of vehicle arriving on green. Zheng et al. (2013) proposed an approach to measure queue length and vehicle control delay at signalized intersections using traffic count data archived by traffic sensors. They implemented their algorithm in a computerized system and evaluated it by both field and simulation experiments. They stated that the algorithm showed encouraging results on queue length and control delay measurements.

CHAPTER 3 AN ENHANCED MAXBAND FORMULATION WITH ROBUST SOLUTION OPTIMALITY

This chapter focuses on a bandwidth optimization mixed-integer linear program (MILP) that was given the name MAXBAND in a 1981 paper by Little, Kelson, and Gartner. MAXBAND has been modified and expanded over the years in formulations known as MAXBAND-86, MULTIBAND, and MULTIBAND-96 (Chang et al. 1988, Gartner et al. 1991, Stamatiadis and Gartner 1996). However, the key formulation structures are essentially unchanged across the succeeding versions. More specifically, the research presented in the present paper discovered that all the MAXBAND formulations will yield sub-optimal two-way bandwidth solutions under certain conditions. As will be explained later, such suboptimal solutions arise from the specific way the MAXBAND objective function and constraints are designed to ensure that the more heavily traveled direction does not consume too much of the bandwidth in situations where there is a direct tradeoff between bandwidth in the two directions.

In this chapter, an enhanced formulation of MAXBAND is introduced. The proposed new formulation is identical to the MAXBAND formulation with the exception of the objective function and two constraints. Case studies are presented to show that the proposed formulation eliminates the possibility of finding suboptimal offset solutions. The original MAXBAND formulation is applied to two hypothetical scenarios designed to illustrate that the original formulation is over-constrained in situations where the maximum possible inbound and outbound bandwidth sizes are different. The revised MAXBAND formulation is

then presented and tested on the same hypothetical networks. Finally, the enhanced formulation is tested for consistency with the original formulation using the Euclid Avenue arterial in Cleveland, Ohio and hypothetical networks presented in Little's 1966 sole authored mixed integer linear programming paper. In addition to dealing with the issue of suboptimal solutions, the proposed enhanced formulation also provides the maximum proportion of estimated traffic demand that can be served within the optimal bandwidth.

3.1 MAXBAND Formulation Evaluation

MAXBAND is built on a mathematical program structure for solving two-way bandwidth optimization problems. MAXBAND considers directional imbalance in traffic demand through the directional demand ratio, *k*, which is calculated as the ratio of inbound demand to outbound. This ratio is incorporated into a general constraint to ensure that bandwidth allocation is proportional to directional demand. The general constraint is expressed as $(1-k)\bar{b} \geq (1-k)kb$, where b and $\bar{b}$ represent the bandwidth in the outbound and inbound directions, respectively. For the three specific demand ratio conditions, this constraint can be rewritten as:

$$b = \bar{b} \qquad if \qquad k = 1 \tag{3.1}$$

$$\bar{b} \leq kb \qquad if \qquad k > 1 \tag{3.2}$$

$$\bar{b} \geq kb \qquad if \qquad k < 1 \tag{3.3}$$

The objective function of MAXBAND model is:

$$Maximize\ (b + k\bar{b}) \tag{3.4}$$

The directional sizes of bandwidths are restricted by $(1-k)\bar{b} \geq (1-k)kb$ constraint, and the remaining constraints of MAXBAND MILP are:

$$w_i + b \leq C - r_i,\ \ \bar{w}_i + \bar{b} \leq C - \bar{r}_i, \qquad i = 1, 2, \cdots, n \tag{3.5}$$

$$(t_i + \bar{t}_i) + (w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (\Delta_i - \Delta_{i+1}) - Cm_i =$$

$$-\frac{1}{2}(r_i + \bar{r}_i) + \frac{1}{2}(r_{i+1} + \bar{r}_{i+1}) + (\tau_{i+1} + \bar{\tau}_i), \qquad i = 1, 2, \cdots, n-1 \tag{3.6}$$

b, $\bar{b}$, w_i and $\bar{w}_i > 0$, $\qquad i = 1, 2, \cdots, n$ (3.7)

m_i is integer. $\qquad i = 1, 2, \cdots, n$ (3.8)

Figure 3-1 shows the basic geometry of the MAXBAND MILP. The constraint equations (3.5) and (3.6) are the principal constraints of the MAXBAND model. The model becomes a MILP due to the integer decision variable m_i. The variable m_i is derived in the various works cited and referred to as the loop integer. Although equivalent, the so called looping constraint given in equation (3.6) differs from the original MAXBAND formulation in that the original MAXBAND formulation normalized all time values by the common coordination cycle length, C. The formulation given above uses actual time values, and therefore equation (3.6) can be derived from the original looping constraint by multiplying all

terms in the constraint by C. Rounding out the variable notation, r_i ($\bar{r}_i$) represents outbound (inbound) red indication time at signal i, t_i ($\bar{t}_i$) denote outbound (inbound) directional travel time between signals i and $i+1$, w_i ($\bar{w}_i$) represent time from the right (left) side of red at signal i to the left (right) edge of the outbound (inbound) green band, and Δ_i denotes time between the center of the inbound and outbound red indications at signal i. Δ_i will be a positive time value when center of r_i is later in time than the center of $\bar{r}_i$ and negative otherwise.

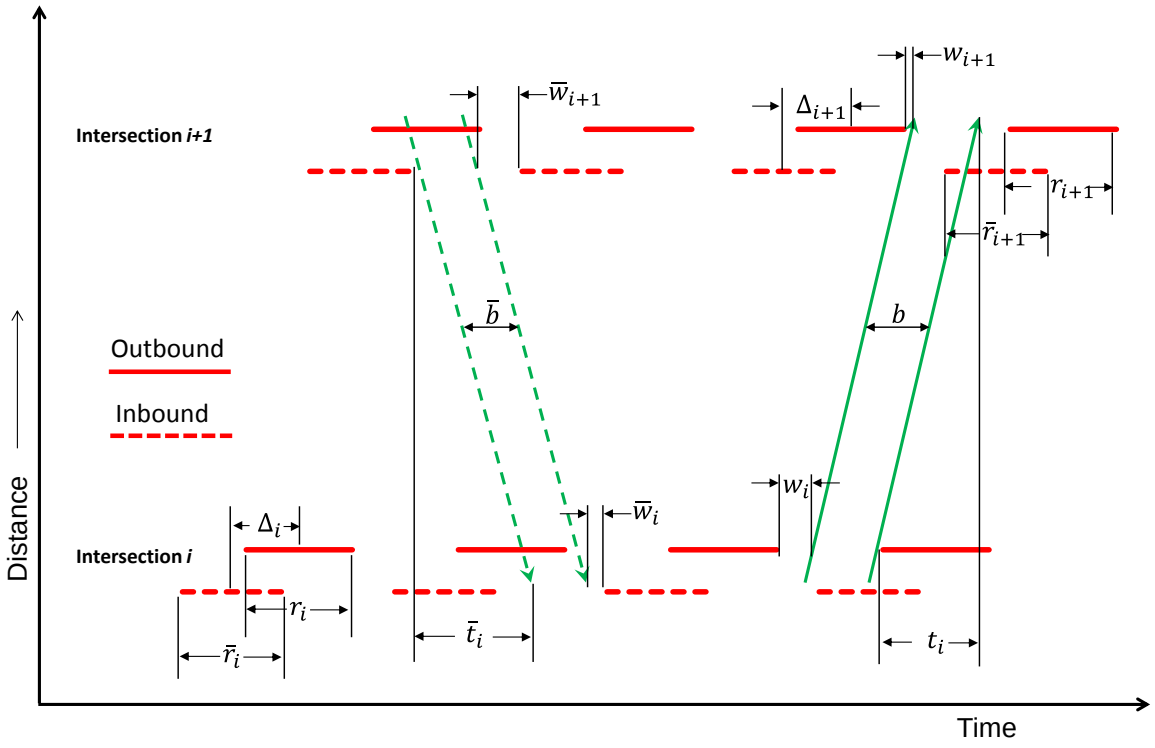


Figure 3-1 MAXBAND Bandwidth Geometry

Two hypothetical network scenarios were created to examine the potential for suboptimality in the family of MAXBAND MILPs. The characteristics of the first hypothetical network scenario is:

- Three coordinated signalized intersections
- Common cycle length of 100 seconds
- A distance of 825 feet between adjacent intersections
- Progression speed of 66ft/sec (45mph)
- Effective red time duration of 50 seconds for all intersection in both directions
- No left turn phases and no queue clearance time
- Total demand of 1,000 vehicles per hour per lane (vphpl)
- Inbound to outbound demand proportion varying from 9:1 to 1:9

Under given conditions equation (5.6) becomes:

$$(t_i + \bar{t}_i) + (w_i + \overline{w}_i) - (w_{i+1} + \overline{w}_{i+1}) - Cm_i = 0 \quad i = 1, 2, \cdots, n-1 \tag{3.9}$$

Furthermore, t_i ($\bar{t}_i$) is a constant value of 12.5 seconds (825/66). Therefore, maximum sum of two-way bandwidth, $Max\ (b + \bar{b})$, will be determined by only $w_i(\overline{w}_i)$, $w_{i+1}(\overline{w}_{i+1})$ and m_i within constraint equation (3.5). The calculated bandwidth under various "*k*" and offset with decision variable w and *m* are shown in Table 3.1. If the optimal offsets are defined using outbound start of green as the zero reference point, the offsets can be calculated by equation (3.10).

$$OFS_{i+1} = OFS_i + w_i + t_i + \tau_{i+1} + w_{i+1} \quad (3.10)$$

Table 3.1 provides the results from the original MAXBAND formulation under the nine different directional proportion conditions. Figure 3-2 and Figure 3-3 shows all possible offset combinations with a one-second resolution of offset change found by an exhaustive search approach (Sangkey Kim 2013). The Solution IDs in Figure 3-2 and Figure 3-3 correspond to a unique combination of offsets. The solution ID can be used to determine the offset combination. The exhaustive search method starts with incrementally increasing offset value at the last intersection of the system while other intersections have an offset of zero, and assigns an ID (starting from one) to each offset combination. After searching over all offset values at the last intersection, the offset value of the second to last intersection becomes one, and all offsets at the last intersection are searched again and IDs are assigned. This process is continued for all intersection of the system except for the first. On the Figure 3-2, each black cross corresponds to a combination of offsets that yields a total sum of bandwidth of 50 seconds for the case with 400 and 600 vphpl demands in the outbound and inbound directions, respectively. Therefore, the exhaustive search results support that the MAXBAND solution is *one* of the Pareto optimal global solutions in each case. In addition, Table 3.1 illustrates perfect trade off relationship between outbound and inbound bandwidth which means the target value related constraint equation $(1-k)\bar{b} \geq (1-k)kb$ is working perfectly. The directional bandwidth is allocated proportionally according to the directional demand in keeping with the intended purpose of the target value *k*.

Table 3.1 MAXBAND Optimal Solutions for Hypothetical Test Network Scenario 1

Demand (vphpl)		k	w_1 ($\bar{w}_1$)	w_2 ($\bar{w}_2$)	w_3 ($\bar{w}_3$)	m_1, m_2, m_3	Offset			Bandwidth (seconds)		
OUT	IN						Int. 1	Int.2	Int. 3	b	$\bar{b}$	Sum
100	900	9.00	0 (0)	25 (0)	45 (5)	0, 0, 0	0	-12.5	-20	5	45	50
200	800	4.00	0 (0)	25 (0)	40 (10)	0, 0, 0	0	-12.5	-15	10	40	50
300	700	2.33	0 (0)	25 (0)	35 (15)	0, 0, 0	0	-12.5	-10	15	35	50
400	600	1.50	0 (0)	25 (0)	30 (20)	0, 0, 0	0	-12.5	-5	20	30	50
500	500	1.00	0 (0)	25 (0)	25 (25)	0, 0, 0	0	-12.5	0	25	25	50
600	400	0.67	0 (0)	25 (0)	20 (30)	0, 0, 0	0	-7.5	5	30	20	50
700	300	0.43	0 (0)	15 (10)	15 (35)	0, 0, 0	0	-2.5	10	35	15	50
800	200	0.25	0 (0)	10 (15)	10 (40)	0, 0, 0	0	2.5	15	40	10	50
900	100	0.11	0 (0)	5 (20)	5 (45)	0, 0, 0	0	7.5	20	45	5	50

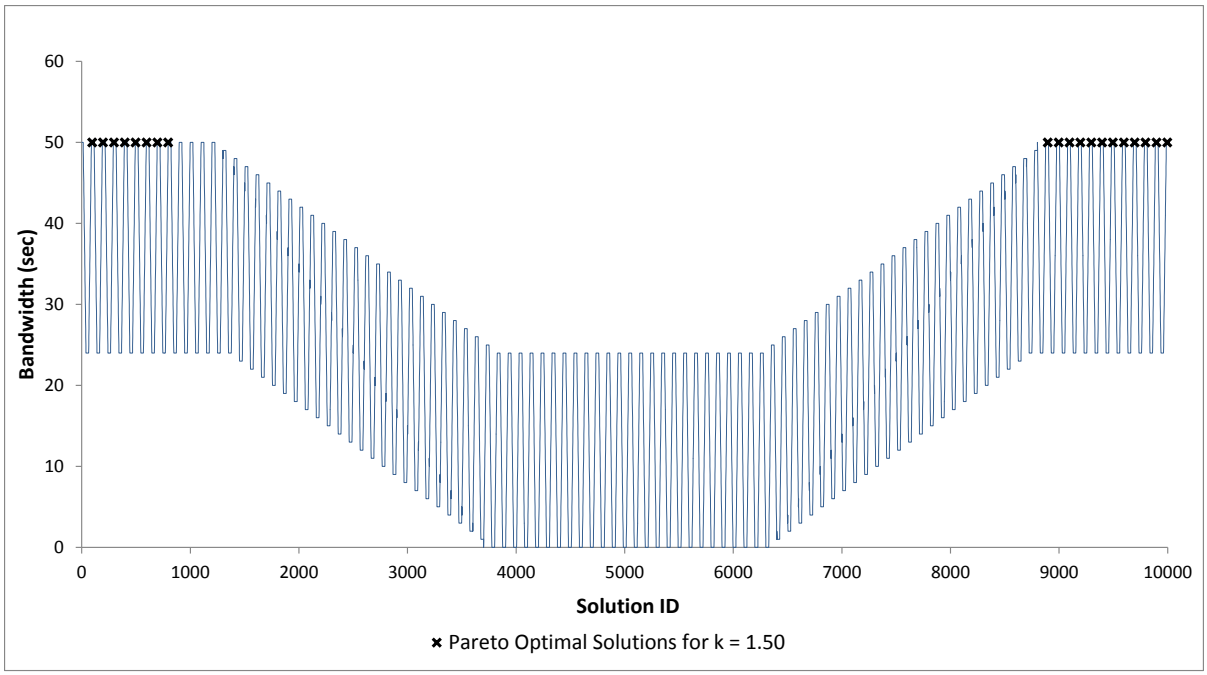


Figure 3-2 Hypothetical Test Network 1 Exhaustive Search Results

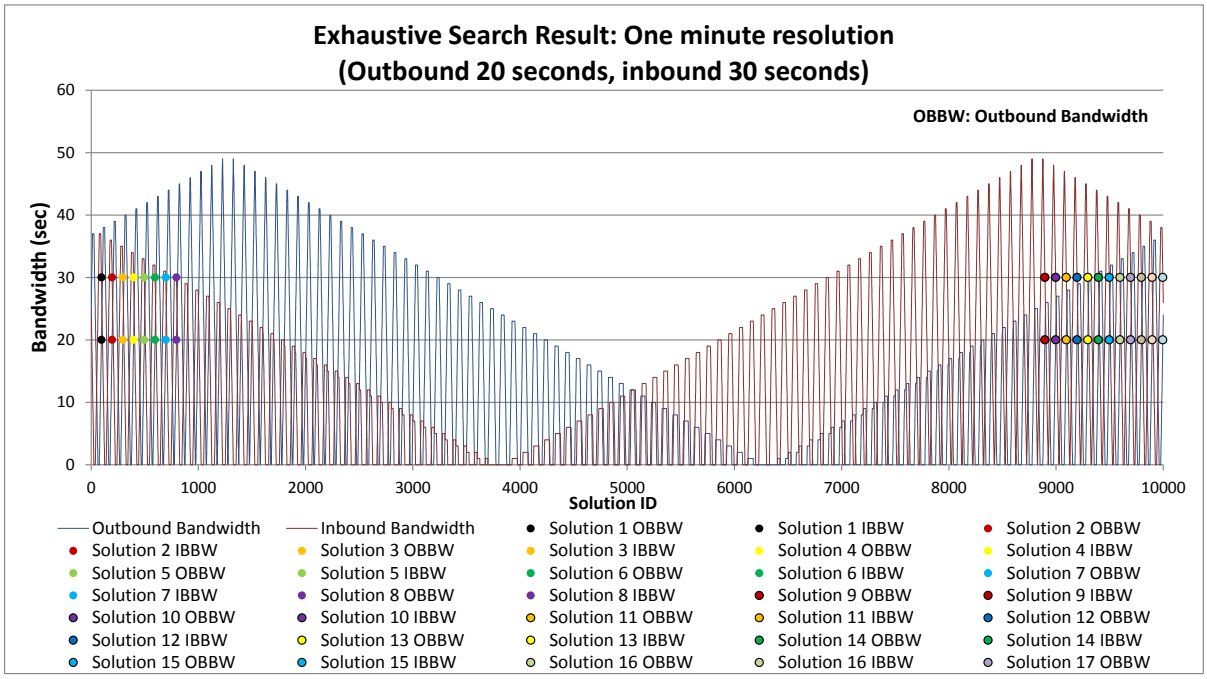


Figure 3-3 Hypothetical Test Network 1 Exhaustive Search Results of Directional Bandwidth

Figure 3-4 shows the example time-space diagram of the MAXBAND optimal solution under the $k = 1$ condition. The diagram helps in understanding the role of equations (3.1) to (3.3). For example, when k becomes 1.5, intersection 3 offset moves to left side for 5 seconds in order to allocate an additional 5 seconds outbound bandwidth to inbound.

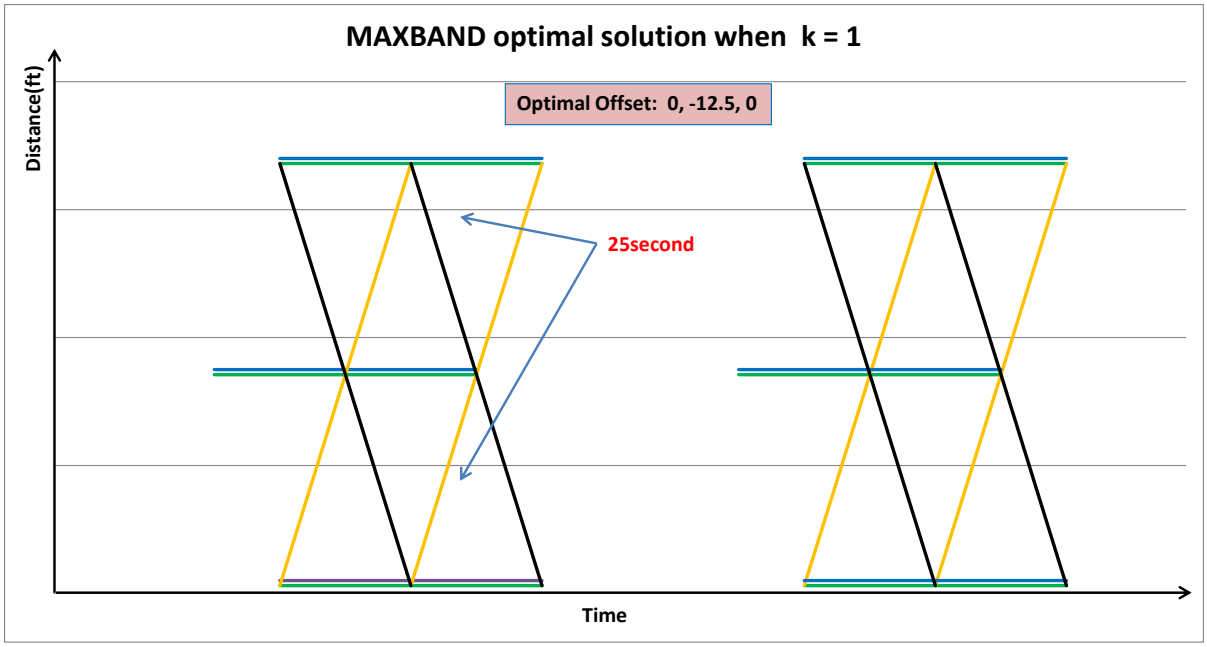


Figure 3-4 Hypothetical Test Network 1 MAXBAND Optimal Offsets when $k = 1$

The second hypothetical network conditions are same as first network conditions except:

- Second intersection inbound effective red time is 90 seconds
- Second intersection has a 40 seconds outbound lagging left turn phase.

Table 3.2 shows the second hypothetical network MAXBAND solutions. The suboptimal solutions are highlighted in gray. For example, because the maximum possible inbound bandwidth is limited to 10 seconds, the outbound bandwidth is also limited to 10 seconds when k equals 1, and MAXBAND therefore provides a suboptimal solution.

Figure 3-5 shows the MAXBAND solutions when k is equal to 1 for the second hypothetical network scenario. The calculated MAXBAND optimal solution satisfies constraints but is clearly suboptimal. The suboptimality occurs because of the combination of unbalanced green durations with specific values of k. Figure 3-5 illustrates that, as mentioned above, how unbalanced green duration at the second intersection limits the maximum possible inbound bandwidth to 10 seconds, which in turn constrains outbound bandwidth to be equal to inbound bandwidth by constraint equation (3.1). However, it is obvious that the offsets at intersections 2 and 3 could be reduced to provide significantly more outbound bandwidth without lessening the inbound bandwidth. In effect, equation (3.1) has gone beyond the intended purpose of preventing the heavier demand direction from taking too much time from the lighter demand direction by rigidly enforcing the bandwidth proportion even under conditions where no such directional bandwidth tradeoff exists. The scenario with k = 0.67 provides a helpful further illustration. In this case, the outbound demand exceeds the

inbound demand by a factor of 1.5. However, even though an offset solution exists that would provide 25 seconds of outbound green band while still achieving the maximum possible inbound bandwidth of 10 seconds, constraint equation (3.3) limits the outbound bandwidth to no more than 1.5 times the inbound band, and therefore, MAXBAND provides a solution with only 15 seconds of outbound bandwidth.

Table 3.2 MAXBAND Optimal Solutions for Hypothetical Test Network Scenario 2

Demand (vphpl)		k	w_1 ($\overline{w}_1$)	w_2 ($\overline{w}_2$)	w_3 ($\overline{w}_3$)	$\Delta_1, \Delta_2, \Delta_3$	m_1, m_2, m_3	Offset			Bandwidth (seconds)		
OUT	IN							Int. 1	Int. 2	Int. 3	b	$\bar{b}$	Sum
100	900	9	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0	27.5	15	25	10	35
200	800	4	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0	27.5	15	25	10	35
300	700	2.33	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0	27.5	15	25	10	35
400	600	1.5	0 (15)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0	12.5	0	25	10	35
500	500	1	40 (10)	35 (0)	0 (0)	0, 20, 0	0, 1, 0	0	17.5	65	10	10	20
600	400	0.67	35 (15)	35 (0)	0 (0)	0, 20, 0	0, 1, 0	0	12.5	60	15	10	25
700	300	0.43	15 (0)	0 (0)	26.7 (38.3)	0, 20, 0	0, 1, 0	0	27.5	13.3	23.3	10	33.3
800	200	0.25	15 (0)	0 (0)	22 (43)	0, 20, 0	0, 0, 0	0	27.5	18	28	7	35
900	100	0.11	15 (0)	0 (0)	18.5 (46.5)	0, 20, 0	0, 0, 0	0	27.5	21.5	31.5	3.5	35

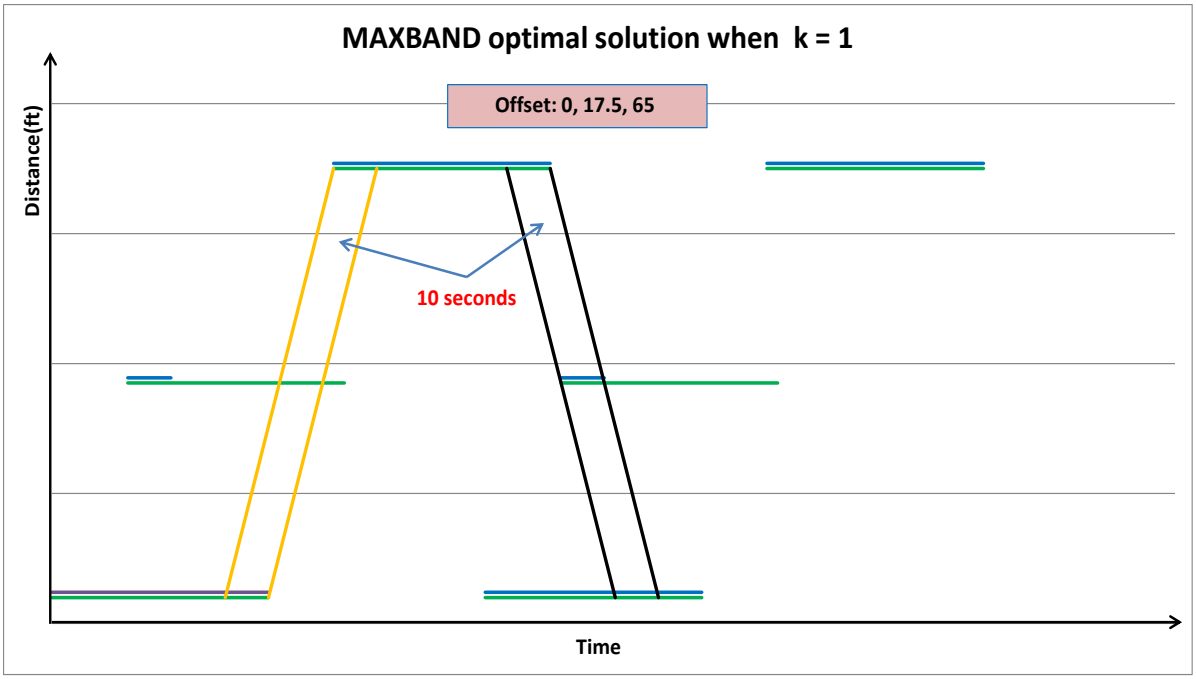


Figure 3-5 Hypothetical Test Network 2 MAXBAND Optimal Offsets when $k = 1$

3.2 New Objective Function and Constraints

An enhanced MAXBAND formulation was developed to address the suboptimality possibility illustrated in the previous section. The new formulation is:

$$Maximize\ \{(b + k\bar{b}) + \alpha \cdot M\} \tag{3.11}$$

$$b \geq \alpha \cdot D, \tag{3.12}$$

$$\bar{b} \geq \alpha \cdot \bar{D}, \tag{3.13}$$

$$w_i + b \leq C - r_i,\ \ \bar{w}_i + \bar{b} \leq C - \bar{r}_i, \qquad i = 1, 2, \cdots, n \tag{3.14}$$

$$(t_i + \bar{t}_i) + (w_i + \bar{w}_i) - (w_{i+1} + \bar{w}_{i+1}) + (\Delta_i - \Delta_{i+1}) - Cm_i =$$

$$-\frac{1}{2}(r_i + \bar{r}_i) + \frac{1}{2}(r_{i+1} + \bar{r}_{i+1}) + (\tau_{i+1} + \bar{\tau}_i), \qquad i = 1, 2, \cdots, n-1 \qquad (3.15)$$

b, $\bar{b}$, w_i, and $\bar{w}_i > 0$, $\qquad i = 1, 2, \cdots, n \qquad (3.16)$

m_i is integer, $\qquad i = 1, 2, \cdots, n \qquad (3.17)$

$k = \bar{D}/D$, (3.18)

$0 < \alpha \leq 1$. (3.19)

The first two terms of the revised objective function are identical to the original objective function, and as before, these terms include the target value (k) to maximize the sum of two-way bandwidths while favoring the direction with the heavier traffic demand level. The conditional constraint in the original formulation selected from equations (3.1) through (3.3) as appropriate are replaced with constraint equations (3.12) and (3.13). Two new variables are introduced. First, D and $\bar{D}$ respectively denote the average number of seconds per cycle required to process the demand level in the outbound and inbound directions. In the new formulation $k = \bar{D}/D$. Second, α denotes the minimum fraction of the directional demand that can be processed in the optimal directional bandwidths. The objective function in the new formulation maximizes this fraction of demand α ($0 < \alpha \leq 1$) that can be processed in the directional bandwidths. The objective function is designed to make optimization of α dominant by using a large multiplier M. Using constraints (3.12) and (3.13) guarantees that in each direction, the provided directional bandwidths is greater than or equal to the total seconds required to process the maximized fraction of the demand level.

The remaining constraints from the original formulation are retained unchanged. As will be illustrated below, the proposed enhanced MAXBAND formulation is consistent with the original MAXBAND formulation for conditions where the potential for suboptimality is not an issue while removing the potential for suboptimality in situations where it would arise for the original formulation.

3.3 New Model Test

As mentioned above, new formulation requires determining the directional demand in terms of seconds per cycle for D and $\overline{D}$ rather than merely setting the target bandwidth proportion value *k*. Using 2 seconds as a uniform platoon headway, the hypothetical demand values from the previously introduced scenarios were converted to seconds per cycle. Table 3.3 shows the enhanced MAXBAND results for the first hypothetical test network scenarios. The new decision variable α is calculated at 0.9 which means that provided bandwidth can serve 90% of demand within the optimal bandwidth. Although some of the solution offsets are different from the original MAXBAND solution due to slack in the intersection 2 optimal offsets, the optimized outbound and inbound bandwidth values are identical to the solutions from the original MAXBAND formulation for all the values of *k*.

Table 3.3 New Model Optimal Solutions for Hypothetical Test Networks 1

Demand (seconds)		k	w_1 ($\bar{w}_1$)	w_2 ($\bar{w}_2$)	w_3 ($\bar{w}_3$)	m_1, m_2, m_3	α	Offset			Bandwidth (seconds)		
OUT	IN							Int. 1	Int. 2	Int. 3	b	$\bar{b}$	Sum
5.56	50	9.00	0 (0)	25 (0)	45 (5)	0, 0, 0	0.9	0	-7.5	-20	5	45	50
11.1	44.4	4.00	0 (0)	15 (10)	40 (10)	0, 0, 0	0.9	0	-2.5	-15	10	40	50
16.7	38.9	2.33	0 (0)	10 (15)	35 (15)	0, 0, 0	0.9	0	2.5	-10	15	35	50
22.2	33.3	1.50	0 (0)	5 (20)	30 (20)	0, 0, 0	0.9	0	7.5	-5	20	30	50
27.8	27.8	1.00	0 (0)	0 (25)	25 (25)	0, 0, 0	0.9	0	12.5	0	25	25	50
33.3	22.2	0.67	0 (0)	0 (25)	20 (30)	0, 0, 0	0.9	0	12.5	5	30	20	50
38.9	16.7	0.43	0 (0)	0 (25)	15 (35)	0, 0, 0	0.9	0	12.5	10	35	15	50
44.4	11.1	0.25	0 (0)	0 (25)	10 (40)	0, 0, 0	0.9	0	12.5	15	40	10	50
50	5.56	0.11	0 (0)	0 (25)	5 (45)	0, 0, 0	0.9	0	12.5	20	45	5	50

Table 3.4 shows the enhanced MAXBAND results for the second hypothetical network scenarios. For the scenarios highlighted in gray, the original MAXBAND provided suboptimal solutions (see the corresponding rows in Table 3.2). Note that the maximum possible inbound bandwidth in this case is limited to 10 seconds. However, as can be seen below, the new formulation provided optimal bandwidth solutions for these cases. The calculated α varies from 0.2 to 0.63 which indicates the critical movement demand (seconds) to bandwidth ratio. When both directional bandwidths are larger than demand the calculated α is 1. When at least one of the directional bandwidth is less than demand (α less than 1), α equals the bandwidth to demand ratio of the critical direction. The model pushes to maximize

the value of α to ensure that the critical direction receives the highest possible bandwidth. The non-critical direction will serve at least the same proportion (α) of its demand.

Table 3.4 New Model Optimal Solutions for Hypothetical Test Networks 2

Demand (seconds)		k	w_1 ($\overline{w}_1$)	w_2 ($\overline{w}_2$)	w_3 ($\overline{w}_3$)	$\Delta_1, \Delta_2, \Delta_3$	m_1, m_2, m_3	α	Offset			Bandwidth		
OUT	IN								Int. 1	Int. 2	Int. 3	Out	In	Sum
5.56	50	9.00	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.2	0	27.5	15	25	10	35
11.1	44.4	4.00	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.225	0	27.5	15	25	10	35
16.7	38.9	2.33	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.257	0	27.5	15	25	10	35
22.2	33.3	1.50	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.3	0	27.5	15	25	10	35
27.8	27.8	1.00	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.36	0	27.5	15	25	10	35
33.3	22.2	0.67	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.45	0	27.5	15	25	10	35
38.9	16.7	0.43	15 (0)	0 (0)	25 (40)	0, 20, 0	0, 0, 0	0.6	0	27.5	15	25	10	35
44.4	11.1	0.25	15 (0)	0 (0)	22 (43)	0, 20, 0	0, 0, 0	0.63	0	27.5	18	28	7	35
50	5.56	0.11	15 (0)	0 (0)	18.5 (46.5)	0, 20, 0	0, 0, 0	0.63	0	27.5	21.5	31.5	3.5	35

Figure 3-6 shows the new formulation's optimal solution under the *k* =1 condition with MAXBAND solution in the background. The original MAXBAND provides only 10 seconds of bandwidth in the outbound direction because equation (3.1) imposes a hard constraint that the two bandwidths must be equal (see Figure 3-5). However, the new

formulation provides the full 25 seconds of outbound bandwidth that can be provided without sacrificing any of the 10 seconds of inbound bandwidth. Also, for the k = 0.67 scenario, the new formulation provides 25 seconds of outbound bandwidth rather than being unnecessarily constrained to 15 seconds as was the case with the original formulation.

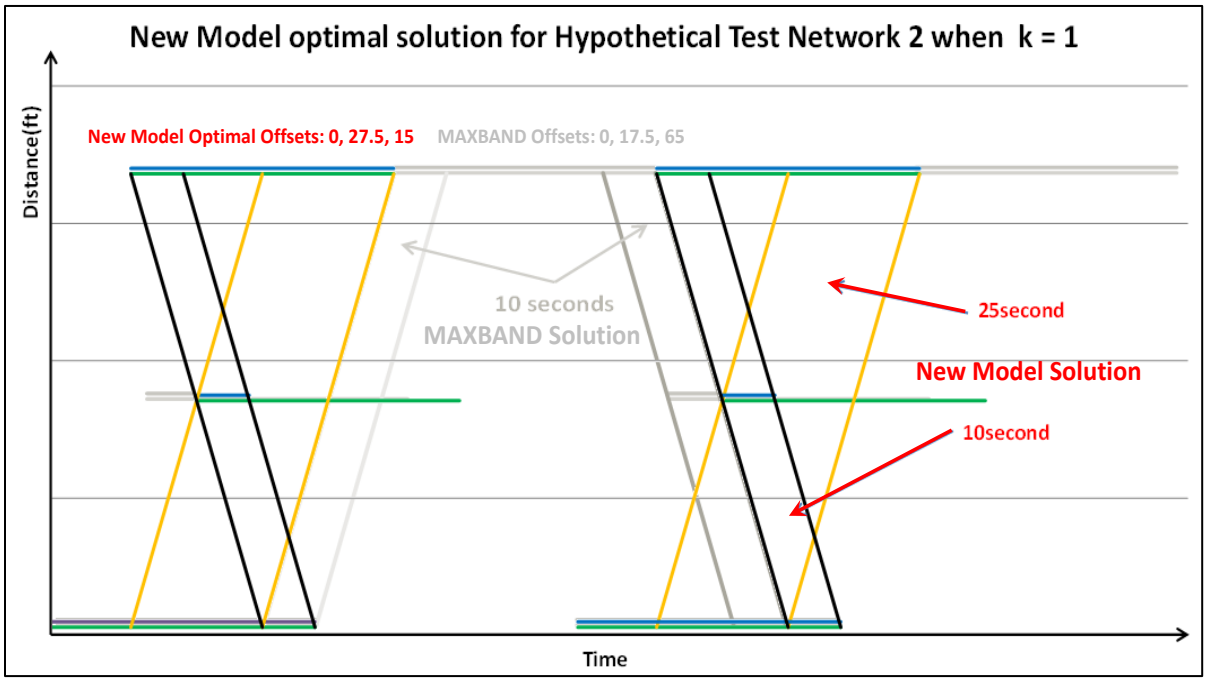


Figure 3-6 Hypothetical Test Networks 2 New Model Optimal Offsets when k = 1

As a further demonstration of appropriate consistency with the original MAXBAND formulation, the enhanced formulation was tested on the Cleveland, Ohio Euclid Avenue scenario used in Little's 1966 paper. The arterial as presented includes ten intersections. The intersections are located at relative distances of 0, 168, 381, 716, 929, 1173, 1371, 1493, 1706, and 1843 meters. The corresponding red times are 0.47, 0.40, 0.40, 0.47, 0.48, 0.42,

0.40, 0.40, 0.40 and 0.42 cycles. The coordinated cycle length is 65 seconds with each intersection's location and green time provided as in Table 3.5. In keeping with Little's example, directional demand is assumed equal (k = 1). The progression speed for all links is 15.2 meters/second (49.87 ft/sec). Metric distance units are converted into US customary units in Table 3.5.

Table 3.5 Euclid Avenue Arterial in Cleveland Information

Intersection	Distance (ft)	Red (Sec)	Green (sec)
1	0	30.55	34.45
2	551.18	26	39
3	1250.00	26	39
4	2349.08	30.55	34.45
5	3047.90	31.2	33.8
6	3848.42	27.3	37.7
7	4498.03	26	39
8	4898.29	26	39
9	5597.11	26	39
10	6046.58	27.3	37.7

Figure 3-7 shows the new formulation test result for Euclid Avenue. Both the original MAXBAND and the proposed new formulation provide identical solutions under the target value $k = 1$ condition (*7*). This result supports that the new formulation duplicates the results from MAXBAND's original formulation when suboptimality is not an issue.

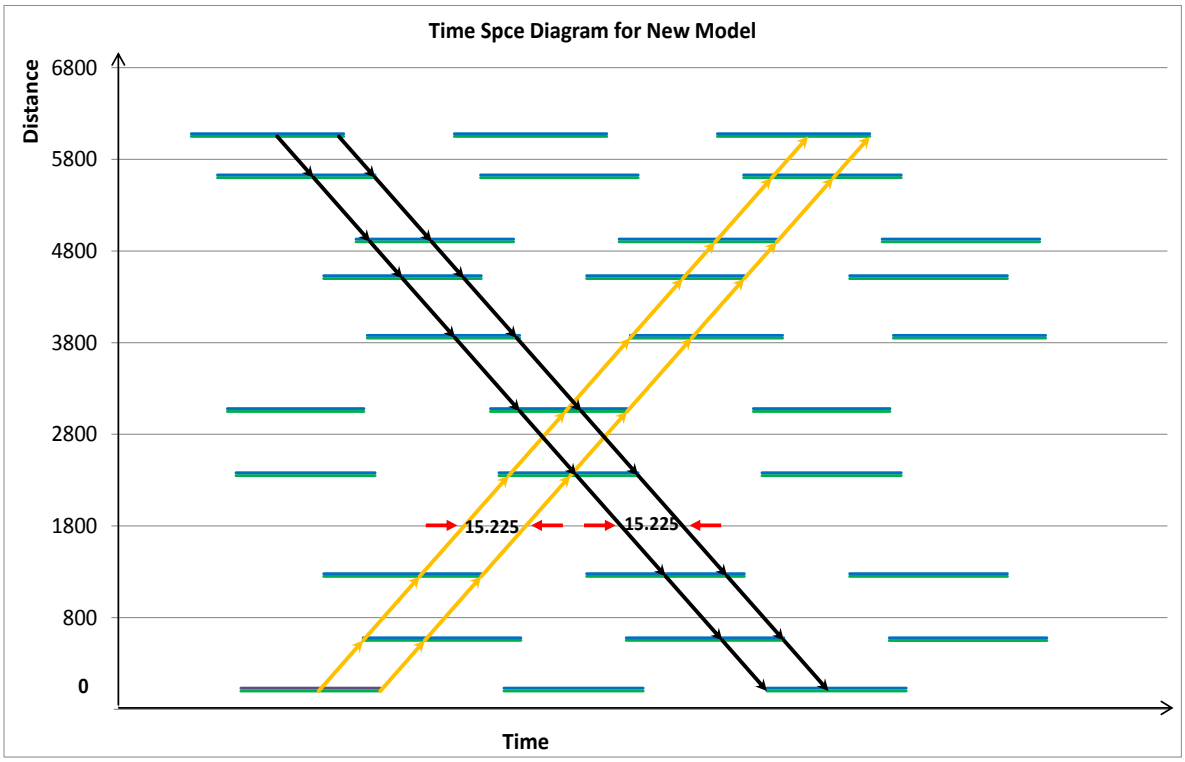


Figure 3-7 Euclid Ave from Little New Model Optimal Offsets when $k = 1$

Table 3.6 shows the comparison result of MAXBAND and the proposed new formulation under various target values for *k*. Under all values of *k*, the original and enhanced MAXBAND formulations yield identical results because the Euclid Avenue scenario has a perfect trade off relation between outbound bandwidth and inbound bandwidth

as did the first hypothetical test network scenarios. Therefore, the total bandwidth sum is always 30.45 seconds, and the directional bandwidth is allocated according to the *k* value.

Table 3.6 Euclid Avenue Arterial in Cleveland Case Comparison Result

Demand (vphpl)			**Outbound Bandwidth**		**Inbound Bandwidth**	
OUT	**IN**	**k**	**MAXBAND (7)**	**NEW Model**	**MAXBAND (7)**	**NEW Model**
100	900	9	3.045	3.045	27.405	27.405
200	800	4	6.09	6.09	24.36	24.36
300	700	2.33	9.135	9.135	21.315	21.315
400	600	1.5	12.18	12.18	18.27	18.27
500	500	1	15.225	15.225	15.225	15.225
600	400	0.67	18.27	18.27	12.18	12.18
700	300	0.43	21.315	21.315	9.135	9.135
800	200	0.25	24.36	24.36	6.09	6.09
900	100	0.11	27.405	27.405	3.045	3.045

3.4 Conclusion

In this chapter, an enhanced MAXBAND formulation is introduced to overcome the potential for the MAXBAND family of bandwidth optimization MILPs to yield suboptimal solutions under certain conditions. Using hypothetical network scenarios, the MAXBAND over constrainted problem is revealed, and the proposed enhanced formulation is tested.

Optimization performance of the original and enhanced formulations are compared using two hypothetical test network scenarios and the real world arterial used in Little's 1966 paper. The test result supports the notion that the new formulation is consistent with the original MAXBAND formulation in cases where over constraint and suboptimality are not an issue and that it also yields an optimal solution in cases where the original formulation suffers from suboptimality due to over constraint of the bandwidth ratio.

The modified objective function and constraints in the new formulation are proposed to replace MAXBAND's formulation as well as new editions to the MAXBAND family of optimization models such as MULTIBAND, MULTIBAND-96, and AM-BAND. Although the MILP formulations for MULTIBAND, MULTIBAND-96, and AM-BAND have increasingly developed innovative objective functions and constraints compared to the original MAXBAND MILP, all the MAXBAND-related models have similar constraints incorporating directional bandwidth target value *k*. For example, all three models include the general $(1 - k_i)\overline{b}_i \geq (1 - k_i)k_i b_i$ constraint that yields the situational constraints shown in equations (3.1) through (3.3). Therefore, all these models inherently have the potential over constraint and suboptimality issue illustrated above.

The proposed new formulation overcomes the MAXBAND MILP family over-constrained problem assuming a fixed duration of the effective red and green intervals. However, coordinated arterial green in most modern signal systems is not static but varies dynamically on a cycle-by-cycle basis yielding in turn progression bandwidths that vary from cycle-by-cycle. Therefore, further development of a proposed enhanced MAXBAND

formulation as well as all the other arterial performance optimization efforts should be designed to consider the impact of dynamic bandwidths.

CHAPTER 4 FIELD OBSERVATIONS AND PROCESSING

The purposes of this chapter are to explore real field cycle-by-cycle dynamic greens from ATMS data sources and develop dynamic bandwidth analysis tool (DBAT). In this chapter, feasible ATMS data sources, study area selection and dynamic bandwidth analysis tool (DBAT) developing results are summarized. This chapter is organized as follows: Section 4.1 introduces the feasible data source used to develop the dynamic bandwidth analysis tool. Section 4.2 shows the differences between programmed green and dynamic green as well as early return to green distribution under selected study sites. This is followed by the tool development (section 4.3) and the dynamic bandwidth analysis tool test result (section 4.4).

4.1 Data and Site Description

The NCDOT closed-loop signal systems for the most part utilize OASIS^{TM} for closed loop signal systems operations. OASIS is a traffic control firmware developed by Econolite for implementation in an Advanced Transportation Controller (ATC) Type 2070 published by AASHTO, ITE, NEMA, and CALTRANS. NCDOT's effort to transition all state maintained systems to OASIS system should enable the streamlining of access to data for state maintained signals and systems. This system provides log data for monitoring the system's operational status.

OASIS provides seven system event log file histories, which are shown in

Table 4.1. These include: system alarms, special events, front panel data entry, coordination plans, implemented functions, split monitoring, and detector count station data. These system logs are stored in the non-volatile RAM memory and can be cleared upon upload from a central computer. Among those seven logs, the log data of interest for this research is split monitoring log data for exploring real field green and early return to green with green extension phenomenon.

Table 4.1 Available OASIS Log Files

Logs	Data Type	
System Alarms Log	• Detector Failures • Hardware Failures	• Phase Conflict • Logs Full
Special Events Log	• Stop Time • Police Switch	• Preemptions
Front Panel Entries Log	• Data Element modified • Old data value • New data value	• Current user • Timestamp
Coordination Plans Log	• Source of plan implementation • Plan implemented	• Offset • Timestamp
Implemented functions Log	• Source of function implementation • Function implemented	• Timestamp
Split Monitor Log	• Active Vehicle Phases • Active Vehicle Phases State • Active Pedestrian Phases • Active Pedestrian Phases State • Active Overlaps • Active Pedestrian Overlaps • Coordination Plan	• Local Clock • Offset • Preemptions • Vehicle Calls • Pedestrian Calls • Status Response Packet
Detector Data Log	• Detector Reference • Detector Status • Average Wait • Volume	• Occupancy • Average Speed • Average Gap

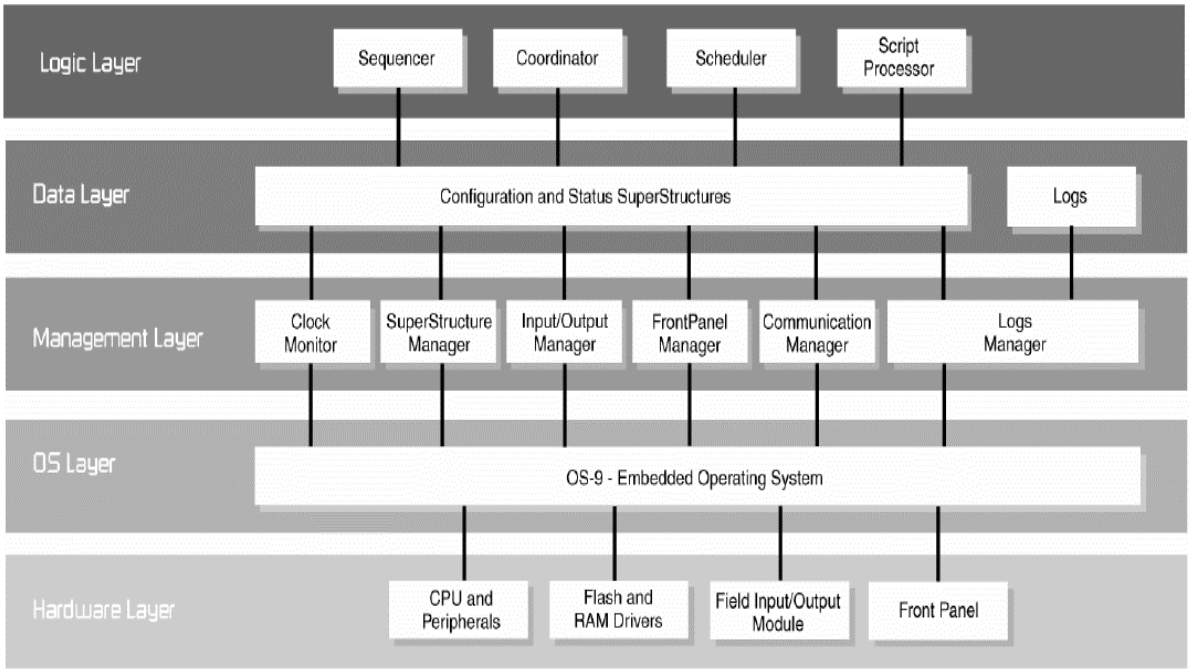


Figure 4-1 OASIS System Overview

4.1.1 OASIS Split Monitor Data

The TransLink 32 software is required to download the OASIS log file. It is designed to monitor the 170 and 2070 master and local controllers by Econolite. Through TransLink 32, the OASIS log file can be manually or automatically downloaded and archived into the Access database. The OASIS split monitor reports the following: “Time Stamp”, “Cycle”, “Offset”, “Plan”, “CoordPhases”, “ExtraTimeCP”, “UsedPhase” and “AllottedPhase” as explained next.

- SMTimeStamp: Start time of each cycle (reference point)
- Cycle: Coordinated cycle length

- Offset: A time relation (local clock) with regard to the system time reference (master clock) to indicate where, in that cycle, the intersection begins / ends its coordinated phase(s) Green (main street Green)
- CoordPhases: Coordinated phase number (e.g., CoordPhases2, CoordPhases6)
- ExtraTimeCP: Remaining time in the current cycle allocated to the coordinated phase
- UsedPhase: Actual green time used in the current cycle (After reference Point)
- AllottedPhase: Maximum phase time, which includes yellow and red times

OASIS use standard National Electrical Manufacturing Agency (NEMA) phase numbers. Figure 4.3 shows a sample intersection phase sequence with each movement number.

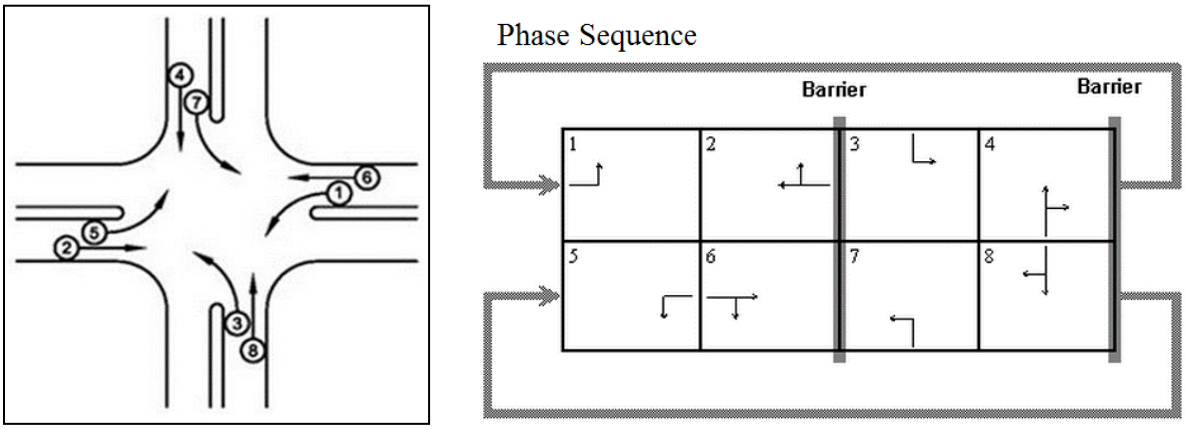
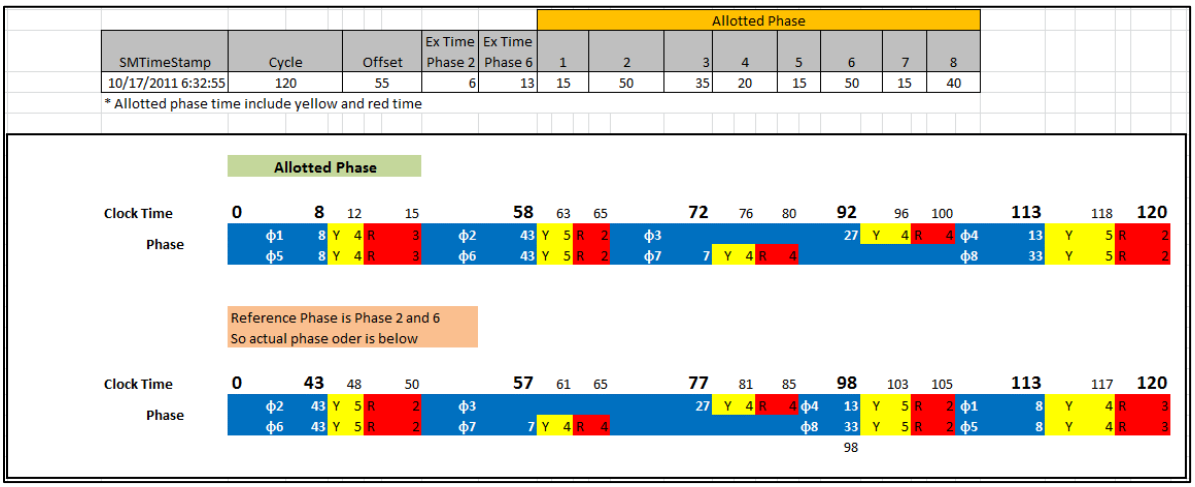


Figure 4-2 NEMA Phase with Phase Sequence

It is possible to build programmed signal phases and cycle by cycle real dynamic phases from the split monitor data. Those results are shown in Figure 4-3 and Figure 4-4.



					Allotted Phase							
SMTimeStamp	Cycle	Offset	Ex Time Phase 2	Ex Time Phase 6	1	2	3	4	5	6	7	8
10/17/2011 6:32:55	120	55	6	13	15	50	35	20	15	50	15	40

* Allotted phase time include yellow and red time

Figure 4-3 Programmed Intersection Signal Phases

Dynamic green can be generated from the "Time Stamp" (reference point) and "UsedPhase" with "ExtraTimeCP" which indicates how early the green is returned, compared to the reference point. As an example, Figure 4-4 shows the Split Monitor log table with dynamic phase sequences which are created from the Split Monitor log.

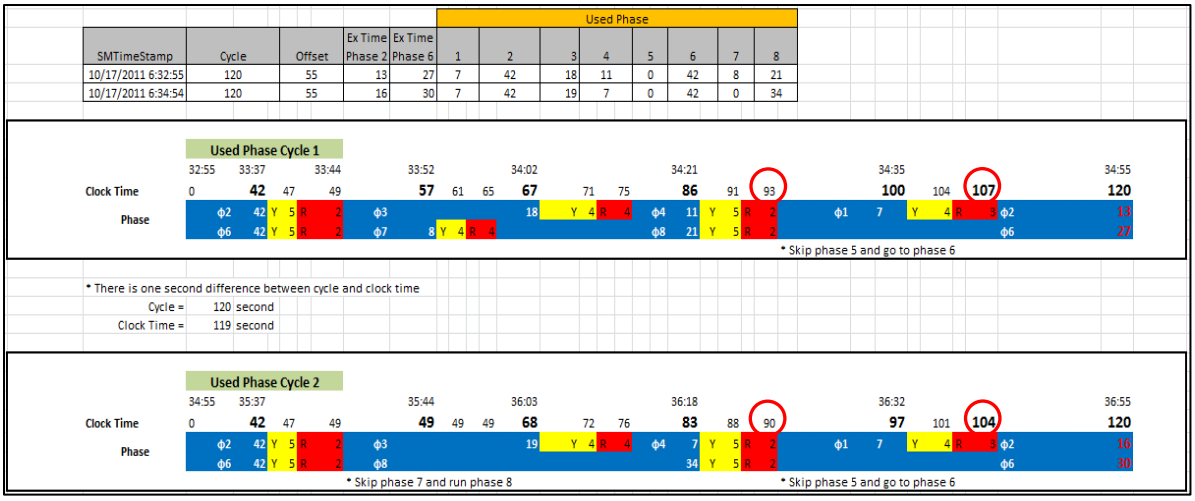


Figure 4-4 Dynamic Intersection Phases

The coordinated phases are phases 2 and 6. During the first cycle duration that is 120 seconds long, there is unused time that goes to ExtraTimeCP. This includes 13 seconds from phase 1 (which terminates at 107 seconds) and 27 seconds from phase 5 (which is skipped) and phase 8 is terminated at 93 seconds (with 27 seconds remaining). Phases 2 and 6 greens will then have early returns by those amounts of time.

However, it should be notes that the ExtraTimeCP is not exactly the same as early return to green when lead-lag phasing is used or with lead-lead phasing with different greens for phase 1 and phase 5. Phase 1 time will go to “ExtraTimeCP6” for phase 6 and the remaining time which is phase 6 green time minus phase 1 will go to phase 6 “UsedPhase” (see Figure 4-5).

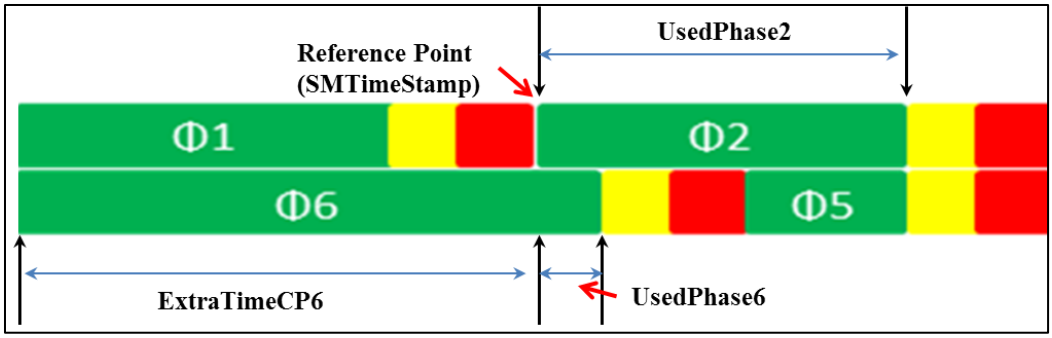


Figure 4-5 Lead-Lag phase

The coordinated phase green start clock time is the reference point time minus the previous cycle's "ExtraTimeCP"; the green terminated clock time is the reference point time plus the current cycle's "UsedPhase". This rule is not affected by the left turn phase sequence.

At that point, the author is sufficiently familiar with both the capabilities and limitations of the OASIS log data files to begin planning for a pilot data collection, as described in the next section.

4.1.2 Data Collection

For this study, three closed loop systems are selected by following criteria:

- The signal control system is closed loop
- The site covers a wide range of traffic conditions
- The corridor has at least three coordinated signalized intersections
- The corridor contains a few driveways and un-signalized intersections

The first study site is on US 70 in Clayton, NC. This system includes three signalized intersections. The second site is on US 70 in Garner, NC with four signalized intersections. The final site is on NC 55 arterial in Apex, NC with seven signalized intersections. These study sites are shown in Figure 4-7.

The OASIS split monitor log data were collected by the TransLink 32 software for a period exceeding two weeks, then downloaded and archived in a Microsoft Access database. Figure 4-6 shows a screenshot of the TransLink32 interface with the scheduler.

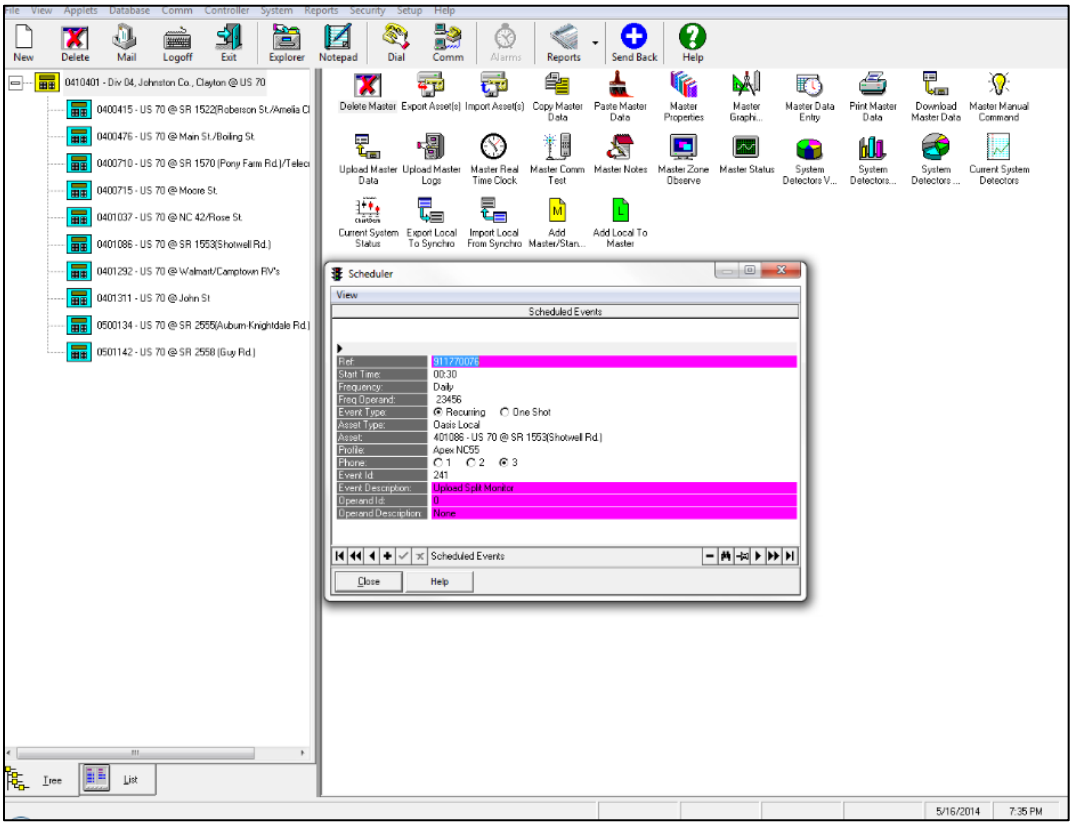
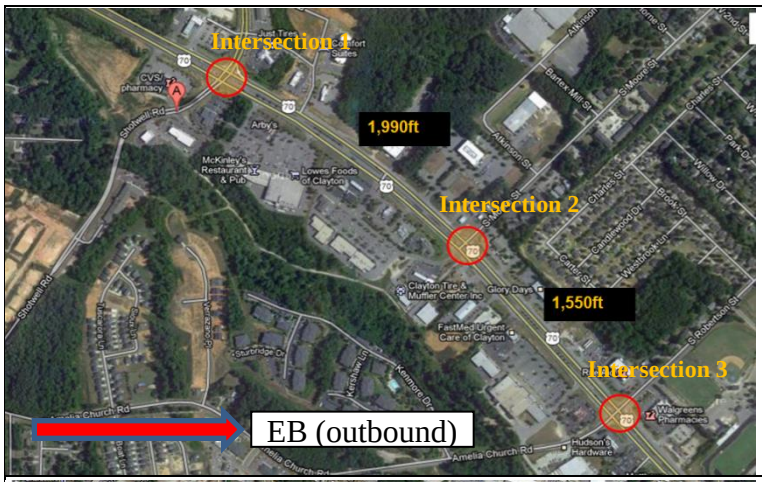
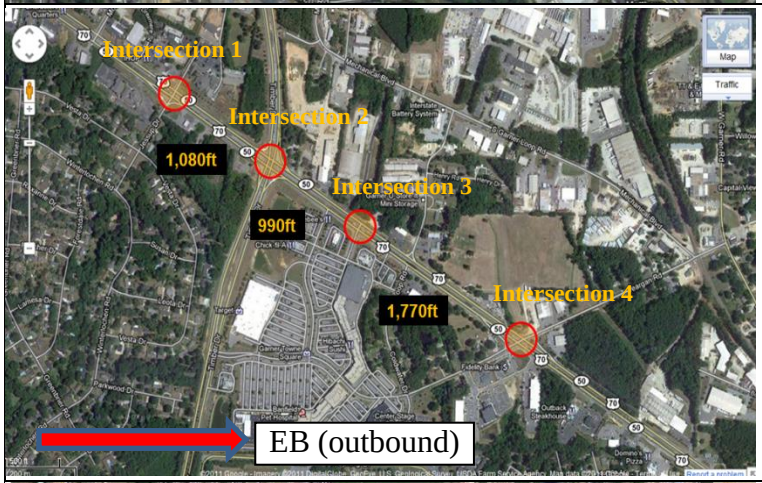


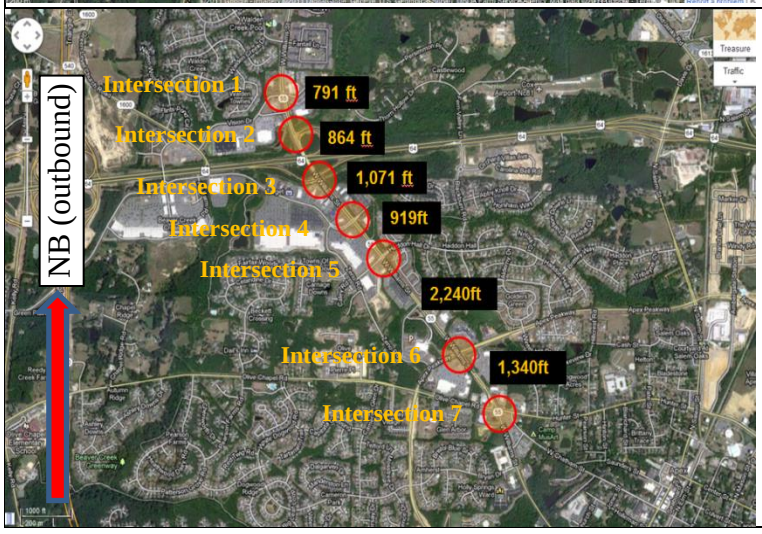
Figure 4-6 Translink32 Data Download Schedule Display



Site A
- Arterial on US 70 in Clayton, NC
- 3 signalized intersections
- No un-signalized intersections
- 22 driveways
- 3,540 ft. total route length
- Selected plan: AM peak plan
- Cycle Length: 170 seconds
- Intersection 1 offset: 34 seconds
- Intersection 2 offset: 37 seconds
- Intersection 3 offset: 120 seconds



Site B
- Arterial on US 70 in Garner, NC
- 4 signalized intersections
- No un-signalized intersections
- 15 driveways
- 3,840 ft. total route length
- Selected plan: AM peak plan
- Cycle Length: 120 seconds
- Intersection 1 offset: 62 seconds
- Intersection 2 offset: 55 seconds
- Intersection 3 offset: 45 seconds
- Intersection 4 offset: 115 seconds



Site C
- Arterial on NC 55 in Apex, NC
- 7 signalized intersections
- One un-signalized intersection
- 12 driveways
- 7,225 ft. total route length
- Selected plan: AM peak plan
- Cycle Length: 160 seconds
- Intersection 1 offset: 95 seconds
- Intersection 2 offset: 86 seconds
- Intersection 3 offset: 72 seconds
- Intersection 4 offset: 51 seconds
- Intersection 5 offset: 56 seconds
- Intersection 6 offset: 15 seconds
- Intersection 7 offset: 0 seconds

Figure 4-7 Data Collecting Sites

4.2 Data Monitoring Results

Signal system operation log data (split monitor data log) for selected pilot study site's (site B) dynamic green and programmed green comparison results are introduced in this section. In addition, early return to green and green extension distribution as well as difference between programmed g over C and field observed g over C values is explained. These monitoring result support the limitations of current bandwidth optimization method since most of studies rely on static green or observed average green durations.

4.2.1 Site "B" Programmed Signal Timing Plan

The US 70 arterial in Garner, NC consists of four intersections: Jessup Dr, Timber Dr, Garner Towne Square, and Yeargan Rd. intersections. All weekdays (Monday to Friday) have the same time of day plan: AM Peak, Midday, and PM peak plans. The intersection spacing is 1,080ft, 990ft and 1,770ft from Jessup to Yeargan. Table 4.2 shows US 70 Garner arterial time of day plan.

Table 4.2 US 70 Arterial in Garner Time of Day Plan

					Offset (s)				
Monday to Friday	Plan	Plan Start Time	Plan End Time	Cycle (s)	Jessup	Timber	Garner	Yeargan	Plan Number
	1	6:15:	11:00	120	62	55	45	115	10-1
	2	11:00	16:00	170	0	10	155	70	23-1
	3	16:00	19:00	170	145	0	10	15	21-1

A common cycle length of 120 seconds is used for the AM peak plan. For the midday and PM peak plans the common cycle length is 170 seconds. The Time Of Day (TOD) plan starts at 6:15 AM and ends at 19:00 PM. Table 4.3through

Table 4.6 show intersection signal timing parameters (cycle, split and phase sequence) for the US 70 arterial. The coordinated phases are phase 2 and 6 which are East and West directions.

Table 4.3 Jessup Dr. Intersection Programmed Signal Timing

Plan Number	Phase Sequence		Split (s)								Cycle (s)
	Phase 1	Phase 5	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	
10-1	Lag	Lead	20	75	0	25	20	75	0	25	120
23-1	Lead	Lead	25	125	0	20	15	135	0	20	170
21-1	Lead	Lead	30	120	0	20	20	130	0	20	170

Table 4.4 Timber Dr. Intersection Programmed Signal Timing

Plan Number	Phase Sequence		Split (s)								Cycle (s)
	Phase 1	Phase 5	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	
10-1	Lead	Lead	15	50	35	20	15	50	15	40	120
23-1	Lag	Lead	20	77	25	48	16	81	48	25	170
21-1	Lead	Lag	20	77	25	48	16	81	48	25	170

Table 4.5 Garner Towne Square Intersection Programmed Signal Timing

Plan Number	Phase Sequence		Split (s)								Cycle (s)
	Phase 1	Phase 5	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	
10-1	Lead	-	20	75	0	25	0	95	0	0	120
23-1	Lead	-	25	120	0	25	0	145	0	0	170
21-1	Lead	-	25	120	0	25	0	145	0	0	170

Table 4.6 Yeargan Rd. Intersection Programmed Signal Timing

Plan Number	Phase Sequence		Split (s)								Cycle (s)
	Phase 1	Phase 5	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7	Phase 8	
10-1	Lag	Lead	25	60	0	35	17	68	0	35	120
23-1	Lag	Lead	25	105	0	40	25	105	0	40	170
21-1	Lead	Lag	25	110	0	35	25	110	0	35	170

4.2.2 Coordinated Phase g over C

Intersection capacity is determined by g/C ratio and saturation flow rate (saturation headway). The capacity of a given lane group is:

$$c = N * s * \frac{g}{C}$$

Where, c: lane group capacity,

N: Total number of lanes for the given lane group,

s: Adjusted saturation flow rate for the given lane group,

g: Effective green time for the given movement, and

C: Cycle length of the intersection.

A semi-actuated controller does not allocate a green signal to side streets as long as there is no demand for it. The controller allows skipping or gapping out of that specific phase so that the unused green time is allocated to the major street. This is called an “early return to green.” Also, the coordinated phases green can be extended until the yield point or next reference point. Therefore, each cycle green duration is dynamic. Therefore, programmed

greens of coordinated phases indicated minimum amount of green duration for coordinated movements. However, programmed effective green value (minimum effective green value) is conventionally used for estimating lane group capacity, intersection signal failure, and level of service due to absence of field observed effective green information. Table 4.7 shows the programmed g/C ratio and average of split monitor g/C (over six weekdays) for Timber Dr. intersection, which is the critical intersection on US 70 arterial in Garner, NC. In case of US 70 Timber Dr. intersection, 23% more green was provided compare to programmed green during AM plan. 18% and 20% more green was provided during MD plan for phase 2 and phase 6, respectively. During PM plan, 10% more green was provided for phase 2 and 12 % of more green was provide for phase 6.

Table 4.7 Timber Dr. Intersection Programmed g/C and Field g/C Comparison

	AM		**MD**		**PM**	
Phase	**Phase 2 (EB)**	**Phase 6 (WB)**	**Phase 2 (EB)**	**Phase 6 (WB)**	**Phase 2 (EB)**	**Phase 6 (WB)**
Cycle	120		170		170	
Sample size	827 cycles		626 cycles		375 cycles	
Average of Used Green	55.08	55.31	84.42	91.5	77.88	83.6
Split Monitor g/C	0.46	0.46	0.5	0.54	0.46	0.49
Programmed Green	42	42	69	73	69	73
Programmed g/C	0.35	0.35	0.41	0.43	0.41	0.43

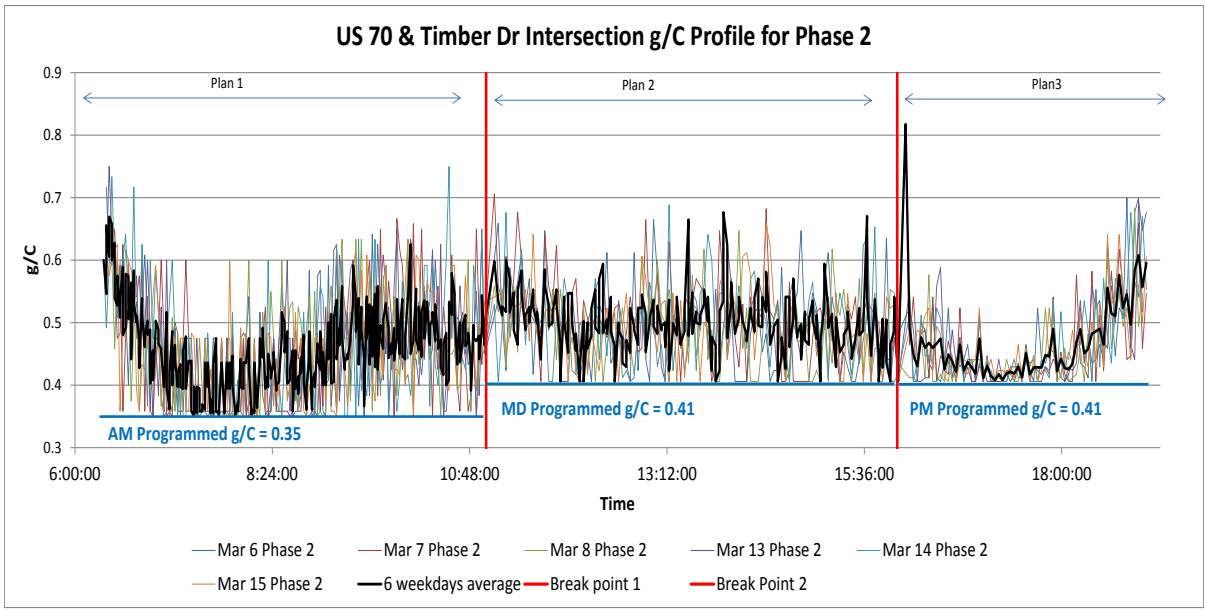


Figure 4-8 Phase 2 g/C Profile for Timber Dr. Intersection

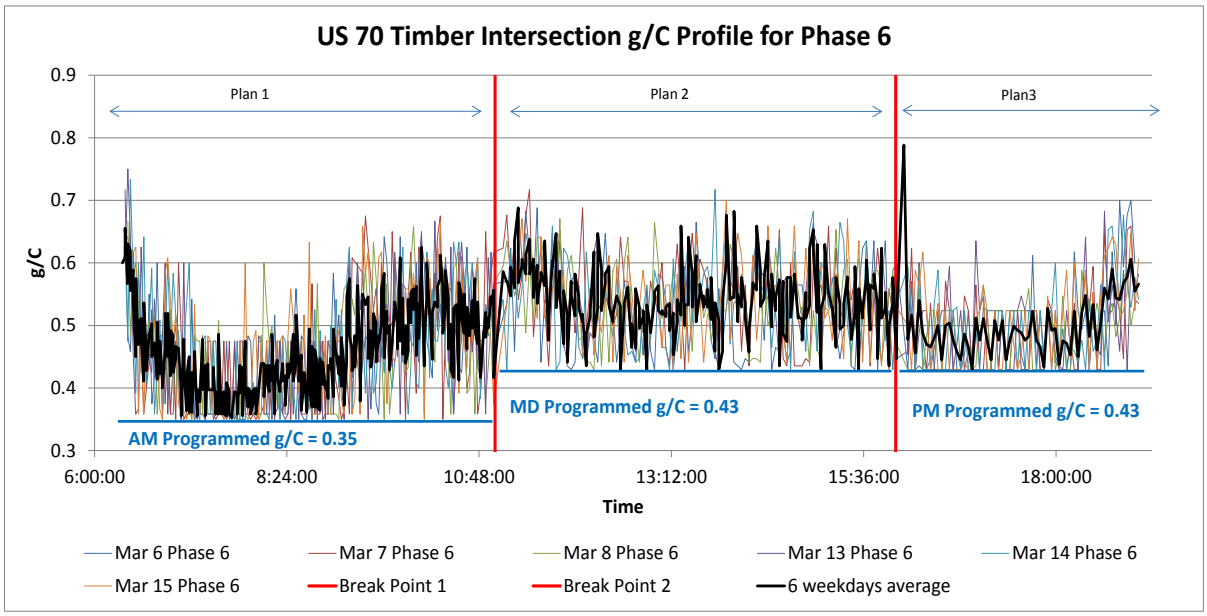


Figure 4-9 Phase 6 g/C Profile for Timber Dr. Intersection

Figure 4-8 and Figure 4-9 show g over C profile plots over six weekdays. The black lines indicate coordinated phase's average profile. The g over C plot illustrates that even critical intersection has considerably larger field g over C than its programmed g over C.

Table 4.8 shows programmed and dynamic g/C ratios on Jessup Dr. intersection (non-critical intersection). Dynamic green was 38% more than programmed green for phase 2 and 53% for phase 6 in the AM plan. 19% and 21% more green was provided during MD plan for phase 2 and phase 6, respectively. During PM plan, 21% more green was provided for phase 2 and 22% more green was provide for phase 6.

Table 4.8 Jessup Dr. Intersection Programmed g/C and Field g/C Comparison

	AM		**MD**		**PM**	
Phase	**Phase 2 (EB)**	**Phase 6 (WB)**	**Phase 2 (EB)**	**Phase 6 (WB)**	**Phase 2 (EB)**	**Phase 6 (WB)**
Cycle	120		170		170	
Sample size	815 cycles		622 cycles		361 cycles	
Average of Used Green	96	106	143	153	137.7	151.3
Split Monitor g/C	0.797	0.88	0.84	0.90	0.81	0.89
Programmed Green	69	69	119	129	114	124
Programmed g/C	0.575	0.575	0.7	0.758	0.67	0.729

Figure 4-10 and Figure 4-11 shows g over C profile for coordinated phases in Jessup Dr. intersection. The g over C plot illustrates that the coordinated movements' lane group on non-critical intersection has considerably more g over C ratio compared to the programmed greens.

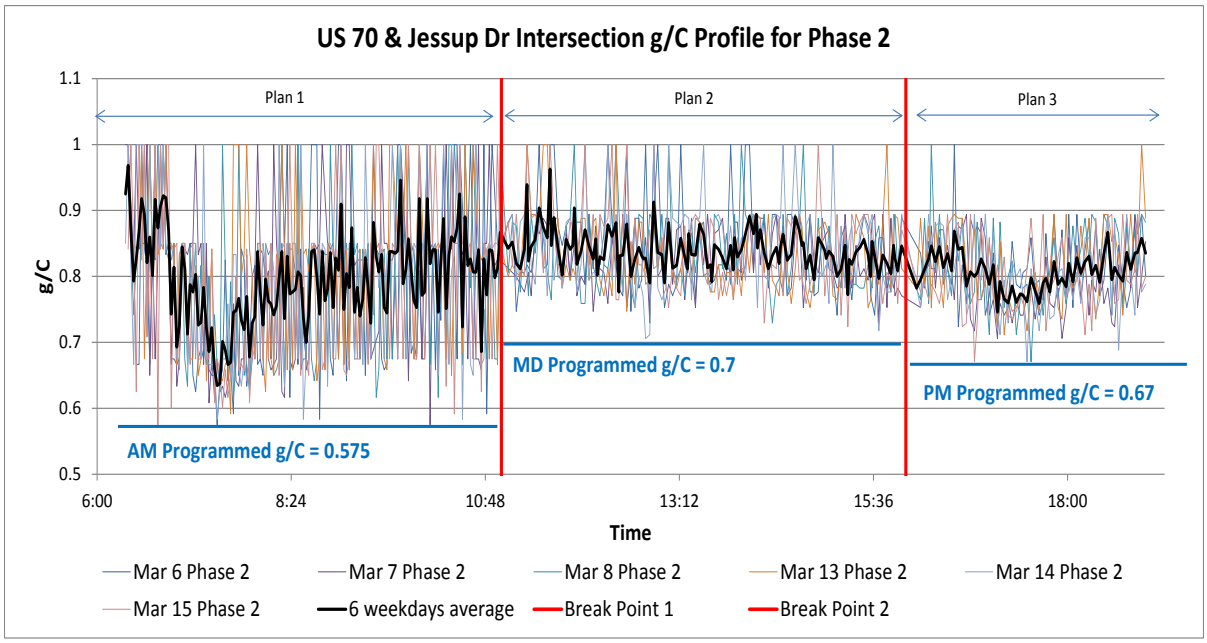


Figure 4-10 Phase 2 g/C Profile for Jessup Dr. Intersection

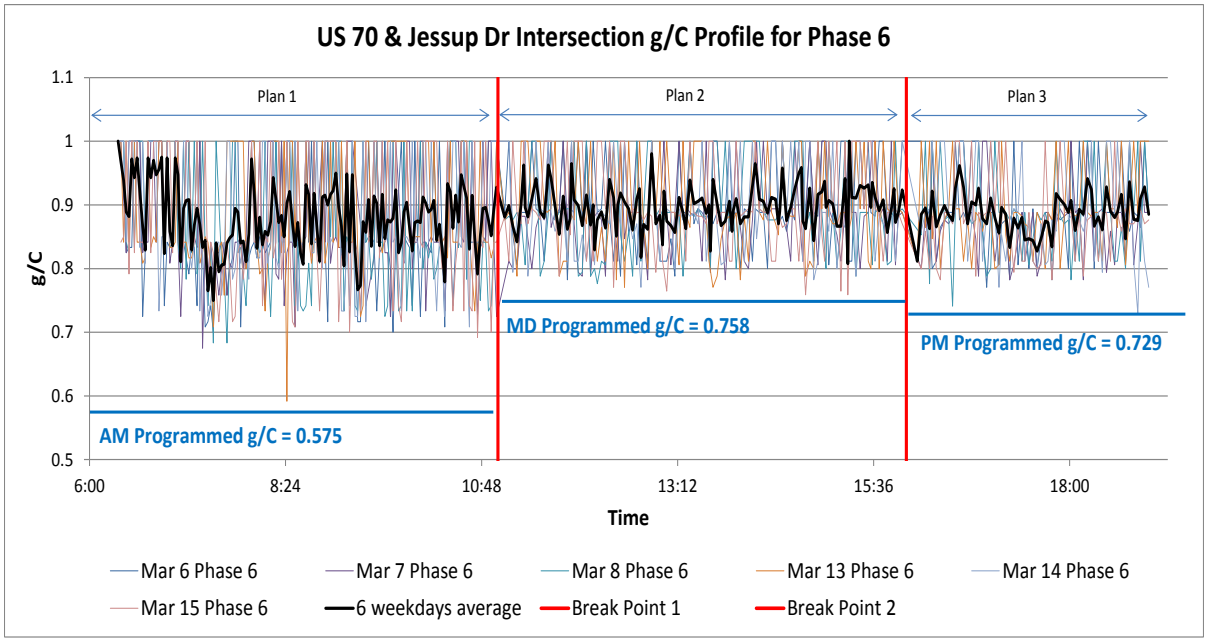


Figure 4-11 Phase 2 g/C Profile for Jessup Dr. Intersection

The average g over C line for six weekdays shows that it never reaches its low boundary, which is the programmed g over C. In addition, even single day's plot also hardly hits the programmed g over C line. The g over C plot gives important information about coordinated movement lane group's capacity. According to findings, coordinated movement lane group's capacity is much higher than its programmed capacity.

4.2.3 Early Return to Green Distribution

Semi-actuated signal control arterials have the potential for the coordinated phase to begin earlier than expected. This phenomenon happened when non-coordinated phases are gaped out or skipped. This early return to green may increase system delay when this was out of the field engineer's prospect ranges. During offset fine tuning procedure, signal engineers consider how often and how many seconds early return to green occurs in the field.

However, it is hardly possible to select one number, which can nicely represent specific intersection's early return to green. This is because each intersections' early return to green is independent from any upstream or downstream intersection flow, but only depends on its own non-coordinated approach traffic flow and arrival pattern. In other words, each intersection has a different early return to green distribution.

Figure 4-12 and Figure 4-13 show the distribution of early return to green for coordinated phases for 824 cycles at Timber Dr. intersection. Although the Timber Dr. intersection is a critical intersection, there are only 9.1% and 7.2% of zero early return to green. For more than 90% of time during the AM period, Timber intersection has early return to green while there is zero green extension.

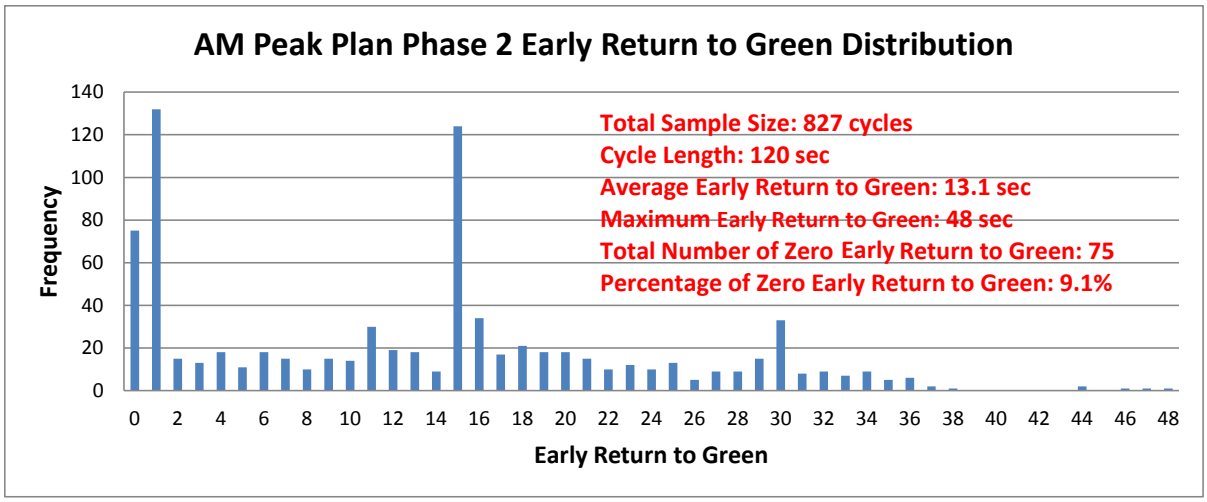


Figure 4-12 Timber Dr Intersection Phase 2 Early Return to Green Distribution

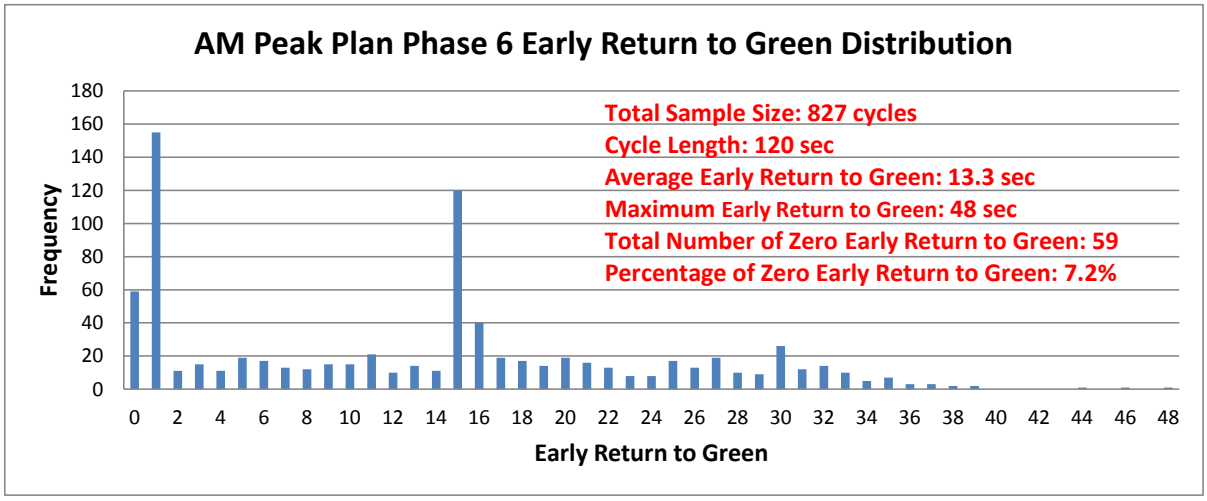


Figure 4-13 Timber Dr Intersection Phase 6 Early Return to Green Distribution

Figure 4-14 and Figure 4-15 illustrates Jessup Dr. intersection's (non-critical intersection) early return to green distribution. There are more zero early return greens compare to Timber Dr. intersection. However, in case of phase 2 (EB), there are 200 zero early return to green but 119 cycles are fully extended. So, there is only 1 zero early return to

green and the percentage is 0.12%. In case of phase 6, all of 286 cycles of zero early return to green happened due to full green extension, so there are no pure zero early return to green.

The early return to green distribution is multi-modal and is affected by conflict phase gap out or skips. As mentioned above, it is difficult to pick up one number that can nicely represent a specific intersection's early return to green.

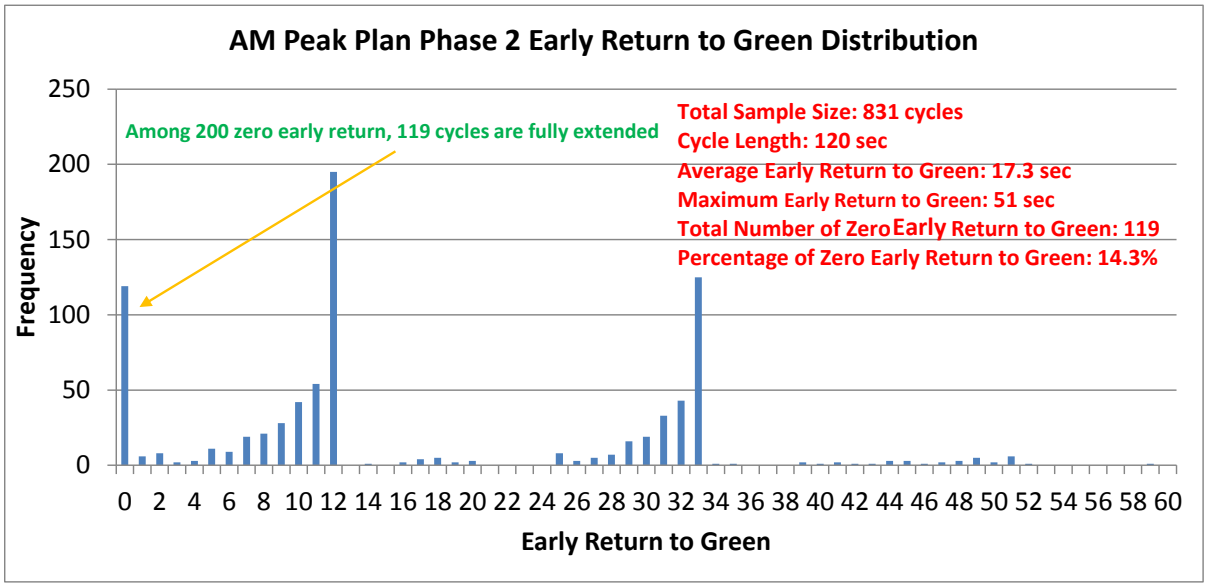


Figure 4-14 Jessup Dr Intersection Phase 2 Early Return to Green Distribution

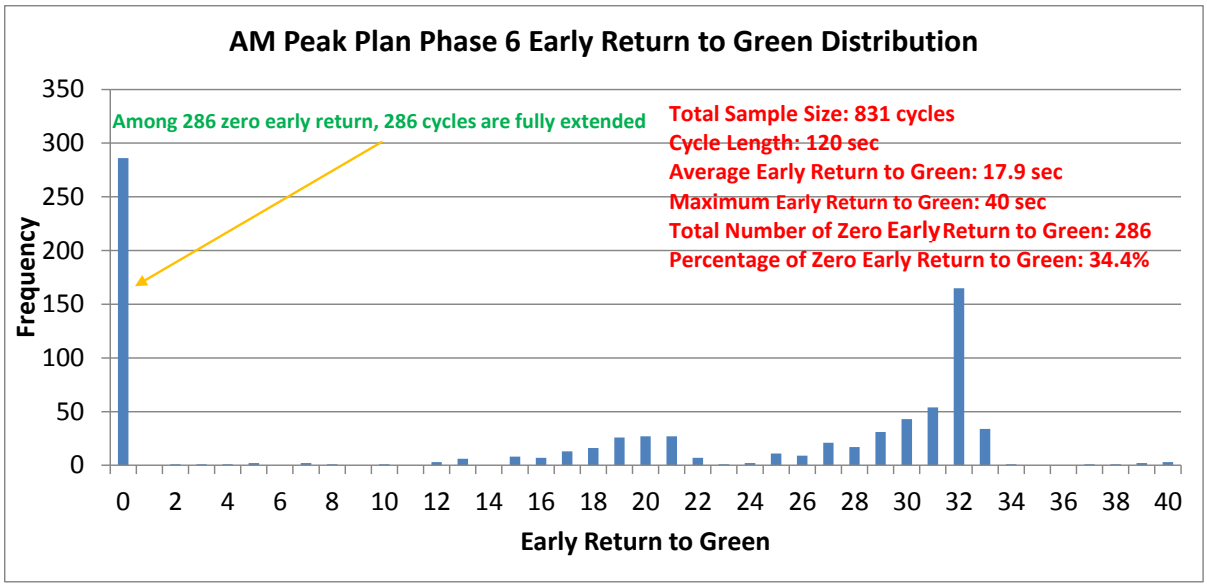


Figure 4-15 Jessup Dr Intersection Phase 6 Early Return to Green Distribution

4.2.4 Non-Coordinated Phase Used Green Distribution

Table 4.9 shows one example of displayed green time of a non-coordinated phase. More than 37 % of time, phase 1 is skipped due to an absence in demand and it used a minimum green time of 7 seconds for 42.8% of time. The Programmed split for phases 1 and 5 are 15 seconds at Timber intersection with 4.0 and 3.0 seconds yellow and all-red indications during the AM peak period, respectively. Therefore, when OASIS displayed green time for more than 9 seconds, phase 1 and phase 5 started earlier than their programmed start time. More than 19 % of time, the Timber Dr. intersection phase 1 is forced off while 3.4% of time, phase 5 is forced-off during the AM peak period. Figure 4-16 shows the distribution of displayed green time for phase 1 and 5 on the Timber Dr.

intersection. Most of the time, both phase 1 and phase 5 are skipped or only used their minimum green time.

Table 4.9 Timber Intersection Phase 1 and 5 Displayed Green Time

	Phase 1		Phase 5	
Displayed Green Time	**Frequency**	**Percentage**	**Frequency**	**Percentage**
0	312	37.7%	315	38.1%
7 (minimum green)	354	42.8%	404	48.9%
8	95	11.5%	80	9.7%
9	22	2.7%	9	1.1%
10	18	2.2%	10	1.2%
11	11	1.3%	3	0.4%
12	7	0.8%	3	0.4%
13	4	0.5%	1	0.1%
14	1	0.1%	1	0.1%
15	2	0.2%	0	0.0%
16	0	0.0%	1	0.1%
17	1	0.1%	0	0.0%
Total	827	100%	827	100%

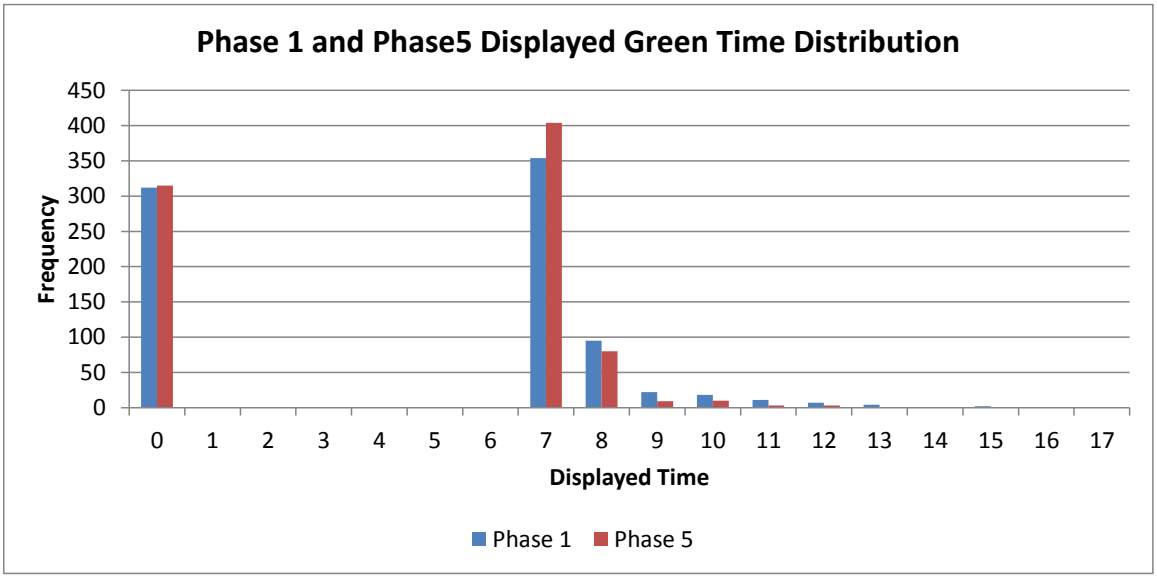


Figure 4-16 Phase 1 and Phase 5 Displayed Green Distribution on Timber Dr. Intersection

Figure 4-17 shows the distribution of displayed green time for both phase 1 and 5 on the Jessup Dr. intersection. The Jessup intersection used "Lag-Lead" phase for left turn phase sequences. Phase 1 used a lag phase so it cannot gap out until phase 6 is terminated. Therefore, most of time (63.9% of time), phase 1 used all of its programmed time while phase 5 was maxed-out only 0.12%.

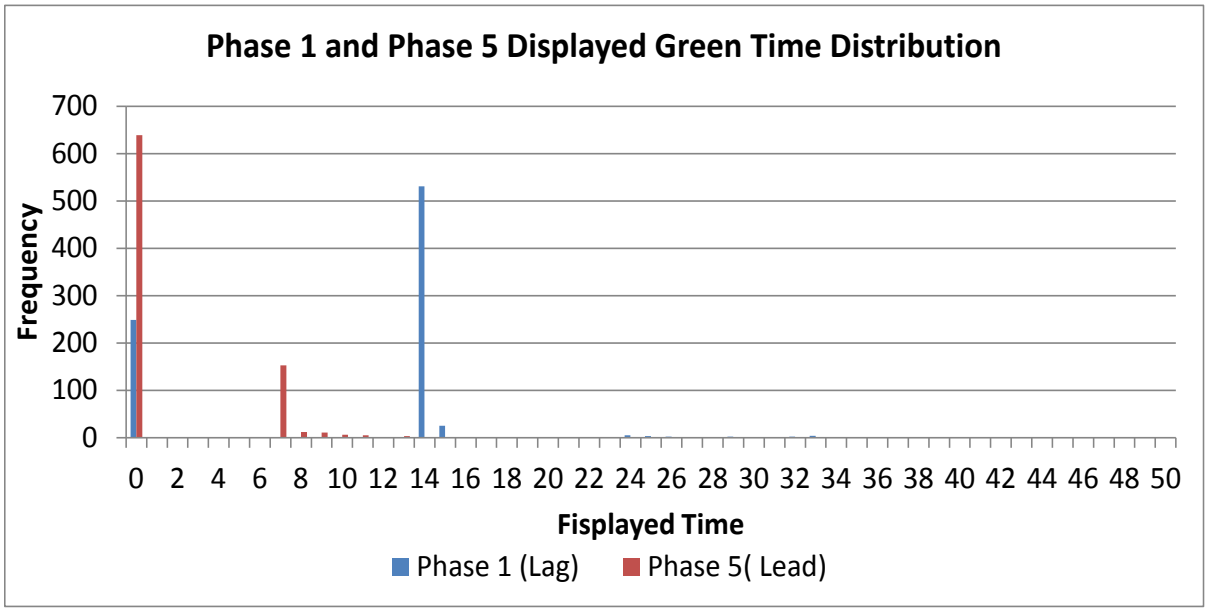


Figure 4-17 Displayed Green Distribution on Jessup Dr. Intersection for AM Plan

This result highlights the disadvantages of a "Lead-Lag" phase sequence. Since lag phase cannot gap out, lag phase will show green ball even without any demand until concurrent phase is terminated.

Jessup Dr. intersection used a “Lead-Lead” phase sequence during the PM peak plan. Phase 1 split is 30 seconds and phase 5 split is 20 seconds. The minimum green is 7 seconds for both phases 1 and 5. Figure 4-18 shows displayed green time in OASIS split monitor for phases 1 and 5 during the PM peak plan (365 cycles). Phase 1 used its minimum green (7 seconds) 16.2 % of the time. It is difficult to estimate the total amount of time that the “lag” phase turned on green without any demand, but the frequency of skipping phase 1 on Figure 4-17 and Figure 4-18 give an idea of inefficiencies of a lag phase. The percentage of skipping during the AM plan for phase 1 is 36.1% while phase 1 only skipped 3.88% of time during the PM plan.

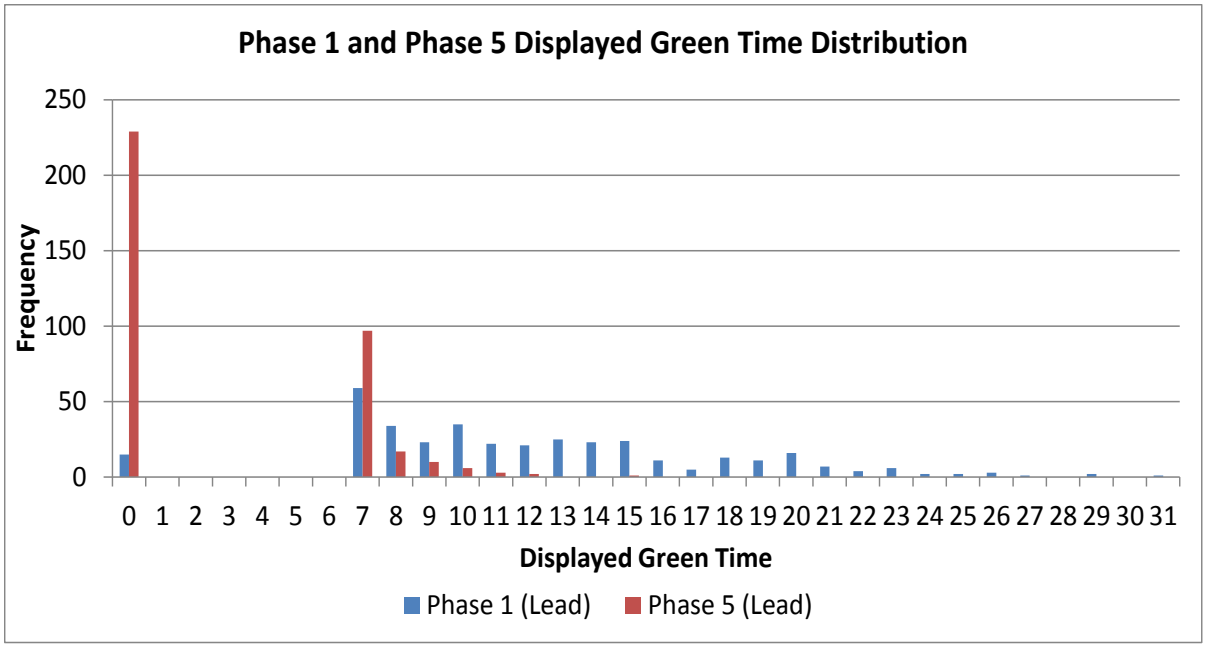


Figure 4-18 Displayed Green Distribution on Jessup Dr. Intersection for PM Plan

4.2.5 Bandwidth Comparison

Unlike freeway facilities where delay results primarily from capacity constrained bottlenecks, delay along signalized arterials whose demand does not exceed capacity results from deceleration and stops as vehicles interact with signal control. Therefore mitigating delay along signalized arterials through improved signal timing is a cost effective congestion management approach that can be accomplished without adding physical roadway capacity.

Well-designed signal coordination along arterial streets minimizes the number of stops and consequently travel delay. Synchronizing the onset of green indication for the intersections along an arterial street is one of the key steps in improving coordination and is known as offset optimization. However, most of current offset optimization methods rely on programmed greens and reds, while arterial coordinated phases' green is a dynamic and time dependent variable. From OASIS split monitor information, it is possible to create real-world cycle-by-cycle bands, hereafter called "Dynamic Bandwidth".

4.2.5.1 Conventional Bandwidth

The Federal Highway Administration Traffic Signal Timing Manual defines Bandwidth as "the maximum amount of green time for a designated movement as it passes through a corridor". Traditional bandwidth is decided by ideal travel time, offset, and constant green and is constant in all cycles.

The Monday to Friday AM plan's programmed outbound bandwidth is 39 seconds and inbound bandwidth is 43 seconds, which is calculated by programmed green with current offset. Therefore, outbound bandwidth efficiency is 32.5% and inbound bandwidth efficiency

is 35.83%. The arterial's two-way bandwidth efficiency is 34.16%. Figure 4-19 and Figure 4-20 show outbound and inbound programmed bands.

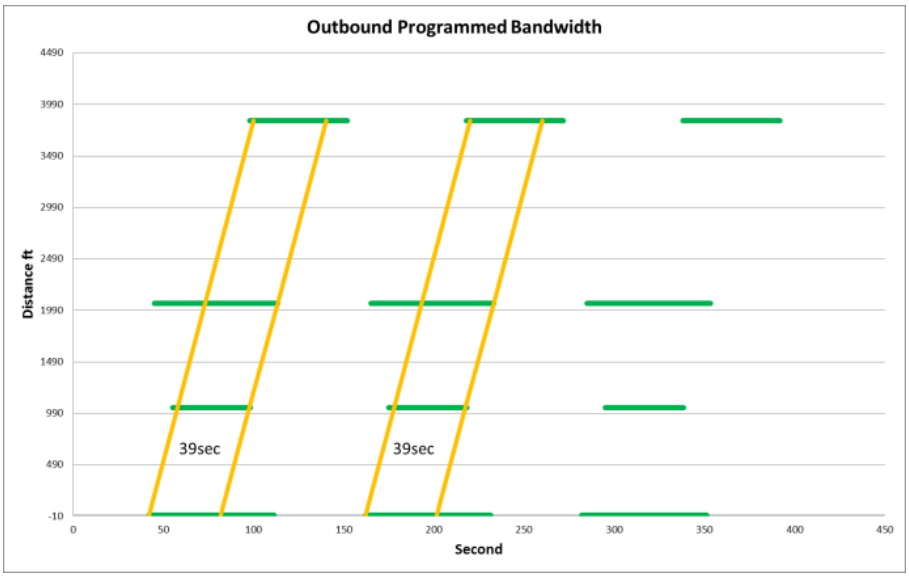


Figure 4-19 AM Plan Outbound Programmed Bandwidth for Site "B"

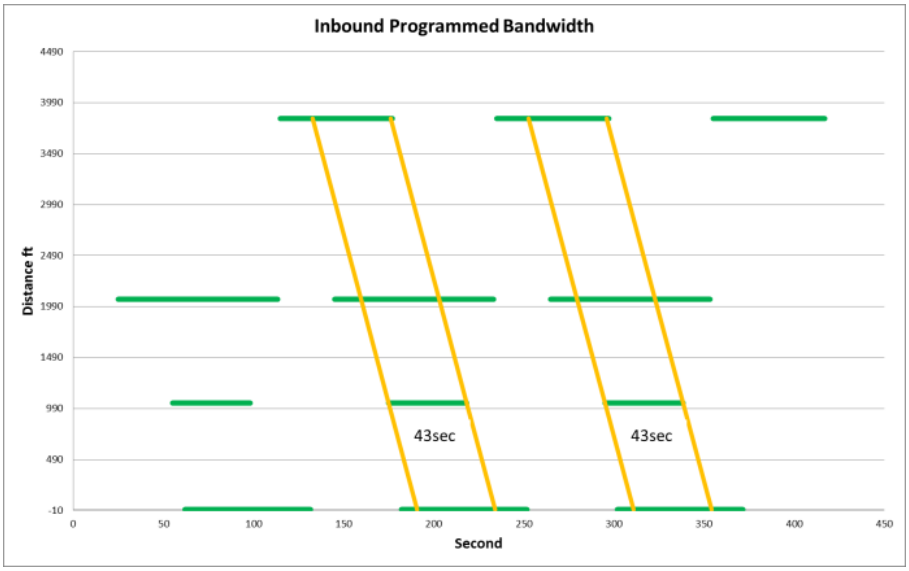


Figure 4-20 AM Plan Inbound Programmed Bandwidth for Site "B"

4.2.5.2 Dynamic Bandwidth

Dynamic bandwidth should be determined using intersection dynamic green durations. Figure 4-21 shows an example of cycle-by-cycle real-world bandwidth on US 70 arterial in Garner, NC. Each intersection's cycle-by-cycle green duration is calculated from OASIS's split monitor log and cycle-by-cycle bandwidths are calculated manually.

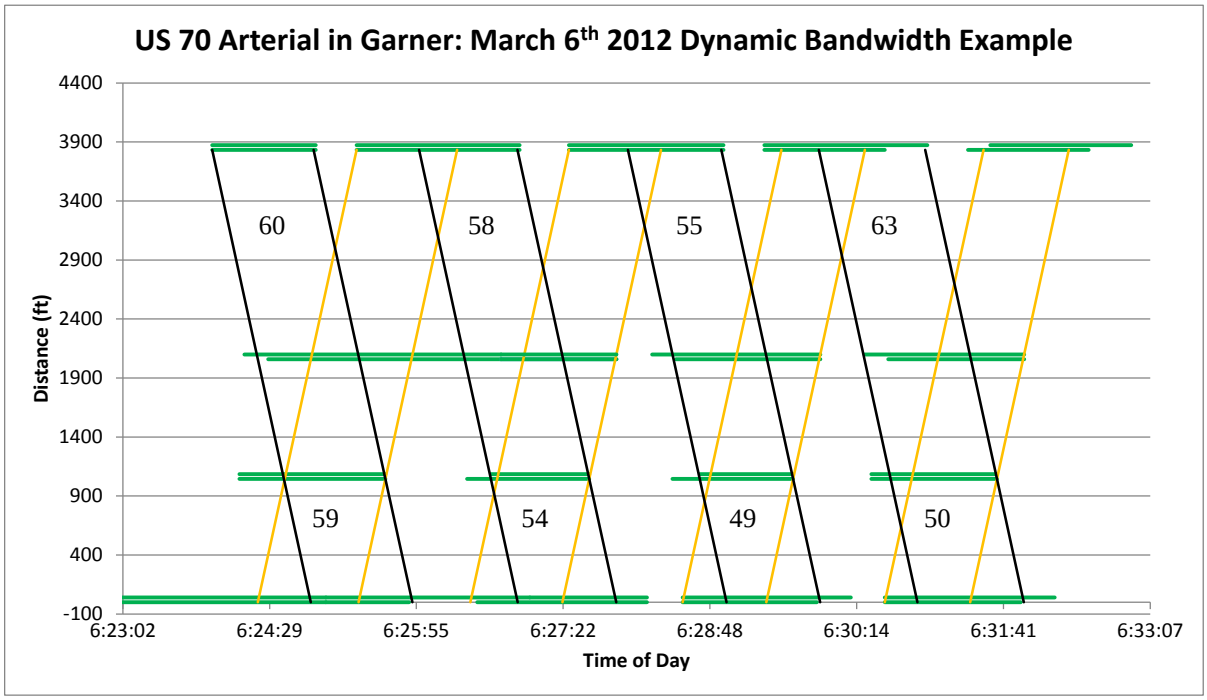


Figure 4-21 Dynamic Bandwidth for Site "B"

Table 4.10 shows corresponding dynamic band sizes. All the bandwidths are larger than the programmed bandwidth as expected.

Table 4.10 Cycle-by-Cycle Dynamic Bandwidth for Site “B”

Outbound Bandwidth	Inbound Bandwidth
59	60
54	58
49	55
50	63

Figure 4-22 and Table 4.11 show “Secondary Bandwidth” which is smaller than primary bandwidth and is created by early return to green and green extension combinations. The programmed bandwidths are 44 seconds and 39 seconds for outbound and inbound, respectively. All the dynamic bandwidths should be greater than or equal to programmed bandwidth, except secondary bands.

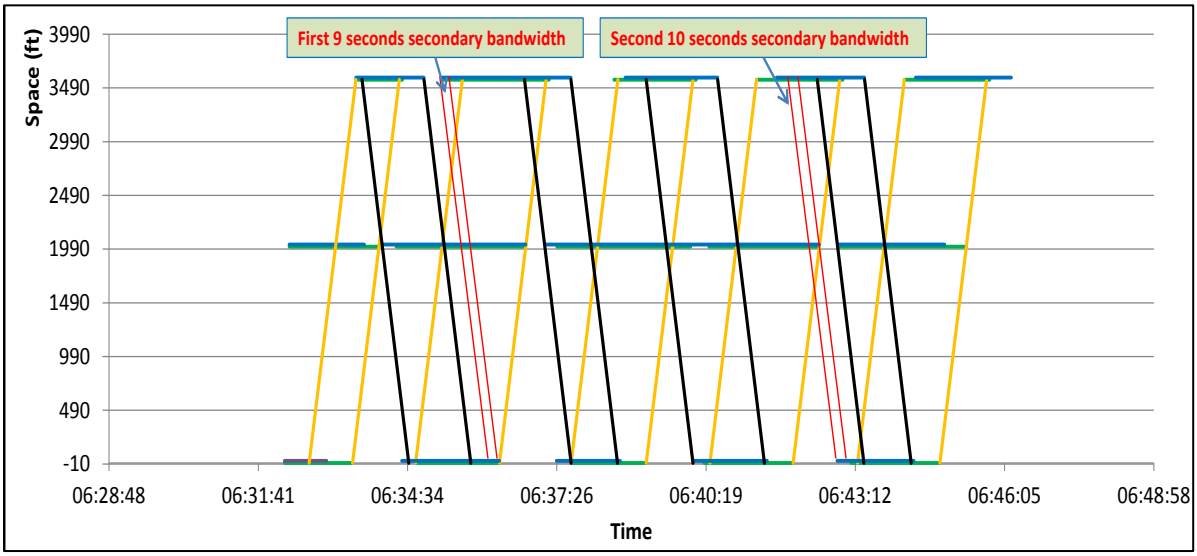


Figure 4-22 Dynamic Bandwidth for Site “A”

Table 4.11 Cycle-by-Cycle Dynamic Bandwidth for Site "A"

Outbound Bandwidth	Inbound Bandwidth
44	77
94	9
87	58
90	85
89	10
	59

Since cycle-by-cycle green durations are dynamic, provided bandwidth should be dynamic and there is possibility to have secondary bandwidth and non-programmed bandwidth (i.e. a band that is not programmed by traffic engineer). However, manually calculating and monitoring all dynamic bandwidth is difficult and time consuming. In addition, there is no tool to allow monitoring cycle-by-cycle bandwidth. This study developed such a tool called Dynamic Bandwidth Analysis Tool (DBAT).

4.3 Dynamic Bandwidth Analysis Tool (DBAT)

Dynamic Bandwidth Analysis Tool (DBAT) was developed for monitoring near real-time arterial dynamic bandwidth and exhaustive search for all feasible offset solution. Departing from the traditional method of defining bandwidth size using ideal travel time, offsets, and programmed green, the dynamic bandwidth is determined for each cycle using the actual coordinated green durations. Developed DBAT C++ code is introduced in Appendix B.

4.3.1 Processing OASIS™ Split Monitor Log Raw Data

Departing from the traditional method of defining bandwidth size using ideal travel time, offsets, and programmed green, the dynamic bandwidth is determined for each cycle using the actual coordinated green durations. An arterial street's bandwidth over the given sample of recorded cycles is the summation of the individual bandwidths determined for each cycle, as shown Figure 4-23.

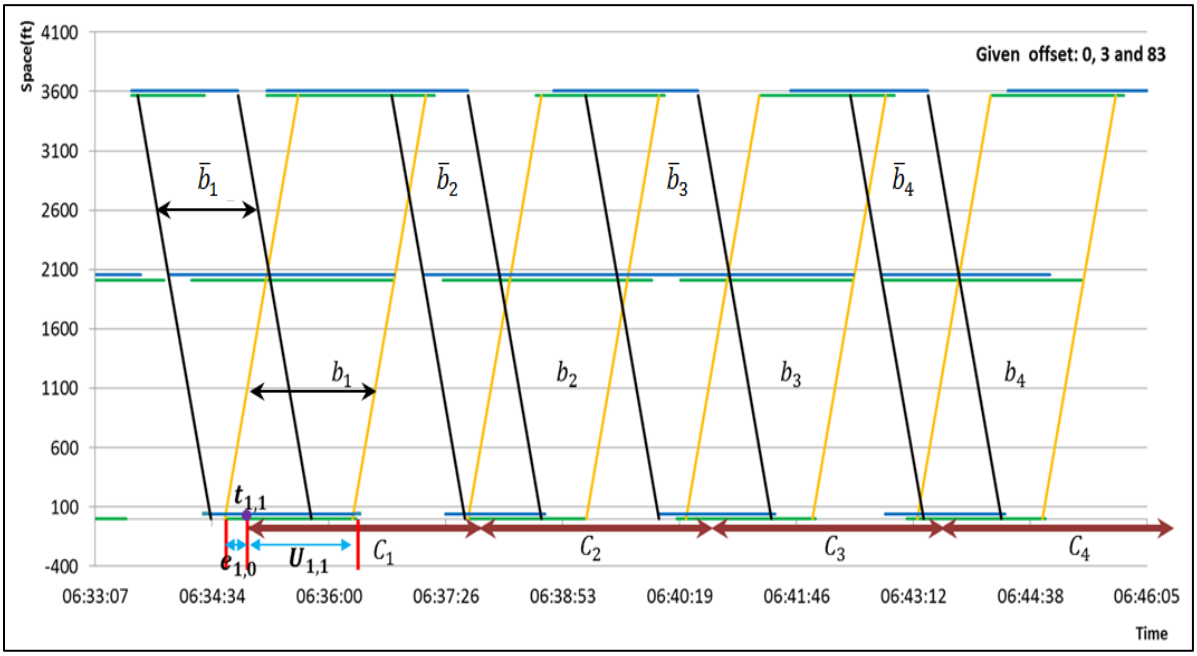


Figure 4-23 Dynamic Bandwidth Example

The following data items from the OASIS split monitor log are key to establishing the actual duration of each cycle's coordinated greens.

- SMTimeStamp: Start time of each cycle (reference point)

- ExtraTimeCP: Remaining time in the current cycle, which is going to be used by the coordinated phase (Used green time before reference point)
- UsedPhase: Actual green time, which is used in the current cycle for each phase (Used green time after reference point)

Figure 4-24 shows split monitor data for two cycles. In this example, the left-turn phase sequence is Lead-Lead so phases 2 and 6 start together. However, the "ExtraTime" for phase 2 has 13 seconds and "ExtraTime" for phase 6 has 27 seconds in the first cycle's. Therefore, phase 2 start 13 seconds earlier and phase 6 starts 27 seconds later than reference point (which is 6:34:54) in the second cycle. In addition, phase 2 and phase 6 in the second cycle will be terminated 42 seconds from the reference point.

					Used Phase							
SMTimeStamp	Cycle	Offset	Ex Time Phase 2	Ex Time Phase 6	1	2	3	4	5	6	7	8
10/17/2011 6:32:55	120	55	13	27	7	42	18	11	0	42	8	21
10/17/2011 6:34:54	120	55	16	30	7	42	19	7	0	42	0	34

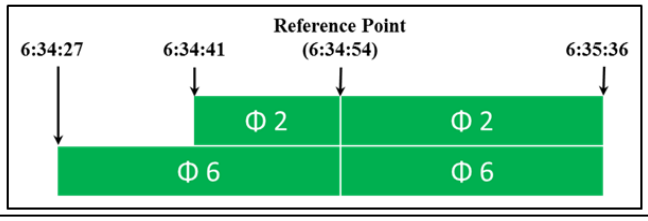


Figure 4-24 Split Monitor Example

Let $I, J,$ and K respectively denote the set of all intersections in the arterial system, all cycles, and both directions of traffic. Let $x_{i,j,k}$ and $y_{i,j,k}$ respectively represent the start and end time of the green phase in cycle $j \in J$ at intersection $i \in I$ in direction $k \in K$; and $t_{i,j}$ be the reference point (SMTimeStamp) in cycle $j \in J$ at intersection $i \in I$. Let $e_{i,j,k}$ (ExtraTimeCP) denote the time difference between the start time of through movement green duration in cycle $j \in J$ for direction $k \in K$ at intersection $i \in I$ and the reference point. Let $u_{i,j,k}$ (UsedPhase) denote the amount of used green time started from $t_{i,j}$ in cycle $j \in J$ at intersection $i \in I$ for direction $k \in K$. The cycle-by-cycle dynamic green is defined by following equations:

$$x_{i,j,k} = t_{i,j} - e_{i,j-1,k},\ \forall i \in I, \forall j \in J, \forall k \in K$$

$$y_{i,j,k} = t_{i,j} + u_{i,j,k}, \quad \forall i \in I, \forall j \in J, \forall k \in K$$

4.3.2 Development of DBAT

DBAT was developed in the C++ programming language. When DBAT reads input data from a dialog box, it generates each intersection's actual coordinated green time as an array. This process requires defining each intersection's green start and end time. The data processing equations shown above are an OASIS^{TM} specific example. For different traffic control software, these equations would need to be modified to match the software's signal data archive convention, to provide accurate green time calculations and time referencing.

DBAT directly reads the split monitor log file and creates dynamic green durations for each intersection and then generates a Boolean array of each intersection dynamic green

and red. Afterwards it initializes the clock time to zero and rearranges each intersection's first green start time from the given offset value in the DBAT dialog box. By definition, DBAT sets the first intersection as the reference intersection. Therefore, the bandwidth search always starts at the first intersection and the first green. The minimum search interval is one second, and DBAT calculates bandwidth by counting the number of seconds that would accommodate a through vehicle trajectory traveling at the specified progression speed without encountering a red indication at any of the intersections. The progression speed (an input into the dialog box) is the slope of the vehicle trajectory. The rearranged Boolean array indicates each intersection's signal status.

DBAT creates a virtual vehicle trajectory for each second of the green durations at the first intersection of the system during the entire study period, T (which includes J cycles) and determines whether the trajectory reaches the downstream intersection during the green signal indication for the coordinated phase or during the red signal. If the trajectory intersects the red signal of the coordinated phase, the process is stopped, otherwise it continues to the downstream intersection. If the trajectory makes is through all intersections of the system, it is counted as a success and the same process is performed for the next time step (one second). The total number of successes in a row is the duration of a directional band.

This method can also identify secondary bands that may occur within a cycle as well as non-programmed bands that may arise in cases where the programmed green time provides no static bandwidth in one of the directions. Secondary bands, when they occur, result from a combination of early return green and/or green extension.

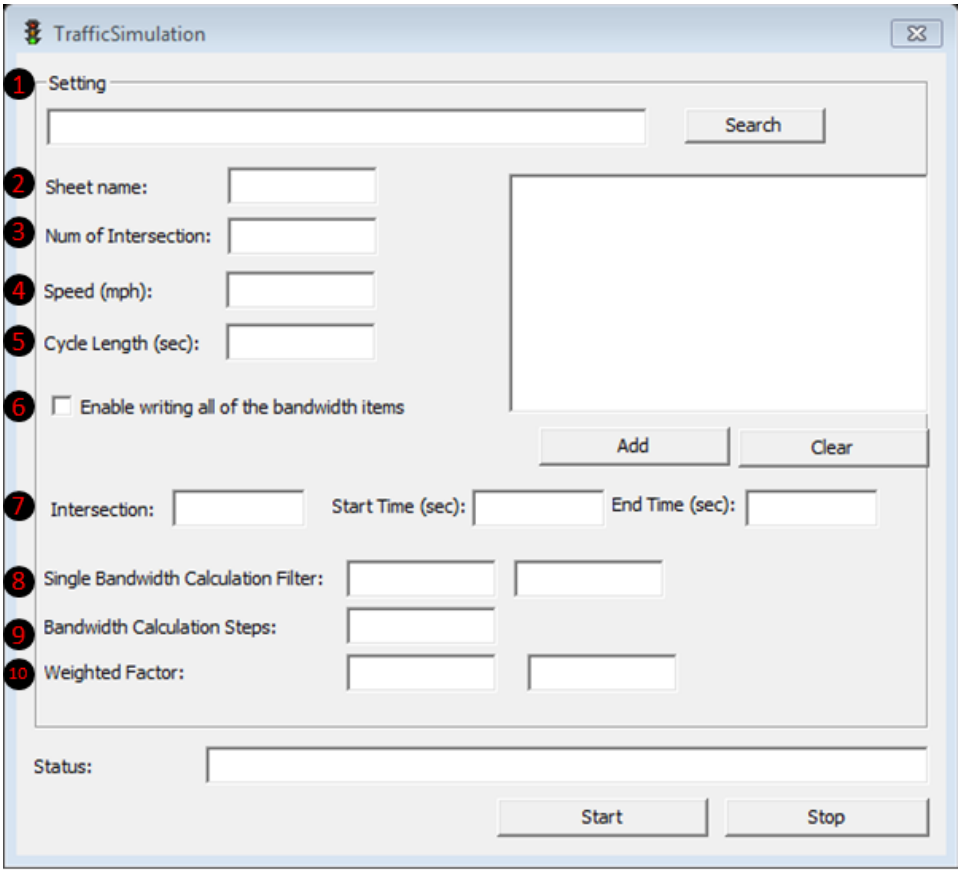


Figure 4-25 DBAT Interface

Figure 4-25 shows the DBAT dialog box. It requests 10 input data in the form. Input data 1 through 5 are essential information to calculate dynamic bandwidth. Input data 7 allows the user to set up the offset searching range and all possible offset combinations can be searched.

DBAT requires the following inputs:

- Free-flow speed or speed limit
- Total number of intersections

- Cycle length for analyzed time of day plan
- Order of intersection within search space
- Bandwidth search interval (minimum is 1 second)
- Directional bandwidth weighting factor

DBAT generates the following outputs:

- Inbound (outbound) sum of bands
- Inbound (outbound) number of bands
- Inbound (outbound) average band size
- Inbound (outbound) standard deviation of band size
- Inbound (outbound) band efficiency
- Bi-directional total sum of bands
- Bi-directional weighted total sum of bands
- Inbound (outbound) individual band per cycle (optional)

In order to calculate bandwidth efficiency, it is necessary to modify the original formulation since the dynamic bandwidth varies each cycle. Furthermore, bandwidth efficiency for dynamic bandwidth analysis has the capability to account for secondary bands and non-programmed bands as described above.

The conventional two-way bandwidth efficiency (CBE) formulation is as follows:

$$\text{CBE} = \frac{BW_1 + BW_2}{2C} * 100$$

Where: $C =$ cycle length in sec,

$BW_1 =$ outbound bandwidth in sec, and

$BW_2 =$ inbound bandwidth in sec.

The proposed two-way dynamic Bandwidth efficiency (DBE) formulation over a time period *T* is as follows:

$$\text{DBE} = \frac{\sum_{i=1}^{i=m} BW_i + \sum_{k=1}^{k=n} BW_k}{C * (M + N)} * 100$$

Where: $BW_i =$ cycle-by-cycle outbound dynamic bandwidth in sec,

$BW_k =$ cycle-by-cycle inbound dynamic bandwidth in sec,

$m =$ total number of outbound dynamic bands in time period *T*,

$n =$ total number of inbound dynamic bands in time period *T*,

$M =$ total number of outbound primary dynamic bands in time period *T*, and

$N =$ total number of inbound primary dynamic bands in time period *T*.

4.4 DBAT Evaluation

DBAT results were verified using three classic and well-understood two-way progression schemes: alternate progression, double alternate progression, and simultaneous progression. The test results confirmed that DBAT provided the correct bandwidth solution in each case.

4.4.1 Alternate Progression

For certain block lengths with 50:50 splits and uniform block lengths, the optimal offset for maximizing bandwidth is a half cycle.

$$\frac{C}{2} = \frac{L}{S} = \text{Ideal offset}$$

Where: $C = \text{Cycle length, sec}$

$L = \text{block length, ft}$

$S = P\text{rogression speed, fps}$

To test DBAT, the following condition is created:

- Three signalized intersections on an arterial,
- Intersection spacing is 2,000 ft,
- Cycle length is 80 seconds,
- Effective green is 40 seconds for all intersections, and
- Progression speed is 50 fps.

Under the above scenario, the offset for the optimum bandwidth solution is 0, 40 and 0 seconds from first to last intersection, respectively. In addition, directional bandwidth is 40 seconds. The DBAT input file is created under the above scenario conditions and Table 4.12 shows the DBAT input file form.

Table 4.12 Input Data for Alternate Progression Scenario Test

Intersession A					
Time	Distance	ExtraTimeCP1	ExtraTimeCP2	UsedPhase2	UsedPhase6
7:00:00	0	0	0	40	40
7:01:20	2000	0	0	40	40
7:02:40		0	0	40	40
Intersession B					
Time	Distance	ExtraTimeCP1	ExtraTimeCP2	UsedPhase2	UsedPhase6
7:00:00	2000	0	0	40	40
7:01:20	2000	0	0	40	40
7:02:40		0	0	40	40
Intersession C					
Time	Distance	ExtraTimeCP1	ExtraTimeCP2	UsedPhase2	UsedPhase6
7:00:00	2000	0	0	40	40
7:01:20	0	0	0	40	40
7:02:40		0	0	40	40

From the input signal data, DBAT provides the following results with offsets 0, 40 and 0 as shown in Table 4.13. The “Band1” column expresses the total sum of bandwidth for phase 2 and the “Band2” column is the sum of bandwidth for phase 6. The “# of Band” column is the total number of directional bands and the “Ave Band” column is the sum of Bandwidth divided by “# of Band”. The DBAT test result exactly matches the alternate progression case example.

Table 4.13 DBAT Test Results of Alternate Progression

Int.1	Int.2	Int.3	Band1	Band2	# of Band1	# of Band2	Ave Band1	Ave Band2	Sum of band1, band2
0	40	0	80	80	2	2	40	40	160

The DBAT result shows that each direction made two bands and the average bandwidth size was 40 seconds. However, Table 4.13 does not provide cycle-by-cycle bandwidth size. When the user clicks the check box (Enable writing all of the bandwidth items), DBAT provides a summary table as well as cycle-by-cycle bandwidth. The result of the cycle-by-cycle bandwidths is 40 seconds for both directions similar to Figure 4-26.

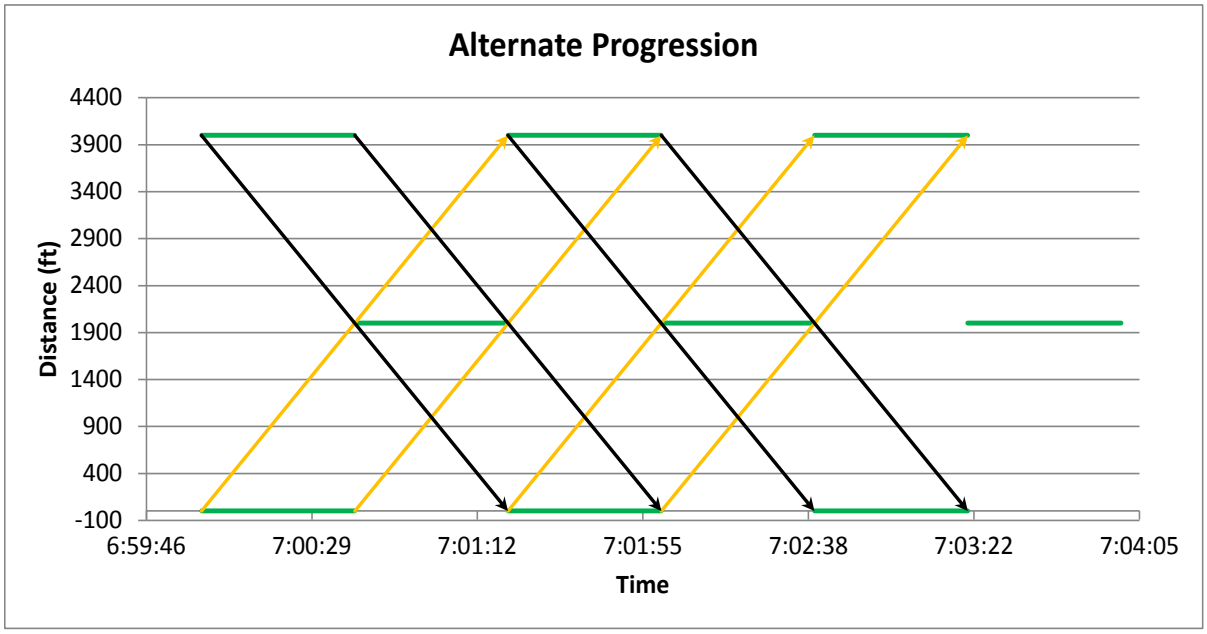


Figure 4-26 DBAT Test Result of Alternate Progression Case

4.4.2 Double Alternate System

For certain uniform block lengths with 50:50 splits, the ideal offset in either direction (over two blocks) is 2L/S, so that the sum of the two desired offsets just happens to be:

$$\frac{2L}{S} + \frac{2L}{S} = \frac{4L}{S} = C$$

$$\frac{C}{4} = \frac{L}{S}$$

The travel time of each platoon along two consecutive blocks is exactly one-half of a cycle length, so that two such travel times add up to the cycle length.

For the testing purpose of DBAT, following condition is created:

- Four signalized intersections on arterial,
- Each intersection spacing is 1,000 ft,
- Cycle length is 80 seconds,
- Effective green is 40 seconds for all intersections, and
- Progression speed is 50 fps.

The ideal offset of the double alternate system is 0, 0, 40 and 40 seconds from first to last intersection, respectively. In addition, both directional bandwidth sizes are 20 seconds. Table 4.14 shows DBAT's test result. The DBAT results shows that the outbound direction made 3 bands and the inbound direction made 2 bands. The average bandwidth size is 20 seconds for both directions. DBAT provides correct bandwidths under the same offset.

Table 4.14 DBAT test results of double alternate progression

ID	Int.1	Int.2	Int.3	Int.4	Band1	Band2	# of Band1	# of Band2	Ave Band1	Ave Band2	Sum of band1, band2
1	0	0	40	40	60	40	3	2	20	20	100

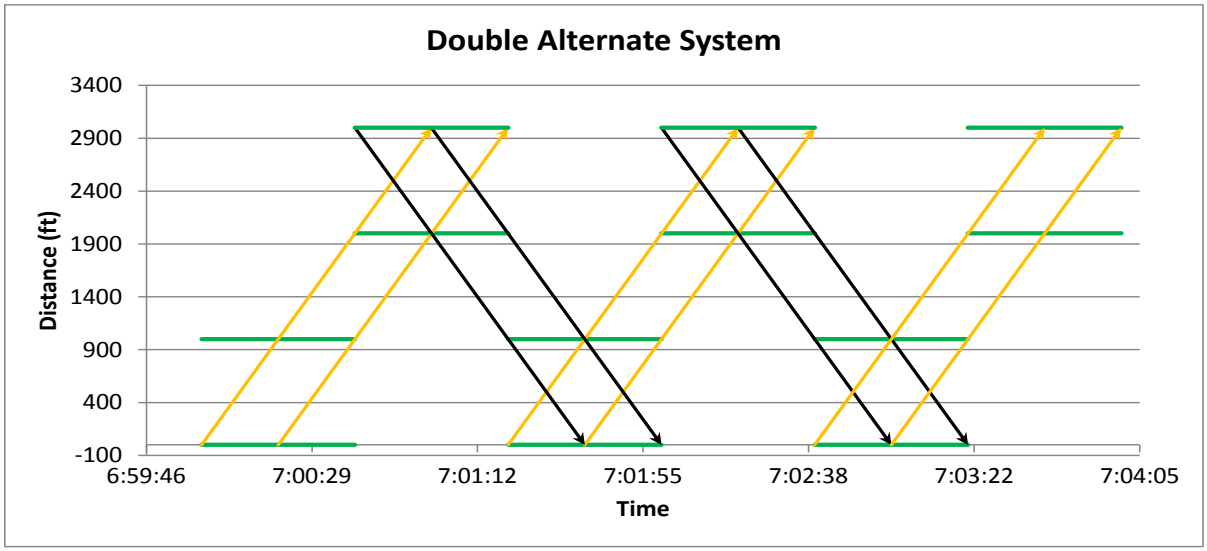


Figure 4-27 DBAT Test Result of Double Alternate System Case

The result of cycle-by-cycle bandwidths is also 20 seconds for both directions like Figure 4-27.

4.4.3 Simultaneous System

For very closely space signals, or for rather high vehicle speeds, simultaneous systems might be one of the best operation strategies. For the testing purpose of DBAT, following condition is created:

- Four signalized intersections on arterial,

- Each intersection spacing is 400 ft,
- Cycle length is 80 seconds,
- Effective green is 40 seconds for all intersections,
- Progression speed is 40 fps, and
- Each intersection offset is 0 for all intersection.

Under the given scenario, outbound and inbound bandwidth should be 10seconds, and the DBAT calculation result was confirmed to have the exact same result like Figure 4-28 and.

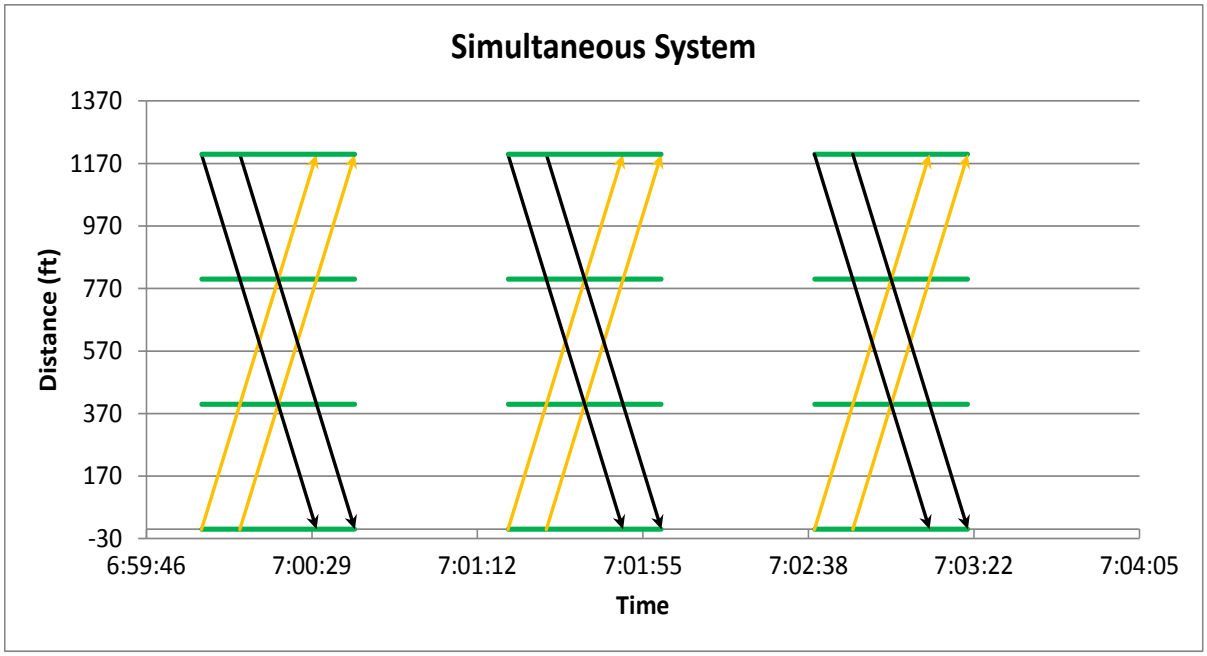


Figure 4-28 DBAT Test Result of Simultaneous System Case

Table 4.15 DBAT test results of double alternate progression

ID	Int.1	Int.2	Int.3	Int.4	Band1	Band2	# of Band1	# of Band2	Ave Band1	Ave Band2	Sum of band1, band2
1	0	0	0	0	30	30	3	3	10	10	60

CHAPTER 5 DYNAMIC BANDWIDTH OPTIMIZATION

The early return to green and green extension bring up the concept of using dynamic greens extracted from OASIS split monitor log for offset optimization. In particular, optimizing the offsets based on the dynamic green times to maximize dynamic bandwidth has great potential to yield even more efficient arterial performance.

The primary objective of this Chapter is to develop dynamic bandwidth optimization methods for improving arterial performance by optimizing offsets, for a set of observed cycles. For this purpose, author introduces exhaustive search method and a new linear programming formulation.

An exhaustive search method is used that enumerates all possible offset combinations and determines bandwidth efficiency for each combination using DBAT. Next, the solutions are sorted based on a performance measure of interest (e.g., bandwidth efficiency or weighted sum of bi-directional bandwidth). From this sorted list, the best or a certain target value of top solutions can be selected. The proposed exhaustive search approach can be used to evaluate the possibility of further improvements in dynamic bandwidth efficiency and to observe the search space of the problem.

Proposing a new LP formulation is developed due to exhaustive search method's calculation time. Exhaustive search method calculation time increases to the point of becoming impractical as the number of intersections increases but provided LP formulation does not this limitation.

5.1 Exhaustive Search Method (DBAT)

The proposed exhaustive search method is very general technique to solve or explore optimal solution and solution space. The exhaustive search method consists of systematically enumerating all possible solutions and checking solutions under given problem statement (constraints).

In order to perform an exhaustive analysis of dynamic bandwidth, the DBAT algorithm described above was embedded in a routine that enumerates all possible offset combinations. As with the DBAT bandwidth analysis algorithm, the minimum offset search interval was set to 1 second. The search space of the offset optimization problem is a function of the cycle length and the number of intersections as follows:

$$\text{Number of feasible solutions} = C^{|I|-1}$$

Where, C = cycle length

$|I|$ = number of intersections

The DBAT exhaustive search routine considers directional demand using a target value "*k*" as defined in the MAXBAND formulation. "*k*" represents the desired ratio of the sum of inbound to the sum of outbound dynamic bandwidth and should be set based on the ratio of the corresponding directional traffic demand.

The exhaustive search method ensures that the solution is globally optimal and provides important information on the shape of the solution space and the robustness of

different solutions. However, calculation time increases to the point of becoming impractical as the number of intersections increases. If spanning of the search space is desired for arterials with higher numbers of intersections, it is possible to reduce the number of solutions by increasing the search time increment. Figure 5-1 shows the example of exhaustive search result.

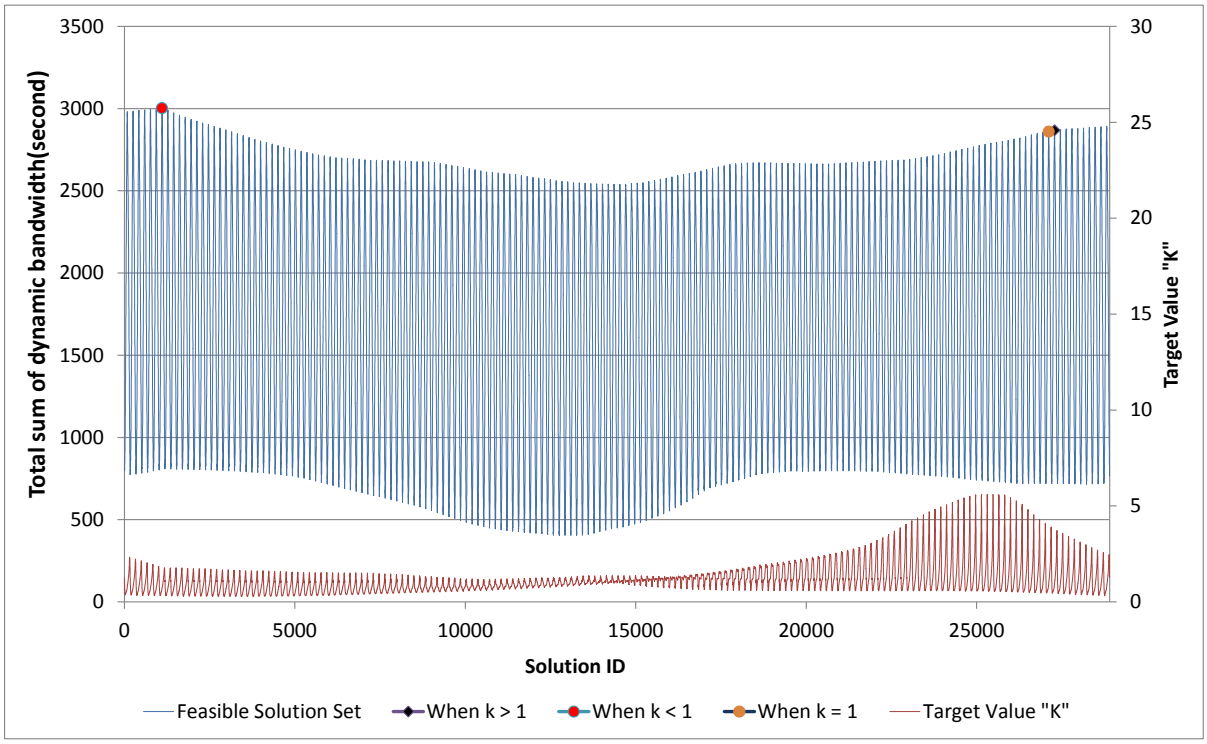


Figure 5-1 Example of Exhaustive Search Result

5.2 Linear Programming

This section presents a linear programming (LP) formulation to enable dynamic bandwidth maximization on semi-actuated arterial streets. The methodology relies on the use of archived signal log data regarding phase start and end times in each cycle in both directions. Those data are used to optimize the offsets that maximize the variable system bandwidth across multiple cycles constituting a coordination plan period. The formulation also is flexible to be used to optimize signal offsets using programmed (fixed) green durations at each intersection. The proposed formulation offers five significant enhancements compared to traditional methods.

1. The formulation is strictly linear (complexity of class P) as opposed to the traditional mixed integer programming formulations (complexity of class NP-hard).
2. It can work with either static or dynamic (cycle varying) green durations.
3. Traditional bandwidth optimization methods have explicit constraints to enforce bandwidth allocation to be proportional to directional demand. However, as longs as the minimum required bandwidth is allocated to each direction, a proportional allocation of bandwidth is not necessary and in fact, can result in reporting sub-optimal solutions. The proposed formulation overcomes this shortcoming.
4. The proposed formulation predicts the maximum proportion of traffic demand that can be served in the bandwidth, a unique attribute absent from other formulations found in the literature.

As an added enhancement, the formulation enables the analyst to carry out a post-processing step which reports the range of offsets at non-critical intersections within which the optimal objective function value will be unaffected.

5.2.1 New LP Formulation

The concept of the proposed bandwidth optimization LP is that green band is continuously created per each cycles, so it is possible to directly constrain each directional band start and end points inside of green durations. After defining green start and end time at each intersection, LP can search for optimal offsets, which maximize the sum of directional bandwidth. The main constraints are each green start and end time at each cycle. The numbers of input cycles are decided by progression speed and arterial length. Figure 5-2 shows basic geometry of the new bandwidth optimization model.

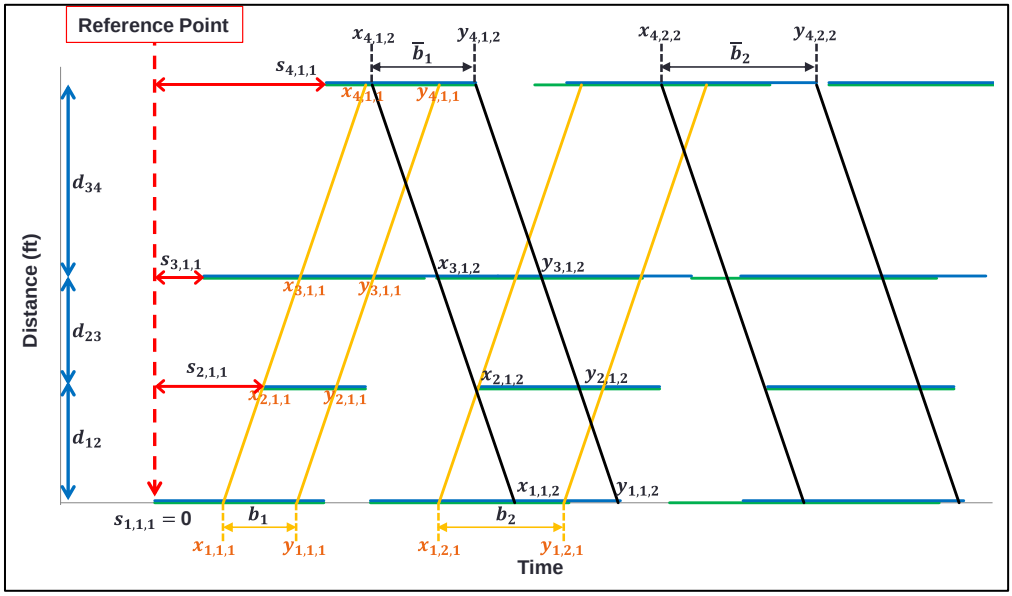


Figure 5-2 Dynamic Bandwidth Optimization Basic Geometry

Variables are defined as follows:

$b_j = j^{th}$ cycle outbound bandwidth;

$\bar{b}_j = j^{th}$ cycle inbound bandwidth;

$g_{i,j,k}$ = Green phase duration for intersection i, in cycle j, direction k;

$i = 1, 2, 3, \cdots, l$ $\quad$ l = number of intersection

$j = 1, 2, 3, \cdots, m$ $\quad$ m = number of cycles

$k = 1, 2$ $\quad$ 1 = outbound direction, 2 = inbound direction

$s_{i,j,k} =$ green start time for intersection i, in cycle j, direction k;

$e_{i,j,k} =$ green end time for intersection i, in cycle j, direction k;

$d_{i,z}$ = distance between intersection i and upstream (ft);

$v_{i,z} =$ progression speed between intersection i and upstream (ft/sec);

$L_{i,z} =$ Lower bound of speed between intersection i and upstream (ft/sec);

$H_{i,z} =$ upper bound of speed between intersection i and upstream (ft/sec);

$t_{i,z}$ = travel time between intersection i and upstream ($\frac{d_{i,z}}{v_{i,z}}$ sec);

$x_{i,j,k} =$ band start time for intersection i, in cycle j, direction k;

$y_{i,j,k} =$ band end time for intersection i, in cycle j, direction k;

$O_i =$ offset for intersection i (sec);

$D =$ outbound average demand(vehicle) per cycle (sec);

$\bar{D} =$ inbound average demand(vehicle) per cycle (sec);

$k =$ inbound demand over outbound demand ($\frac{\bar{D}}{D}$);

$\tau_i(\bar{\tau}_i) =$ oubound (inbound)queue clearance time for intersection i;

$\alpha =$ minimum fraction of the directional demand;

$M =$ very large number;

$S =$ very small number;

Let b_j and $\bar{b}_j$ represent the j^{th} cycle bandwidth in the outbound and inbound directions, respectively, where $j \in J$. The objective function can be written as follows:

$$\max \sum_{\forall j \in J} \left(b_j + k\bar{b}_j\right) + \alpha \cdot M \tag{5.1}$$

The objective function includes a target value (k) to maximize the sum of two-way bandwidths, while favoring the direction with the heavier traffic demand level. The objective function along with constraints (5.2) and (5.3) below insure that the formulation does not constrain the outbound and inbound bandwidth durations to be identical to the directional demand level. The formulation determines what fraction of demand α ($0< \alpha \leq 1$) can be processed in the band, and tries to maximize this value by using a large multiplier (M) in the objective function. When the time required to process the demand for through movement is below the maximum possible bandwidth, α will be 1, otherwise, ($0< \alpha <1$). Let D and $\bar{D}$ respectively denote the average number of seconds per cycle required to process the demand level in the outbound and inbound directions. Using constraints (5.2) and (5.3) guarantees

that in each direction, the sum of the provided bandwidths is greater than the total seconds required to process the highest possible percentage of the demand level.

$$\sum_{\forall j \in J} b_j \geq \alpha \cdot D. |J| \tag{5.2}$$

$$\sum_{\forall j \in J} \overline{b_j} \geq \alpha \cdot \overline{D}. |J| \tag{5.3}$$

The outbound and inbound band sizes are computed by subtracting the start time of the corresponding band from its end time as follows (see Figure 5-2):

$$b_j = y_{1,j,1} - x_{1,j,1}, \quad \forall j \in J \tag{5.4}$$

$$\bar{b}_j = y_{1,j,2} - x_{1,j,2,} \quad \forall j \in J \tag{5.5}$$

Let $x_{i,j,k}$ and $y_{i,j,k}$ respectively represent the start and end time of band cycle $j \in J$ at intersection $i \in I$ for direction $k \in K$. Constraints (5.6) ensure that the bandwidth end time occurs later than its start time.

$$y_{i,j,k} \geq x_{i,j,k}, \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.6}$$

In addition, define $s_{i,j,k}$ and $e_{i,j,k}$ as the start and end times of the green duration at intersection $i \in I$ in cycle $j \in J$ for direction $k \in K$. The start time of the green $s_{i,1,k}$ of the first cycle ($j = 1$) at intersection $i \in I$ for the outbound directions ($k = 1$) is equivalent to its

offset (O_i) which is *decision variables* of proposing LP model, and as a result has to be between 0 and the common cycle (C) as shown below:

$$-C \leq s_{i,1,1} \leq C, \qquad \forall i \in I/\{1\}, \tag{5.7}$$

The first intersection in the outbound direction is used as the reference intersection, and as a result, the start time of the first green at this intersection is set to zero ($s_{1,1,1} = 0$).

Constraints (5.8) ensure that at an intersection $i \in I$, the start time of band $j \in J$ occurs after the start time ($s_{i,j,k}$) and before the end time ($e_{i,j,k}$) of the green cycle in direction $k \in K$, as indicated in Figure 5-2.

$$s_{i,j,k} \leq x_{i,j,k} \leq e_{i,j,k}, \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.8}$$

The travel time between intersections $i \in I$ and $z \in Z_{ik}$ is defined by $T_{i,z}$. The start time of band $j \in J$ at intersection $i \in I$ is the sum of the travel time from the first intersection to intersection $i \in I$ and the start time of band at the first intersection $x_{1,j,k}$ in each direction, as shown in Equation (5.9):

$$x_{i,j,k} = x_{1,j,k} + \sum_{\forall z \in Z_{ik}} T_{i,z} \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.9}$$

The same concept is used to limit the end time of band $j \in J$ at intersection $i \in I$ for direction $k \in K$:

$$s_{i,j,k} \leq y_{i,j,k} \leq e_{i,j,k}, \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.10}$$

$$y_{i,j,k} = y_{1,j,k} + \sum_{\forall z \in Z_{ik}} T_{i,z} \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.11}$$

Next, let d_{iz} denote the distance between intersections $i \in I$ and $z \in Z_{ik}$. The travel time between these intersections is simply found by dividing the distance between the two intersections (d_{iz}) by the progression speed (v_{iz}) as shown by equation (5.12):

$$T_{iz} = \frac{d_{iz}}{v_{iz}} \qquad \forall i \in I, \forall z \in Z_{ik} \tag{5.12}$$

Since link progression speed is a decision variable in the problem, it is bounded by upper (H_{iz}) and lower bounds (L_{iz}) as follows:

$$L_{iz} \leq v_{iz} \leq H_{iz} \qquad \forall i \in I, \forall z \in Z_{ik} \tag{5.13}$$

Finally, to account for the initial queues accumulating at each intersection, the initiation of the green duration is shifted by the time needed to clear the initial queue in the outbound and inbound directions ($\tau_{i,k}$):

$$s_{i,j,k} \leq x_{i,j,k} - \tau_{i,k} \qquad \forall i \in I, \forall j \in J, \forall k \in K \tag{5.14}$$

For an arterial system with at least one non-critical intersection, multiple solutions to the bandwidth optimization problem will exist. That is essentially the amount of time each

offset value can move forward or backward in time without impacting the optimal bandwidth solution. The objective function for the slack analysis process is:

$$\max \sum_{\forall j \in J} \left(b_j + k\bar{b_j}\right) + \alpha \cdot M \pm S \cdot \sum_{\forall i \in I/\{1\}} s_{i,1,1} \quad (5.15)$$

5.2.2 LP Model Test 1 (Optimal Solution for Static Green)

In this section, proposed LP formulation is tested on hypothetical arterial and on the Euclid Avenue arterial in Cleveland, Ohio, a site that was cited in Little's paper (Little, J. D. C., 1966). The application to hypothetical scenarios illustrates MAXBAND over-constrained formulation, which occurs when the maximum possible inbound and outbound bandwidth sizes are different. It shows how the new LP formulation overcomes this problem. The proposed LP model when applied to Euclid Avenue is shown to provide an identical bandwidth solution to MAXBAND.

When the maximum possible inbound and outbound bandwidth size are different, the constraints (5.1) to (5.3) will lead to an over constrained model issue. For example, when outbound maximum bandwidth is 30 seconds and the inbound maximum bandwidth is 20 seconds, Figure 5-3 depicts one of the optimal solutions from MAXBAND formulation. Since maximum outbound bandwidth is larger than maximum inbound bandwidth, when k is larger than 1, MAXBAND provides always global optimum solution.

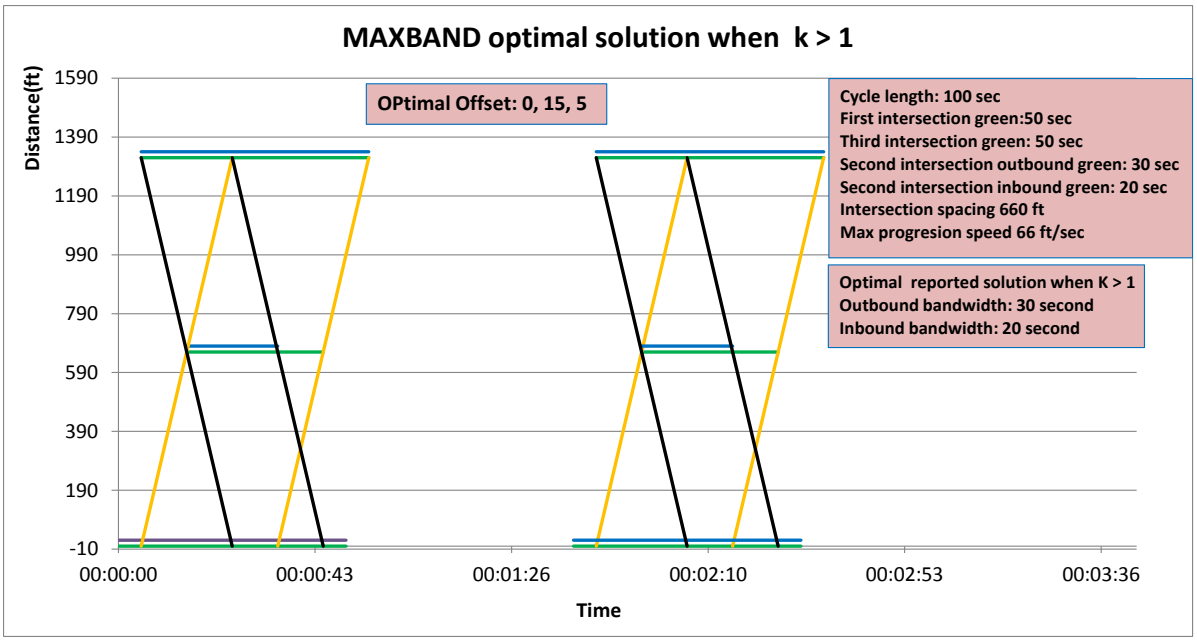


Figure 5-3 MAXBAND Optimal Solution (k>1)

However, it will provide artificially decreased outbound bandwidth when k is between 2/3 and 1. Figure 5-4 and Figure 5-5 shows example. Although outbound bandwidth can be 30 second, MAXBAND adjust offset and provide less bandwidth because of the poorly designed constraint.

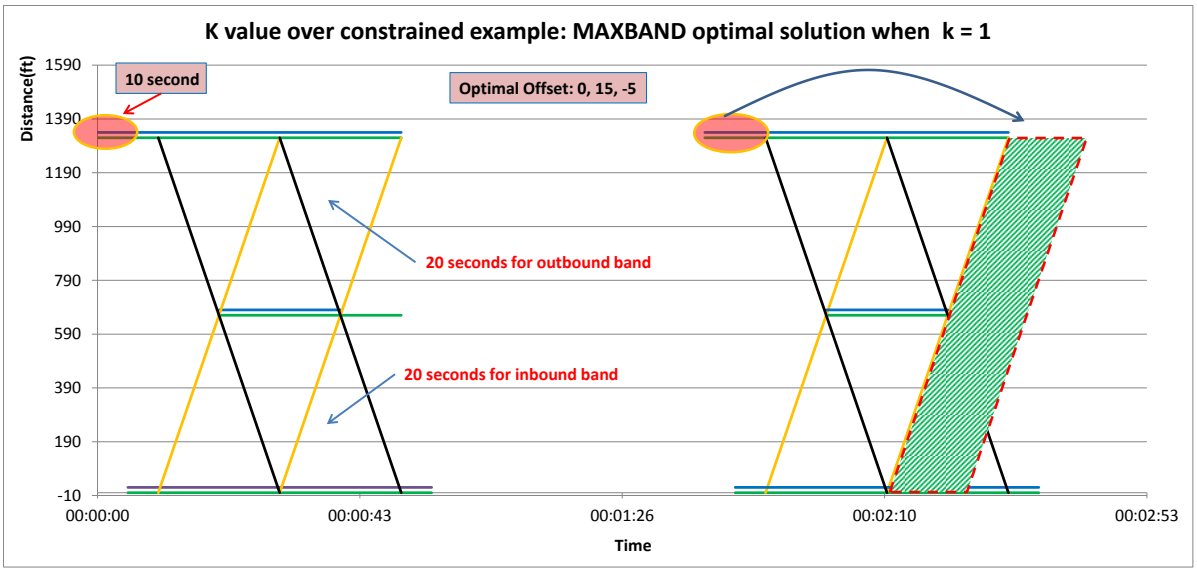


Figure 5-4 MAXBAND Suboptimal Solution (k=1)

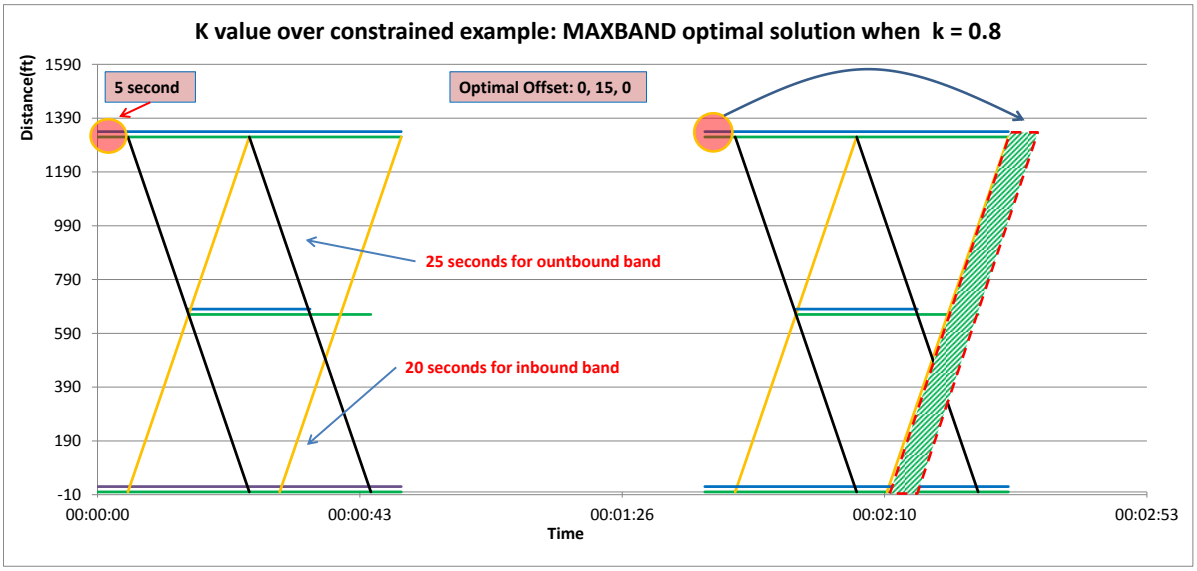


Figure 5-5 MAXBAND Suboptimal Solution (k=0.8)

Two hypothetical arterial cases are tested for model evaluation. First test case is Figure 5-6 and Figure 5-7. This example can show whether proposed LP can overcome MAXBAND over constraint problem. In case of k =1 with directional demand considered equal, LP provides 30 seconds for outbound band and 20 seconds for inbound bandwidth and calculated optimal α is 0.6667. In Figure 5-7 case, LP provides exactly same solution with Figure 5-6 except α value. The calculated optimal α is 0.625. In both cases, the proposed LP dese not have over constrain problem due to target value k and it gives same offset solution.

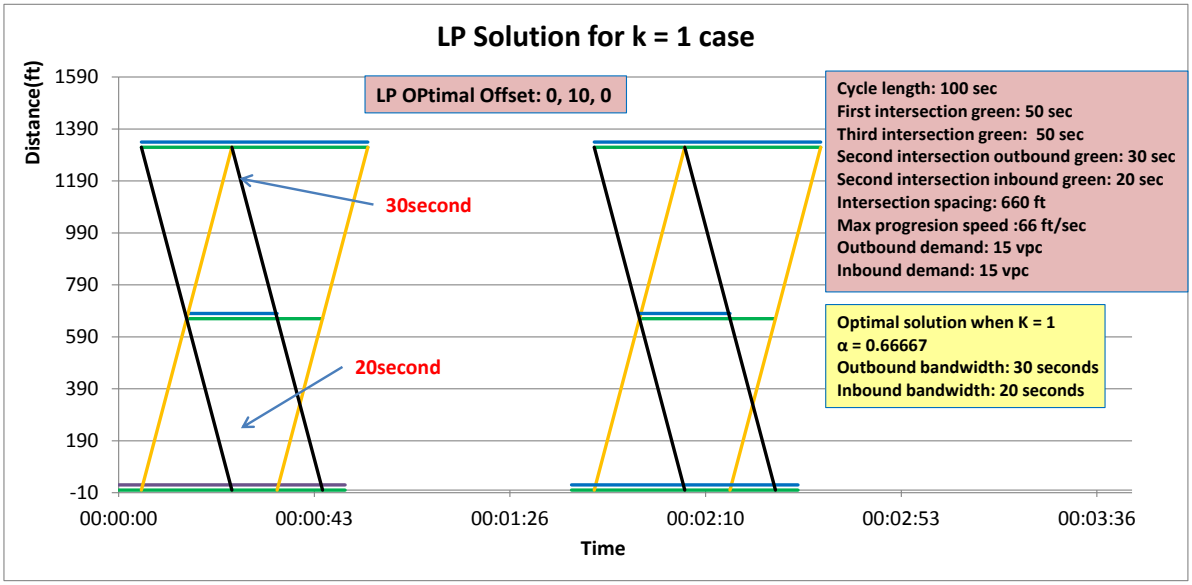


Figure 5-6 Optimal solution from Propose LP (k=1)

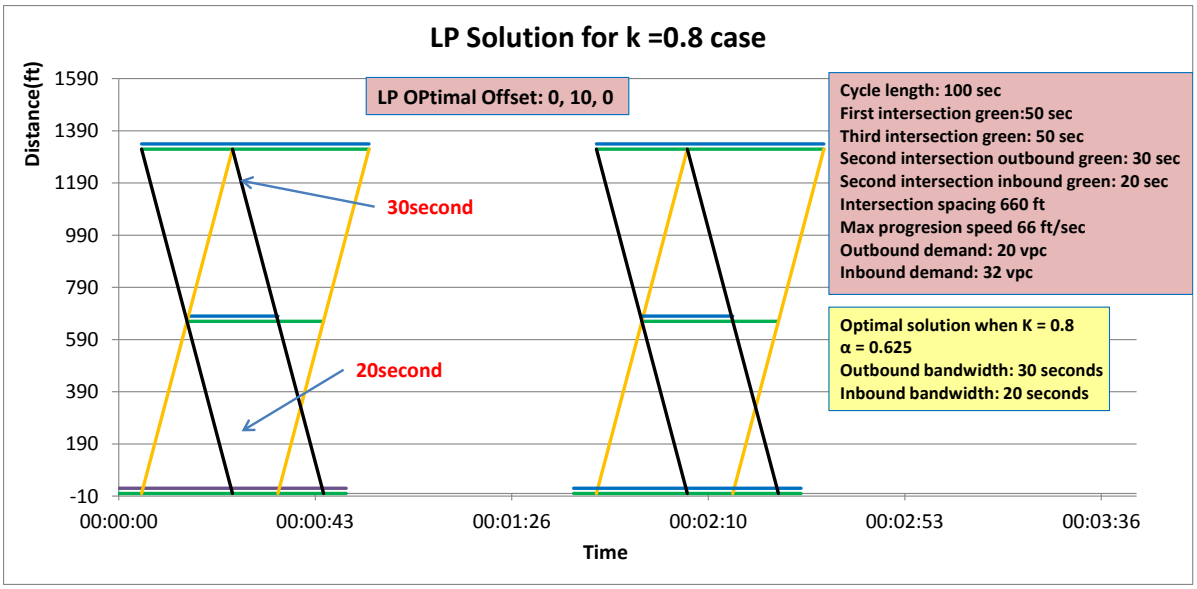


Figure 5-7 Optimal solution from Propose LP (k=0.8)

Second selected test example is Euclid Avenue in Cleveland which was used as example in Little's mixed integer linear programming. The selected example network is exactly same as chapter 5 However, the common cycle length is not 65 seconds but 75 seconds. Therefore, the corresponding green and red values are changed (See Table 3.5 Euclid Avenue Arterial in Cleveland Information). In keeping with Little's example, directional demand is assumed equal ($k = 1$). Little's mixed linear programming optimal solution in his paper provides 21.2 seconds bandwidth for each direction and the proposed LP's optimal solution provides 20.9 seconds. The 0.3 seconds difference comes from unit conversion or rounding. The calculated offsets are 0, -14.6, -2.8, -40.9, -37.2, -2.8, -11.7, -2.6, -40.4 and -46.5. Figure 5-8 shows proposed LP optimal solution result.

Table 5.1 shows detail arterial information. In keeping with Little's example, directional demand is assumed equal ($k = 1$). Little's mixed linear programming optimal solution in his paper provides 21.2 seconds bandwidth for each direction and the proposed LP's optimal solution provides 20.9 seconds. The 0.3 seconds difference comes from unit conversion or rounding. The calculated offsets are 0, -14.6, -2.8, -40.9, -37.2, -2.8, -11.7, -2.6, -40.4 and -46.5. Figure 5-8 shows proposed LP optimal solution result.

Table 5.1 Euclid Avenue Arterial in Cleveland Second Example for LP

Intersection	Distance (ft)	Red (Sec)	Green (sec)
1	0	35.25	39.75
2	551.18	30	45
3	1250.00	30	45
4	2349.08	35.25	39.75
5	3047.90	36	39
6	3848.42	31.5	43.5
7	4498.03	30	45
8	4898.29	30	45
9	5597.11	30	45
10	6046.58	31.5	43.5

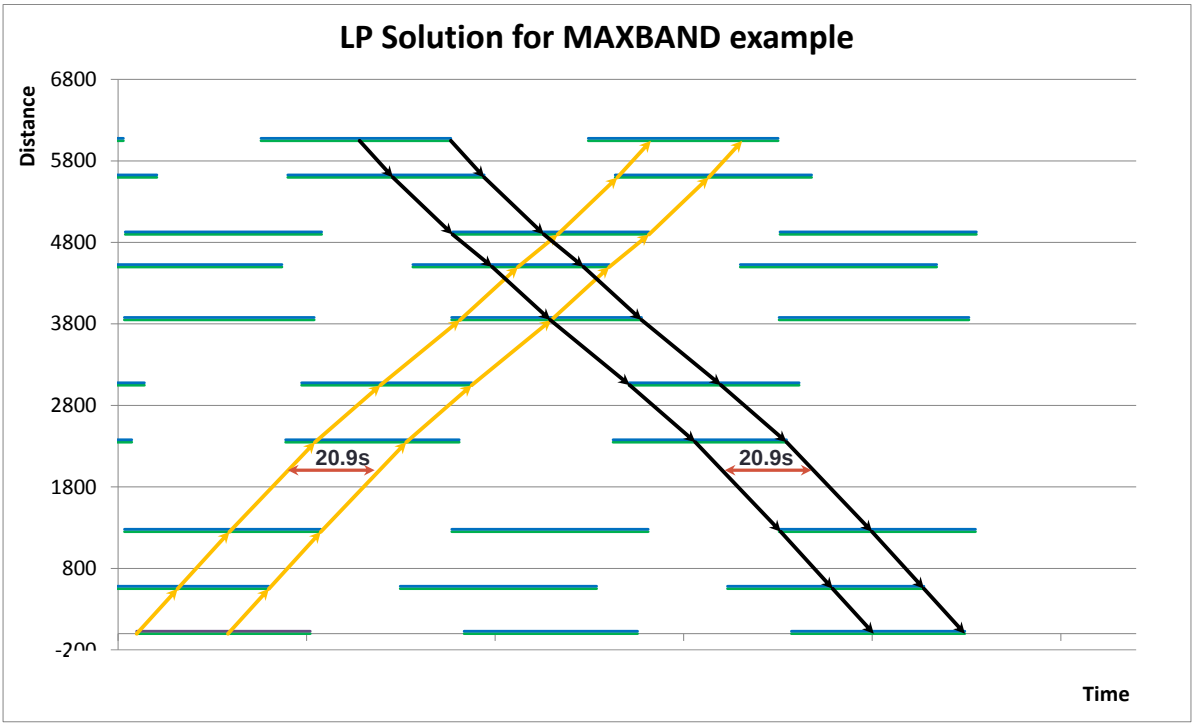


Figure 5-8 LP Test Example 2

5.2.3 LP Model Test 2 (Slack Analysis)

Figure 5-8 example 2 shows that some of intersections have slack. Slack is defined as the time that the start time of green can be moved forward or backward without changing the optimal solution. When at least one non-critical intersection exists, there should be multiple global solutions; however, existing bandwidth optimization approaches can find only one of them. These solutions may result in different arterial performance while yielding identical bandwidth. For example, when the up-stream intersection sends many right turn and left turn vehicle to the downstream intersection, the downstream intersection should start green signal

earlier. Therefore, slack information is important and can be used to improve arterial coordination.

By adjusting the objective function, slack analysis can be performed. The LP objective function is $\max \sum_{\forall j \in J}\left(b_j + k\bar{b_j}\right) + \alpha \cdot M$. The slack analysis LP object function is:

$$\max \sum_{\forall j \in J}\left(b_j + k\bar{b_j}\right) + \alpha \cdot M \pm S \cdot \sum_{\forall i \in \frac{I}{\{1\}}} s_{i,1,1}$$

In the proposed LP formulation, first intersection is always reference intersection so its offset is 0. Therefore, when positive sign is used in the new objective function, non-critical intersections' offset will have largest value. On the other hand, when negative sign is assigned, non-critical intersections' offset will have smallest offset value, however, critical intersection's offset will not be changed. Figure 5-9 and Figure 5-10 shows slack analysis results.

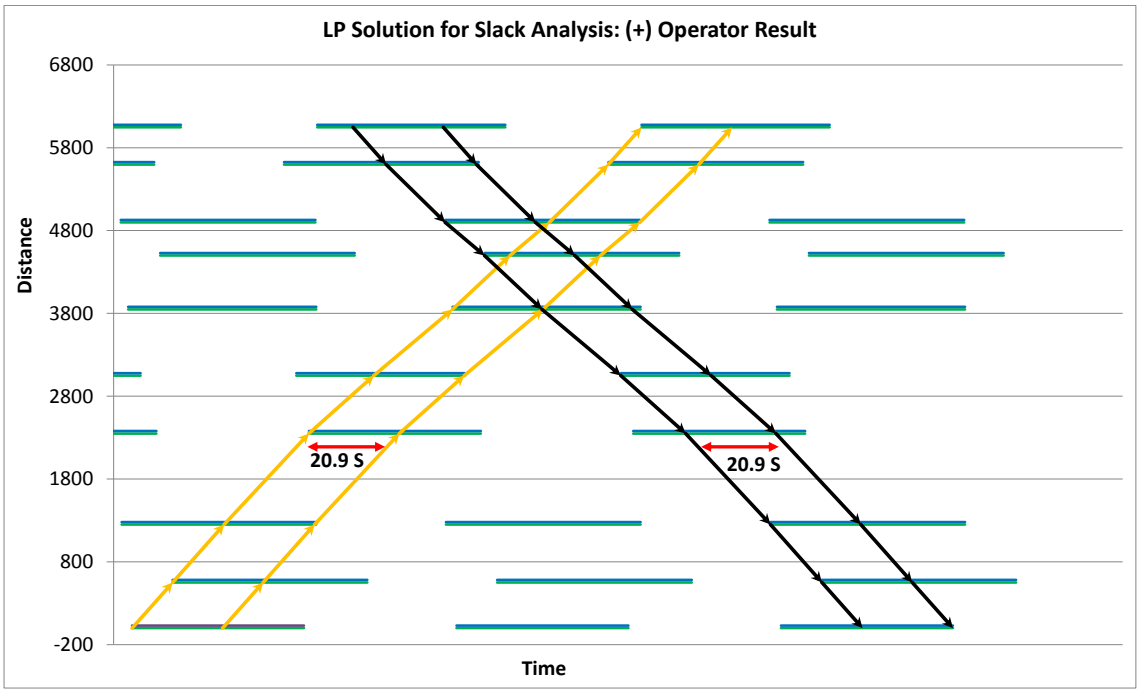


Figure 5-9 “+” Operator Slack Analysis Result

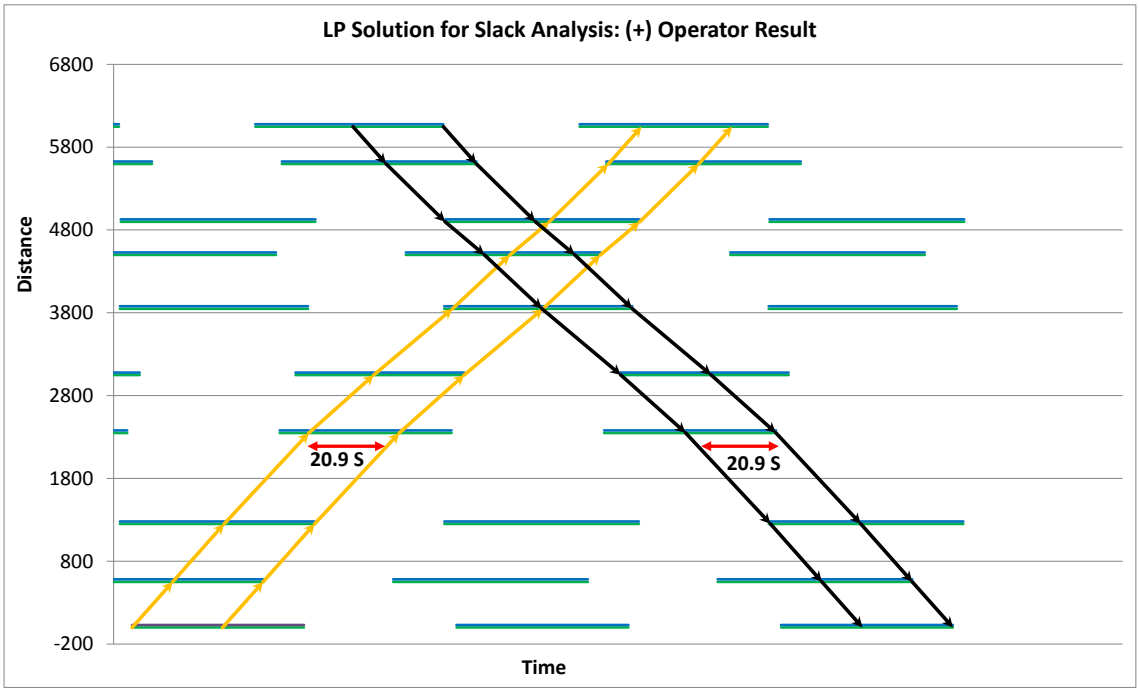


Figure 5-10 “-” Operator Slack Analysis Result

The provided bandwidth for both cases is still 20.9 seconds but, each solution's offsets are different. The (-) operator provide offset as 0, -14.6, -2.8, -40.9, -37.2, -2.8, -11.7, -2.6, -40.4 and -46.5 and (+) operator provide 0, 9.4, -2.8, -34.1, -37.2, -0.9, 6.5, -2.6, -40.4 and -32.2 as each intersection offset. Therefore, first, third, fifth, eighth and 9th intersection are critical intersection and the other four intersections have a slack. Table 5.2 shows all the offset solutions which have 20.9 seconds for each directional bandwidth.

Table 5.2 LP Solution for Slack Analysis Result

Intersection Offset										
	Int. 1	Int. 2	Int. 3	Int. 4	Int. 5	Int. 6	Int. 7	Int. 8	Int. 9	Int. 10
Upper bound of offset	0.0	9.4	-2.8	-34.1	-37.2	-0.9	6.5	-2.6	-40.4	-32.2
Lower bound of offset	0.0	-14.6	-2.8	-40.9	-37.2	-2.8	-11.7	-2.6	-40.4	-46.5
Slack	0.0	24.0	0	6.8	0	1.9	18.2	0.0	0	14.3
Offset Rage	0	-14.6 to 9.4	-2.8	-40.9 to -34.1	-37.3	-2.8 to -0.9	-11.7 to 6.5	-2.6	-40.4	-46.5 to -32.2

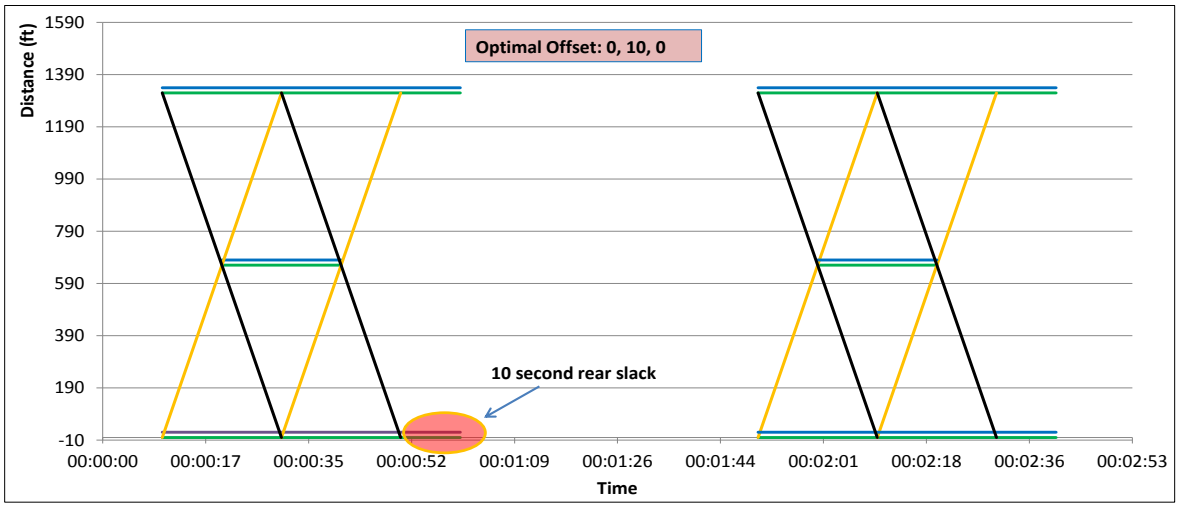
However, when first intersection has a slack, which means that first intersection is a non-critical intersection, one more step is required to verify slack analysis. First intersection's slack should be calculated by following equation.

First intersection front slack: $s_{1,j,1} - x_{1,j,1} > 0$ and $y_{1,j,2} - e_{1,j,2} = 0$

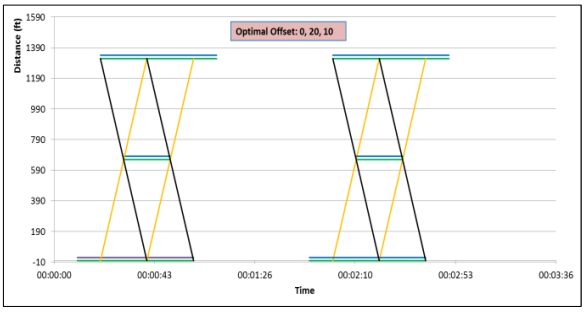
First intersection rear slack: $s_{1,j,1} - x_{1,j,1} = 0$ and $y_{1,j,2} - e_{1,j,2} > 0$

First intersection both side slack: $s_{1,j,1} - x_{1,j,1} > 0$ and $y_{1,j,2} - e_{1,j,2} > 0$

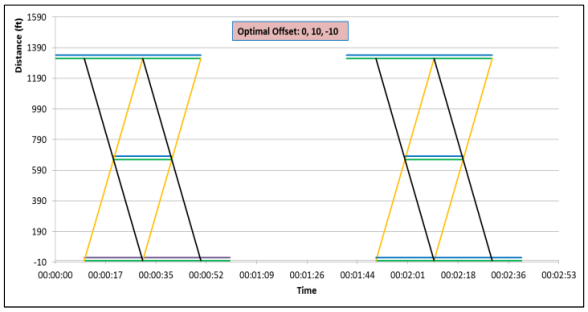
When first intersection has X seconds of front or rear slack, it is necessary to subtract X from all other intersection's calculated offsets. For example, Figure 5-11 (a) depicts the first intersection rear slack case.



(a) LP Optimal Solution



(b) (+) operator result



(c) (-) operator result

Figure 5-11 First Intersection Rear Slack Example

Table 3 shows that the adjusted each intersection real slack analysis result. Since the first intersection has 10 seconds rear slack, the second and third intersection looks has 10 and 20 seconds slack. However, the adjusted slack procedure fixed this error. In this case, the second intersection is the only critical intersection and the first and last intersection have a 10 seconds slack.

Table 5.3 First Intersection Rear Slack Case Analysis Result

	Intersection 1 (sec)	Intersection 2 (sec)	Intersection 3 (sec)
Upper bound	0	20	10
Lower bound	0	10	-10
Optimal Solution	0	10	0
Fist intersection rear Slack	10		
Slack	10	0	10
Offset Rage	-10 to 0	10	-10 to 0

5.2.4 LP Model Test 3 (Optimal Solution for Dynamic Green)

When the arterial has a semi-actuated signal controller, each intersection possibly has an early return to green and green extension. Therefore, green durations vary over cycles. The proposed linear program was applied to Site "A" (US 70 arterial in Clayton, NC). Table 5.4 shows dynamic green time information, which is downloaded from OASIS split monitor.

Table 5.4 Site “A” OASIS Split Monitor Log

US 70 & Shotwell Rd				
Time	ExtraTimeCP1	ExtraTimeCP2	Used Phase 2	Used Phase 6
06:32:11	18	34	79	49
06:35:01	8	25	79	79
06:37:51	24	37	79	49
06:40:41	24	40	79	49
06:43:31	16	37	79	49
US 70 & S Moor St				
Time	ExtraTimeCP1	ExtraTimeCP2	Used Phase 2	Used Phase 6
06:31:41	46	62	104	87
06:34:31	30	44	104	104
06:37:21	24	0	125	170
06:40:11	45	45	104	104
06:43:01	45	70	104	79
US 70 & R Robertson St				
Time	ExtraTimeCP1	ExtraTimeCP2	Used Phase 2	Used Phase 6
06:33:06	70	70	54	79
06:35:56	41	28	54	79
06:38:46	46	23	54	79
06:41:36	45	32	54	79
06:44:26	58	44	54	79

Both the DBAT exhaustive search method and the proposed LP method were applied to site A. LP provides optimal offset as of 0, 5.49, and 85.96 seconds. The calculated objective function, which is sum of both directional dynamic bandwidths, is 692.18 seconds. Figure 5-12 and Table 5.5 show proposed LP calculation results.

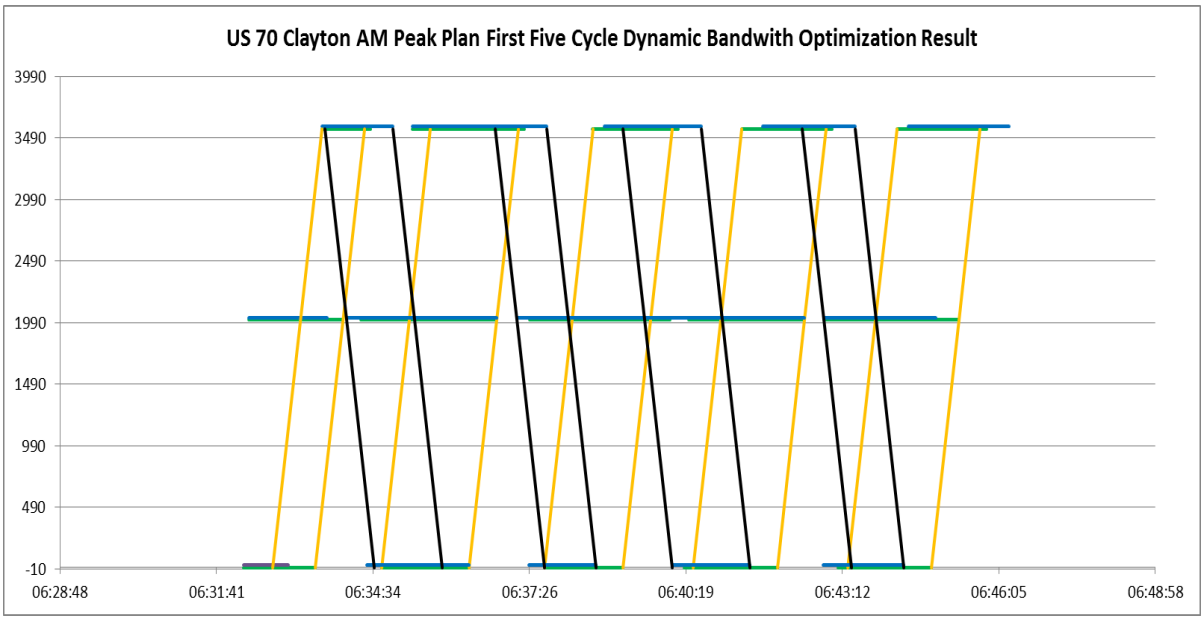


Figure 5-12 Dynamic Bandwidth LP Optimization Result

Table 5.5 Dynamic Green LP Optimization Result

	Band 1	Band 2	Band 3	Band 4	Band 5
Outbound Bandwidth	47.06	97.00	87.00	93.06	92.06
Inbound Bandwidth	-	75.00	57.00	86.00	58.00

The exhaustive search method provides 5 optimal offsets, and the sum of both directions dynamic bandwidths is 707 seconds. Figure 5-13 shows the DBAT exhaustive search result. Table 5.6 shows the summary of optimal offsets. The "Number of Inbound Bands" column indicates there are two secondary bands since linear programming only provided four inbound bands. In addition, the exhaustive search method provides a bigger

objective function value result than linear programming, since linear programming cannot consider the secondary bands.

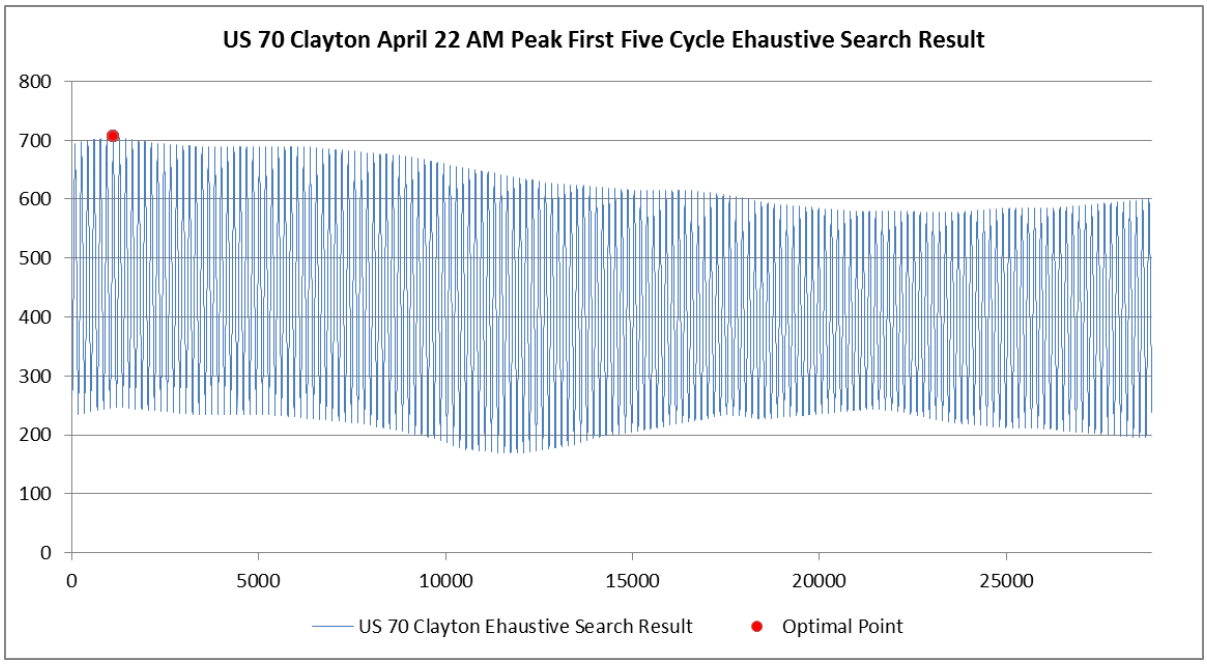


Figure 5-13 Dynamic Bandwidth DBAT Exhaustive Search Result

Table 5.6 DBAT Exhaustive Search Result Including Secondary Bands

ID	Int. 1	Int. 2	Int. 3	Outbound Bandwidth	Inbound Bandwidth	Number of Outbound Bands	Number of Inbound Bands	Sum of Directional Bandwidth
1101	0	6	80	434	273	5	6	707
1102	0	6	81	431	276	5	6	707
1103	0	6	82	428	279	5	6	707
1104	0	6	83	425	282	5	6	707
1105	0	6	84	422	285	5	6	707
1106	0	6	85	419	288	5	6	707

The secondary bands were eliminated using a secondary band filter in DBAT and then the exhaustive search result were summarized in Table 5.7.After eliminating secondary band, exhaustive search only provides one optimal solution. The calculated number of bands is identical to LP, but sum of bands is smaller. The LP objective function is larger than the exhaustive search since exhaustive search takes only integer value for its offsets and minimum search interval is 1 second.

Table 5.7 DBAT Exhaustive Search Result Excluding Secondary Bands

ID	Int. 1	Int. 2	Int. 3	Outbound Bandwidth	Inbound Bandwidth	Number of Outbound Bands	Number of Inbound Bands	Sum of Directional Bandwidth
1106	0	6	85	419	269	5	4	688
1105	0	6	84	422	265	5	4	687
1104	0	6	83	425	261	5	4	686
1103	0	6	82	428	257	5	4	685
1102	0	6	81	431	253	5	4	684
1101	0	6	80	434	249	5	4	683

After adding one constraint (enforcing offsets and start and end time of the bandwidth to take only integer values), the LP and exhaustive search methods give identical solutions. Therefore, the global solution for maximum sum of two way dynamic bandwidth is when each intersection offset is 0, 6, and 85 seconds. The total amount of the sum of bandwidths is

688 seconds. Figure 5-14 and Table 5.8 shows proposed LP solution with dynamic bandwidth. The optimized outbound and inbound dynamic bandwidth sum is 419 seconds and 269 seconds which are exactly same as Table 5.7 exhaustive search result.

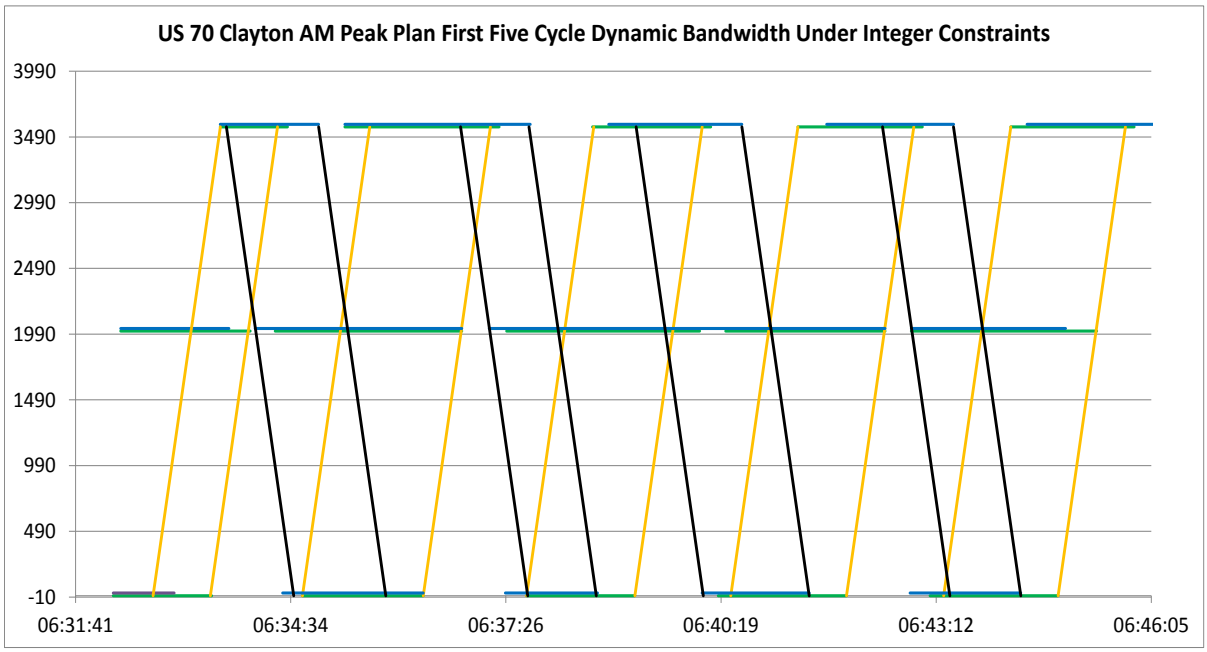


Figure 5-14 Dynamic Bandwidth LP Optimization Result under Integer Constraints

Table 5.8 Dynamic Green LP Solution under Integer Constraint

	Band 1	Band 2	Band 3	Band 4	Band 5	Sum
Outbound Bandwidth	47	97	87	95	93	419
Inbound Bandwidth	-	72	55	85	57	269

For more precise LP solution evaluation, two hour real field dynamic green data are used for both exhaustive search method and LP. Exhaustive search method provides one optimal solution under including secondary bands while LP provides three offset solutions which have identical sum of bidirectional bandwidth. In order to verifying LP solution, secondary bands in exhaustive search result are excluded. Figure 5-15 and Figure 5-16 show both methods solutions. As mention above, LP method cannot consider secondary bandwidth so there could be difference between LP and exhaustive search method provided optimal solution. Therefore, secondary bands are eliminated on exhaustive search result as well as integer constraints are added on proposed LP formulation for direct comparison.

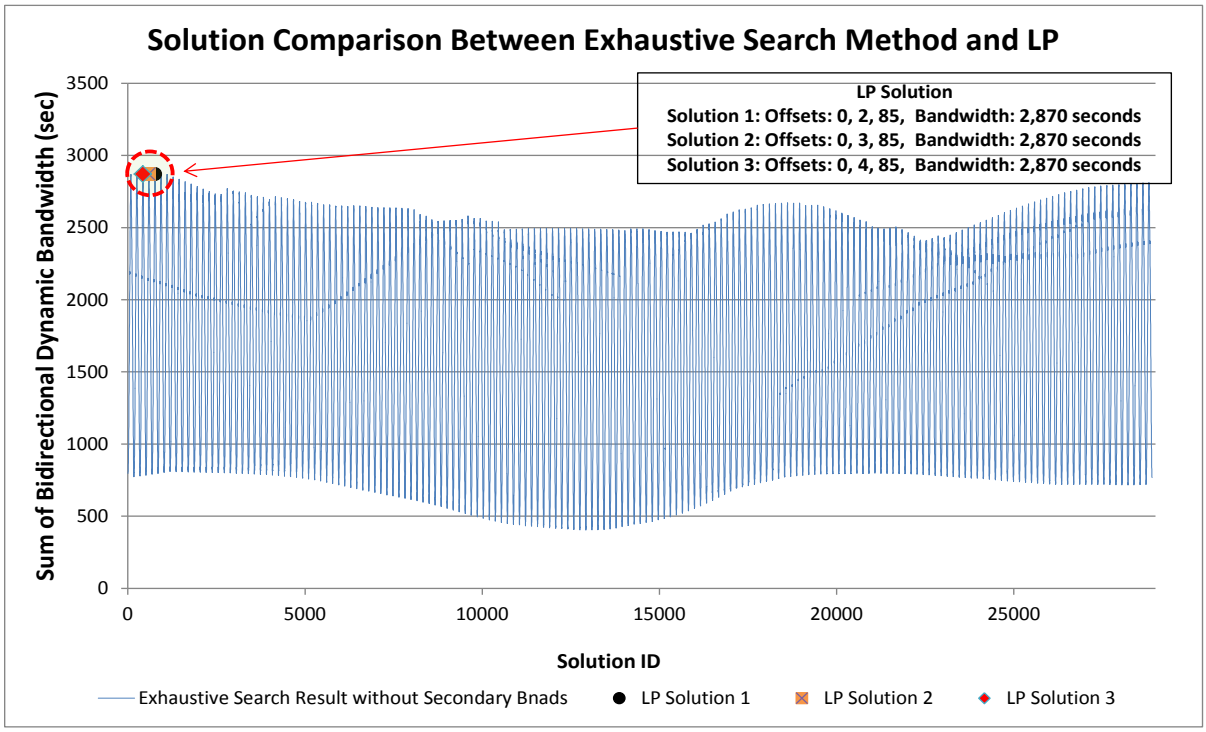


Figure 5-15 LP and Exhaustive Search Method Solution (without Secondary Bands for Exhaustive Search Result and Integer Constraints on LP)

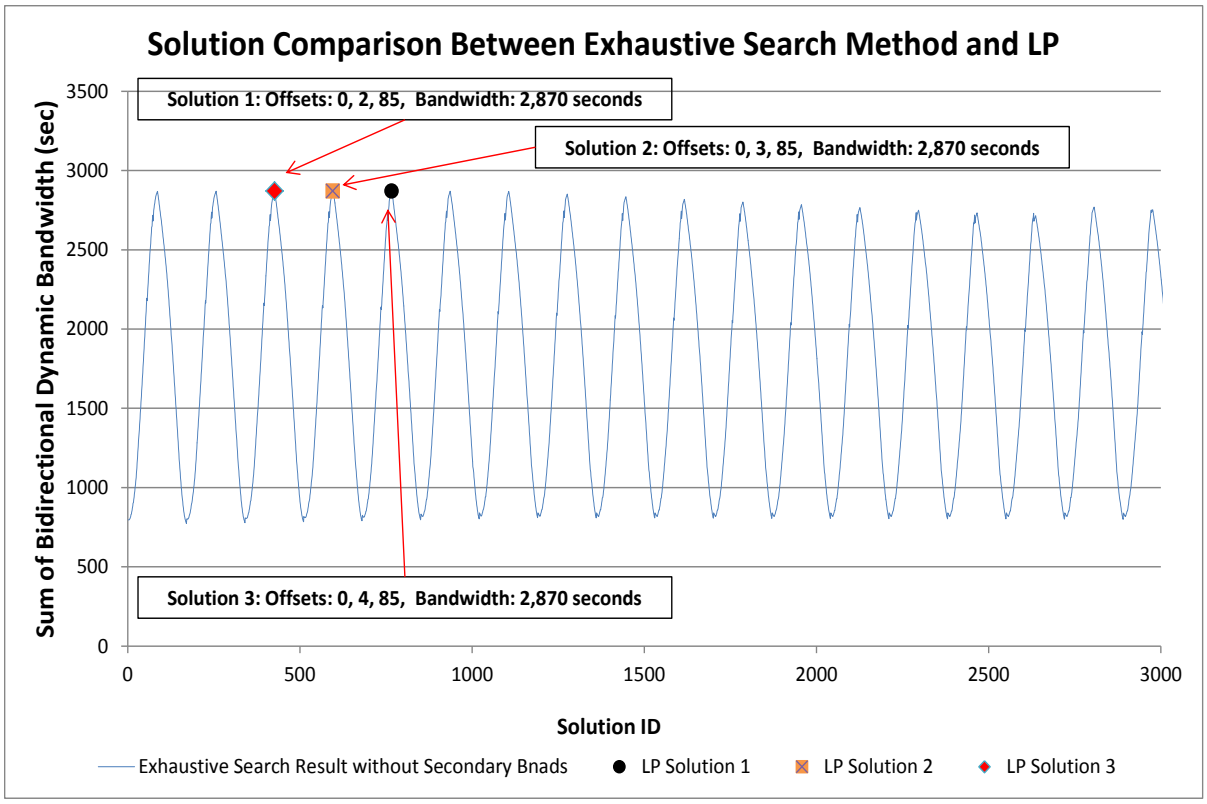


Figure 5-16 Exhaustive Search Result near Optimal Solution Area

In addition, LP provides 2,870 seconds as objective function value and three offsets have identical sum of bidirectional dynamic bandwidth (2,870 sec). However, those LP solutions have different sum of bidirectional bandwidth from exhaustive search result due to secondary bandwidth. Table 5.9 and Figure 5-17 show the difference between LP and exhaustive search result. Even though provided offset solution is different, the sum of directional bandwidth (objective function for both methods) value is very similar (minimum 6 seconds and maximum 14 seconds) to each other. In addition, provided offset solutions are not significantly different. The proposed LP model cannot consider secondary bands, but its computation time is significantly less than the exhaustive search method.

Table 5.9 LP solutions on Exhaustive Search Result (Including Secondary Bandwidth)

ID	Int. 1	Int. 2	Int. 3	Outbound Bandwidth	Inbound Bandwidth	Number of Outbound Bands	Number of Inbound Bands	Sum of Directional Bandwidth
766	0	2	85	1,695	1,301	21	23	2,996
596	0	3	85	1,678	1,314	21	23	2,992
426	0	4	85	1,661	1,327	21	22	2,988

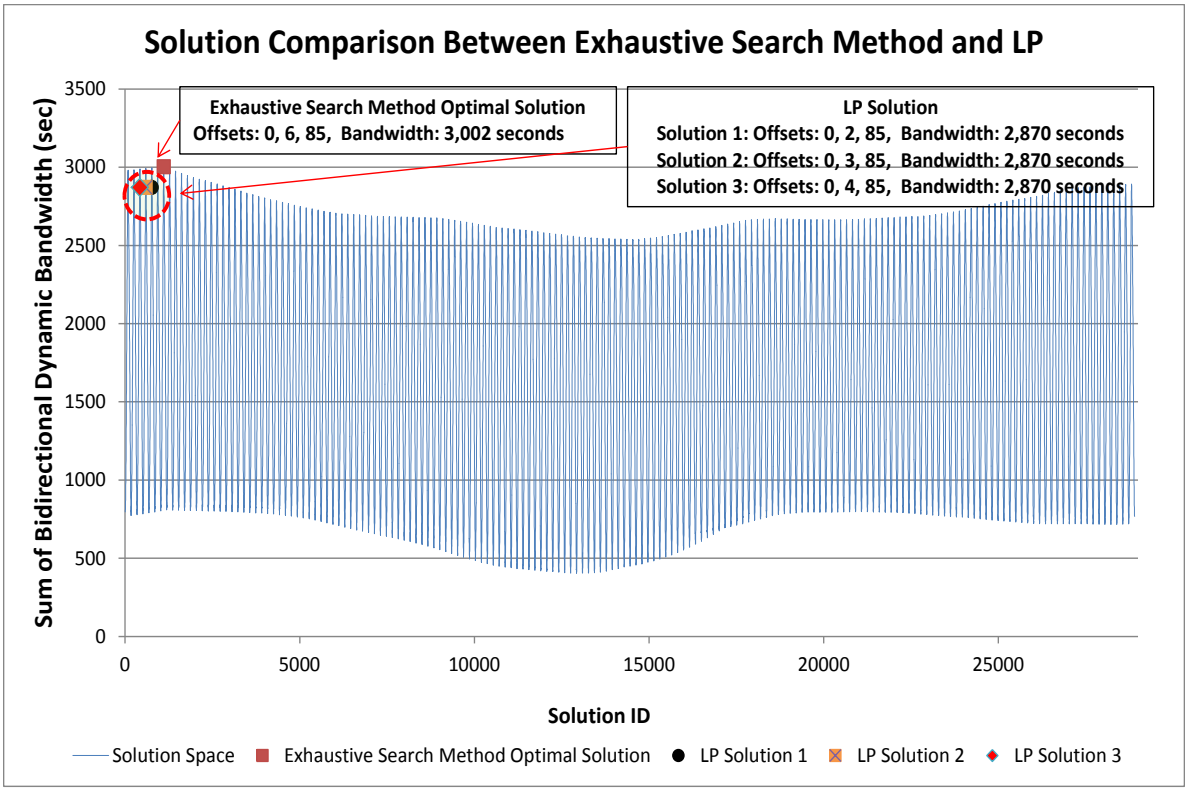


Figure 5-17 Exhaustive Search Result Including Secondary Bands

CHAPTER 6 FIELD DATA ANALYSIS

In this Chapter, real world arterial dynamic bandwidth analysis results from collected ATMS data are introduced in section 6.1. In addition, dynamic bandwidth optimization results and field evaluation are also described in section 6.2 and section 6.3.

Section 6.1 presents dynamic bandwidth analysis results. For all three study sites, one week OASIS split monitor data were processed by the developed Dynamic Bandwidth Analysis Tool (DBAT). Traditionally, calculated programmed green bandwidth provides limited insight for evaluating the quality of arterial coordination because it cannot assess the impact of early return to green and green extension for the coordinated phases. Detailed analysis at three study sites confirmed that real world coordinated green times along signalized arterials have complex multimodal distributions.

Section 6.2 present dynamic bandwidth optimization results. The provided results support that dynamic bandwidth optimization method can improve arterial coordination quality since it can provide better bandwidth solution by optimized offset.

Section 6.3 shows before and after filed test result. Total six time of day plans are evaluated. Five time of day plans' field offsets are very similar to optimized offset but midday offset was improved a lot and before and after field travel time study support the effectiveness of dynamic bandwidth optimization.

6.1 Dynamic Bandwidth Characteristics

In this section, dynamic bandwidth analysis results are introduced for the three study sites introduced in Chapter 3. The collected OASIS split monitor logs for dynamic bandwidth analysis are:

- Site A (US 70 arterial in Clayton): April 22 to 25, 2013 (four weekdays)
- Site B (US 70 arterial in Garner): February 20 to 24, 2012 (five weekdays)
- Site C (NC 55 arterial in Apex): August 27 to 31, 2012 (five weekdays).

6.1.1 Dynamic Bandwidth

The peak directions for site A, B and C are outbound, inbound and outbound, respectively. Table 6.1 summarizes the total duration, number, average duration, and overall efficiency of dynamic green bands for outbound and inbound directions for all three sites. These results are provided for different weekdays, and the total sum of bandwidth in both directions is also provided. The number of bands includes secondary and non-programmed bands. All bands shorter than the programmed band are defined as secondary bands as mentioned earlier in Chapter 4. For Site C, NC 55, the programmed signal timings provide no bandwidth in the inbound direction, and therefore the green bands provided are shaded to indicate they are non-programmed bands. The result shows daily fluctuation of bandwidth as well as the number of bands, which are calculated by DBAT.

Table 6.1 DBAT Arterial Dynamic Bandwidth Analysis Results

Site	Day	Out-bound Total Duration (Sec)	In-bound Total Duration (sec)	# of Out-bound Bands	# of In-bound Bands	Out-bound Average Duration (sec)	In-bound Average Duration (sec)	Out-bound Efficiency (%)	In-bound Efficiency (%)	Out- & In-bound Total Duration (sec)
A	1	1,661	1,315	21	23	79.1	57.17	46.5	38.7	2,976
	2	1,655	1,360	21	22	78.8	61.82	46.4	40.0	3,015
	3	1,673	1,157	21	22	79.7	52.59	46.9	34.0	2,830
	4	1664	1,302	21	20	79.2	65.1	46.6	38.3	2,966
B	1	6,698	7,220	139	138	48.2	52.3	40.5	43.7	13,918
	2	6,846	7,493	139	137	49.3	54.7	41.6	45.6	14,415
	3	6,634	7,186	137	138	48.4	52.1	40.4	43.7	13,896
	4	6,497	7,109	137	137	47.4	51.9	39.5	43.2	13,682
	5	6,532	7,016	137	137	47.7	51.2	39.7	42.7	13,624
C	1	3,152	196	40	36	78.8	5.4	49.1	3.1	3,419
	2	3,159	186	40	30	79.0	6.2	49.2	2.9	3,413
	3	3,207	172	40	33	80.2	5.2	49.9	2.7	3,447
	4	3,204	193	40	37	80.1	5.2	49.9	3.0	3,465
	5	3,135	126	40	28	78.4	4.5	48.8	2.0	3,329

Non-programmed bandwidths are shaded

Table 6.2 shows Site A's cycle-by-cycle bandwidth results for each analysis day. Outbound bandwidth does not have secondary bands but inbounds has several secondary bands. The gray color represents secondary bands. A total of seven secondary bands are

generated naturally four of which are less than 4 seconds and 3 band are longer than 9 seconds.

Table 6.2 Dynamic Bandwidth Computation Result of Site "A" from DBAT

Outbound					Inbound			
Day 1	Day 2	Day 3	Day 4		Day 1	Day 2	Day 3	Day 4
44	44	44	44		77	76	59	57
94	85	97	92		9	73	82	84
87	89	82	83		58	81	11	77
90	87	79	88		85	59	40	59
89	93	86	92		10	3	58	58
92	86	84	89		59	57	59	87
90	91	100	87		84	93	82	59
69	69	74	86		60	76	73	58
72	80	71	69		73	56	60	54
59	69	79	72		56	73	59	77
69	72	72	68		78	56	59	56
79	72	69	74		70	79	74	77
72	82	86	72		1	74	58	82
81	70	75	69		59	76	1	59
78	73	81	82		59	67	60	56
81	82	86	80		80	3	73	57
78	83	80	85		59	49	39	56
74	83	75	78		74	51	54	58
91	77	85	86		56	73	39	73
85	84	83	81		56	58	39	58
87	84	85	87		78	73	39	
					39	54	39	
					39			

Secondary bandwidths are shaded

Figure 6-1 shows the dynamic bandwidth plot of first five cycles on the first day on site A. It shows two secondary bands. The first secondary band is 9 seconds and the second one is 10 seconds.

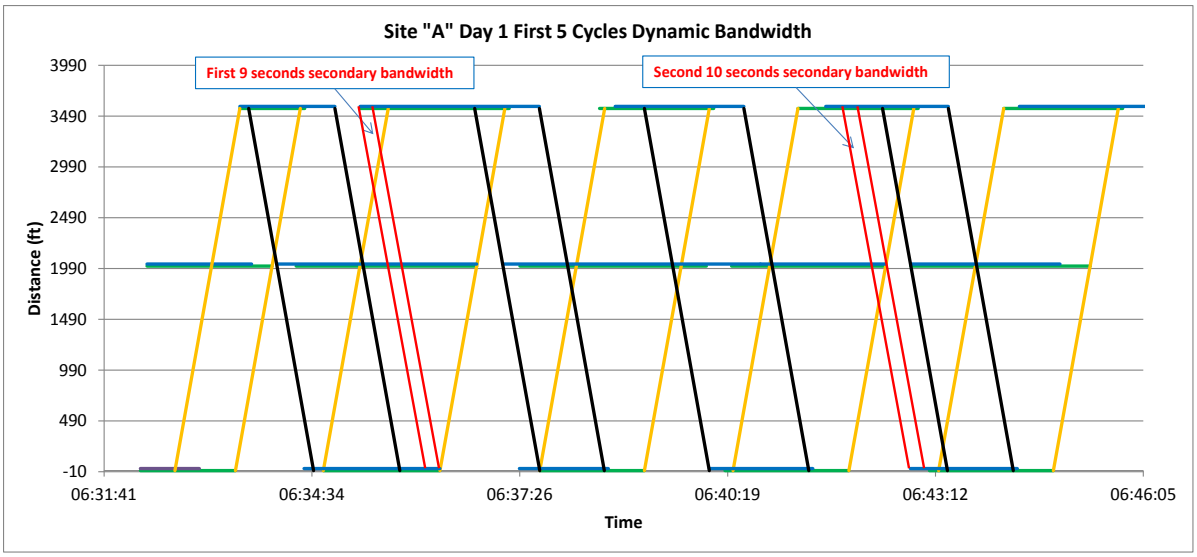


Figure 6-1 US 70 Arterial in Clayton Primary Band with Secondary Band

6.1.2 Observed Dynamic and Programmed Bandwidth Efficiency

Table 6.3 shows the programmed bandwidth and daily dynamic bandwidth efficiency for all three study-sites. As expected, the programmed bandwidth underestimated the actual bandwidth that was provided in all cases. For Site A, programmed bandwidth underestimates the actual directional bandwidths by 48% to 81%. For Sites B and C, programmed bandwidth underestimated the actual bandwidths by 19% to 28% and 13% to 23%, respectively. Furthermore, for the NC 55 arterial in Apex, non-programmed bands were detected. Therefore, the non-coordinated direction has measurable bandwidth efficiency although the average band duration was at most six seconds at this site.

Table 6.3 Bandwidth Efficiency Comparisons

Site	Type		Outbound Bandwidth Efficiency	% Difference	Inbound Bandwidth Efficiency	% Difference	Two way Bandwidth Efficiency	% Difference
A	Programmed Bandwidth		25.88	-	22.94	-	24.41	-
	Observed Dynamic Bandwidth	Day 1	46.53	79.79	38.68	68.61	42.70	74.93
		Day 2	46.36	79.13	40.00	74.37	43.26	77.22
		Day 3	46.86	81.07	34.03	48.34	40.60	66.33
		Day 4	46.61	80.10	38.29	66.91	42.55	74.31
B	Programmed Bandwidth		32.5	-	35.83	-	34.16	-
	Observed Dynamic Bandwidth	Day 1	40.49	24.58	43.73	22.05	42.11	23.27
		Day 2	41.64	28.12	45.58	27.21	43.61	27.66
		Day 3	40.35	24.15	43.71	21.99	42.03	23.04
		Day 4	39.52	21.60	43.24	20.68	41.38	21.14
		Day 5	39.73	22.25	42.68	19.12	41.20	20.61
C	Programmed Bandwidth		43.12	-	0	-	21.56	-
	Observed Dynamic Bandwidth	Day 1	49.13	13.94	3.06	-	26.10	21.06
		Day 2	49.19	14.08	2.91	-	26.05	20.83
		Day 3	49.92	15.77	2.69	-	26.31	22.03
		Day 4	49.88	15.68	3.02	-	26.45	22.68
		Day 5	48.83	13.24	1.97	-	25.4	17.81

The FHWA *Traffic Signal Timing Manual* defines *Great progression, Good progression*, and *Fair progression* based on bandwidth efficiency ranges of 37% to 100%, 25%

to 36%, and 13% to 24%, respectively. For the programmed bandwidth, FHWA arterial coordination guidelines would label the existing two-way bandwidth efficiency for US 70 in Clayton (Site A) as *Fair progression* during the AM peak plan period. In contrast, dynamic bandwidth for the four days analyzed was considerably larger than the programmed bandwidth. Calculating the efficiency of this arterial based on the dynamic bandwidth results in *Great progression* in the AM peak period.

6.1.3 Observed Dynamic Bandwidth Distribution

Figure 6-2 and Figure 6-3 illustrate the outbound and inbound dynamic bandwidth distribution for the US 70 arterial in Clayton (Site A). The programmed outbound and inbound bandwidths are 44 seconds and 39 seconds, respectively. However, most of the observed dynamic bands are considerably larger than these values. In fact, the mode of both outbound and inbound bandwidth is larger than the programmed band.

Table 6.4 provides a summary of dynamic green band frequency for the three study sites. In case of US 70 in Garner (Site B), the inbound programmed bandwidth was also the most frequently observed dynamic green band. Nonetheless, both outbound and inbound dynamic bandwidths cover a wide variety of durations at this site. Even in this case where the programmed bandwidth is the most frequently observed dynamic green band, programmed bandwidth alone do not accurately represent the reality of the coordination. For NC 55 in Apex (Site C), the inbound direction was not coordinated. However, a total of 164 dynamic bands were created over the 200 cycles. Although it should be noted that most of the non-programmed bands were shorter than 6 seconds, several green bands with durations of longer than 10 seconds were observed.

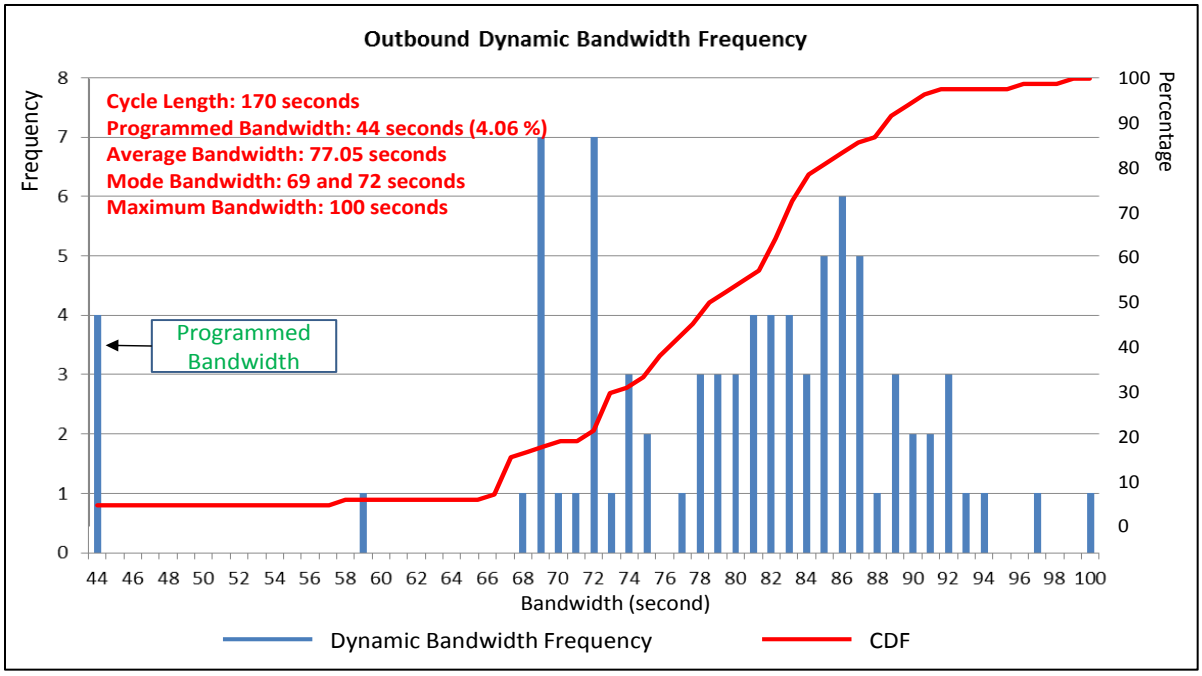


Figure 6-2 Site "A" Outbound Bandwidth PDF and CDF

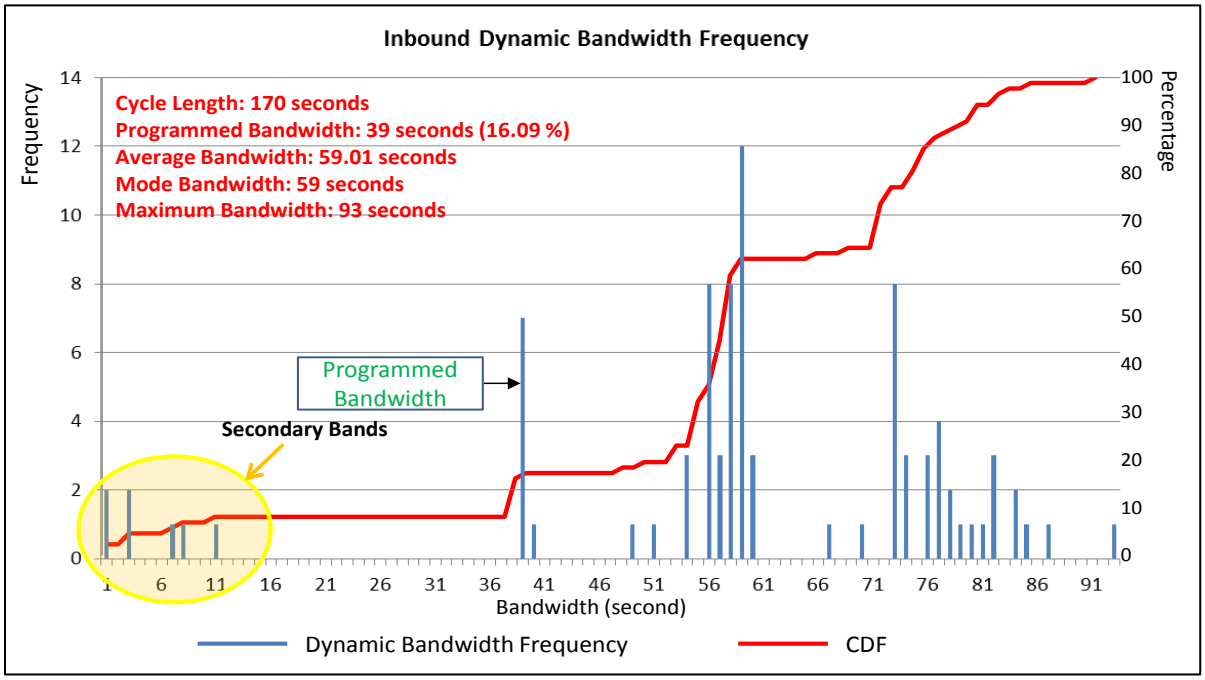


Figure 6-3 Site "A" Inbound Bandwidth PDF and CDF

Table 6.4 Dynamic Bandwidth Frequency Summary

Site	Direction	Programmed Bandwidth (Relative Frequency of Programmed Bandwidth)	Actual Dynamic Bandwidth			
			Number of Bands	Average (sec)	Mode (sec)	Max (sec)
A	Outbound	44 sec (4.06%)	84	77.05	69 and 72	100
	Inbound	39 sec (16.09%)	87	59.01	59	93
B	Outbound	39 sec (0.29%)	689	48.87	51	77
	Inbound	43 sec (22.27%)	687	53.19	43	77
C	Outbound	69 sec (5.50%)	200	79.28	75	93
	Inbound	0 sec (18.00%)	164	5.32	6	33

Programmed bandwidth's proportional frequency ranged from 0.29% to 22.27% across the three study sites. In addition, each arterial has a unique dynamic bandwidth distribution making it difficult to draw general conclusions and to represent dynamic bandwidth distributions with a single statistic.

6.2 Dynamic Bandwidth Offset Optimization Result

All three sites were analyzed with DBAT's exhaustive search routine. For each site, the current programmed offsets were highlighted in the solution set along with two globally optimal solutions. One of the highlighted optimal solutions duplicates the directional weighting implied by the programmed bandwidth, and the other one is based on equal directional weighting.

The results in Figure 6-4 were generated using this function applied to the AM peak plan for the US 70 arterial in Clayton (site A), which has three intersections. The cycle length in the AM peak plan is 170 seconds. Therefore, the total number of unique offset combinations with a 1 second search interval is $170^2 = 28{,}900$. This is a manageable search

space for the exhaustive search method. Figure 6-5 shows the total bandwidth (on the y-axis) for all the combinations of offset (on the x-axis) at this site. The x-axis is plotted using a unique ID assigned to each offset combination. The maximum sum of bandwidth with directional weighting equal to the programmed ratio for this site is 3,002 seconds (see Table 6.5 for more information). In the case of site A, the programmed offset dynamic bandwidth is quite close to the best solution found by the exhaustive search approach. The difference between them is a mere 26 seconds over a one-hour time period. In addition, there are 12 Pareto optimal solutions under equal directional bandwidth weighting, each yielding an optimal equally weighted sum of dynamic bandwidth of 2,860 seconds.

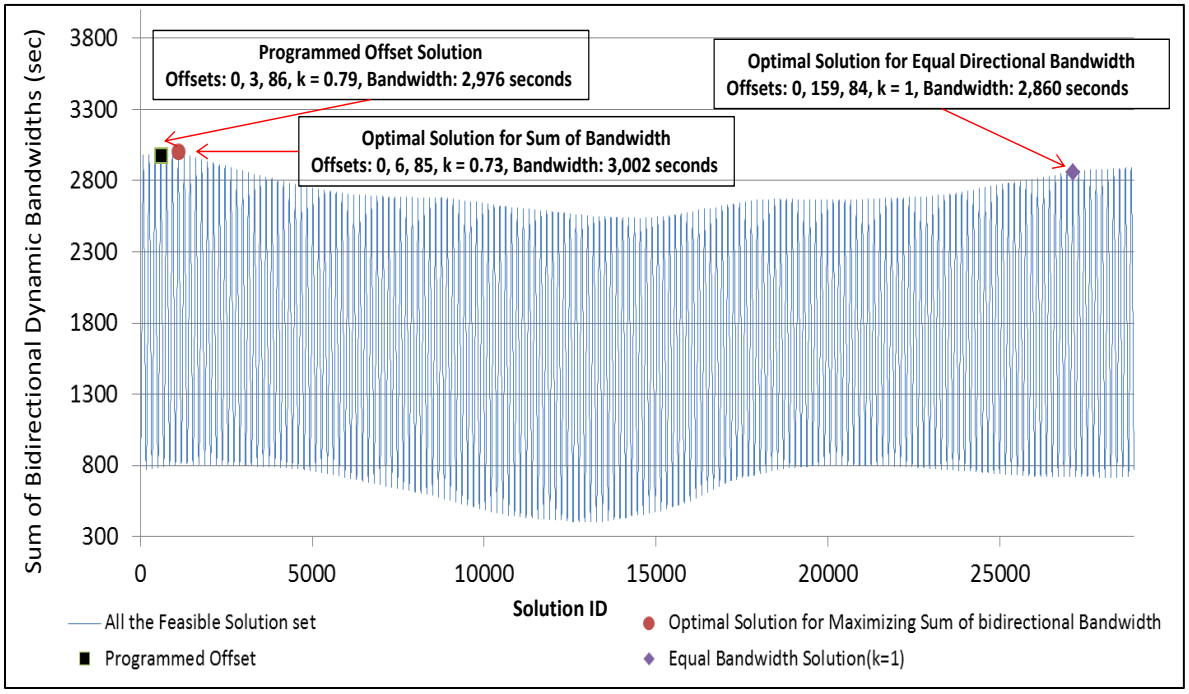


Figure 6-4 Site "A" Day 1 AM Peak Plan Exhaustive Search Result

Figure 6-5 shows the US 70 in Garner's (site B) exhaustive search results. Site B has four signalized intersections with a 120 second cycle length. Therefore, there are $120^3 = 1{,}728{,}000$ feasible offset solutions. Among those feasible solutions, Figure 6-5 highlights the two optimal solutions described above. The optimal solution with weighting based on the programmed offsets yields a dynamic bandwidth total of 14,345 seconds and the equal weighting optimal solutions yields 14,224 seconds. Compared with programmed offsets, the global optimal offset provides 427 seconds more total bandwidth over the four hours of the AM peak plan.

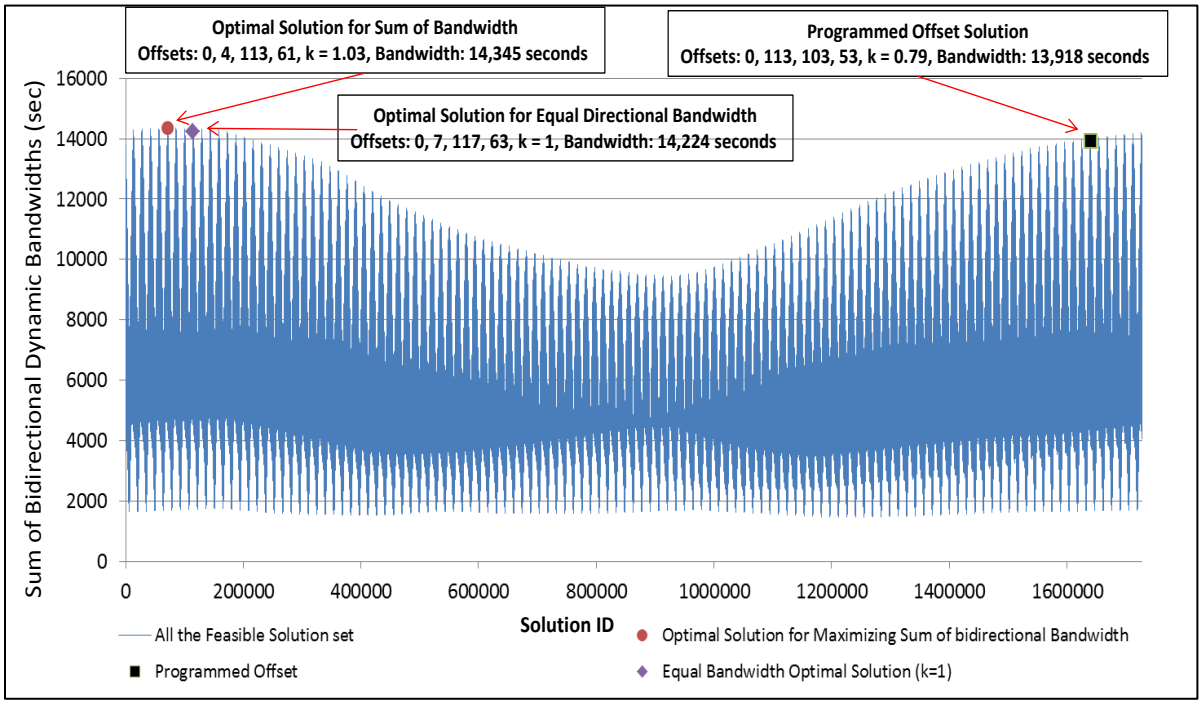


Figure 6-5 Site "B" Day 1 AM Peak Plan Exhaustive Search Result

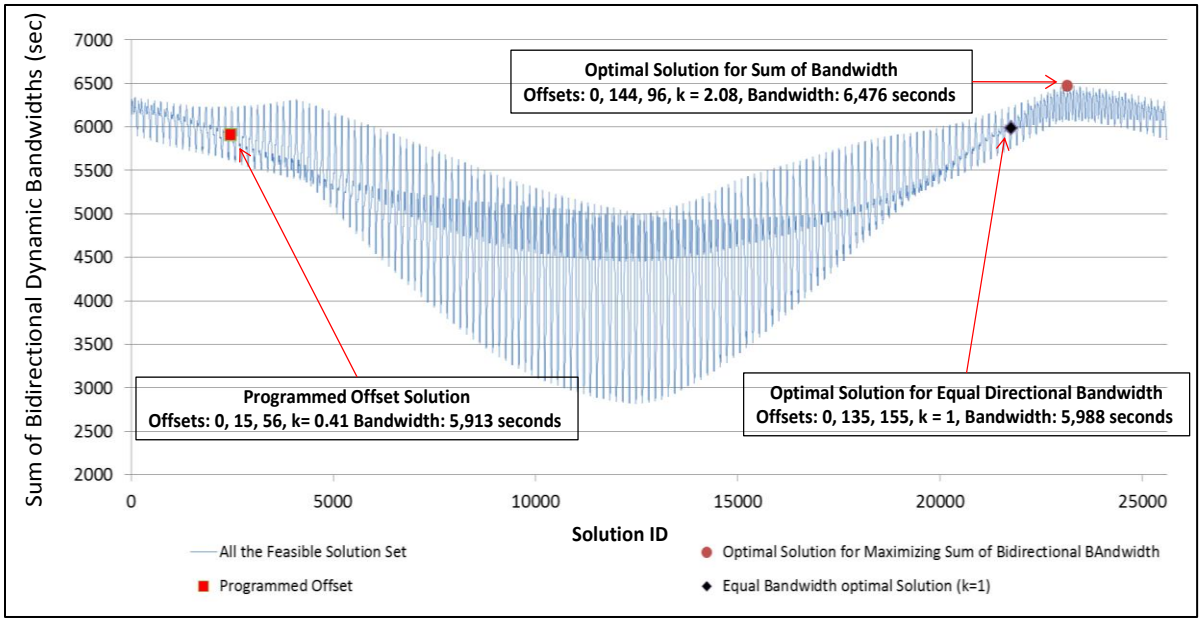


Figure 6-6 Site "C" Day 1 AM Peak Plan Last 3 Intersection Exhaustive Search Result

Figure 6-6 shows the results of the exhaustive search method for the NC 55 arterial in Apex (site C). This arterial has seven intersections. Analyzing all seven intersections with the exhaustive search is impractical, and therefore the study case uses the last three intersections only (intersection 5, 6, and seven in Figure 4-7).

The selected time of day plan is AM peak from 7:00 to 9:10 and the cycle length is 160 seconds. From the exhaustive search, the global directionally weighted solution has a total sum of bandwidth of 6,476 seconds. In addition, it can be seen in Figure 6-6 that this global solution is. Robust against offset changes as minor changes in offset reduce the total sum of bandwidth to around 6,000 in the worst case. The low sensitivity to offset permutation of this solution indicates that the corresponding offsets can provide good progression even if expected early return to green and green extension values vary from those

represented in the analyzed cycles. Compared to the programmed offset sum of dynamic bandwidth (5,913 seconds), the optimal offset solution provides 563 seconds more total bandwidth. This means that the optimal solution would have provided an average of 12.5 seconds more bandwidth per cycle across 45 cycles analyzed in site C.

Table 6.5 Programmed Dynamic Bandwidth with Un-weighted Optimal Solutions

Site	Solution type	k	Offset (sec)	Outbound Sum of Bandwidth (sec)	Inbound Sum of Bandwidth (sec)	Overall % Change
A	Observed Dynamic Bandwidth	0.79	0,3,86	1,661	1,315	-
	Optimal Solution for Sum of Bandwidth	0.73	0,6,85	1,729	1,273	0.87%
	Optimal Solution for Equal Bandwidth	1	0,159,84	1,430	1,430	-3.9%
B	Observed Dynamic Bandwidth	1.08	0, 113, 103, 53	6,698	7,220	-
	Optimal Solution for Sum of Bandwidth	1.03	0, 4, 113, 61	7,079	7,266	3.07%
	Optimal Solution for Equal Bandwidth	1	0, 7, 117, 63	7112	7112	2.20%
C	Observed Dynamic Bandwidth	0.41	0, 15, 56	4,176	1,737	-
	Optimal Solution for Sum of Bandwidth	2.08	0, 144, 96	2,099	4,377	9.52%
	Optimal Solution for Equal Bandwidth	1	0, 135, 155	2,994	2,994	1.27%

6.3 Evaluation of the Effectiveness of Optimal Solution

Using an exhaustive search, the set of offsets that are expected to provide the highest value of weighted sum of bandwidths on the US 70 arterial in Clayton, NC (Site A) were found for each time of day plan. To find these offsets, two weeks data from split monitor logs were collected in April 2013 and then analyzed. An exhaustive search method was used to cover all possible combinations of offsets. For each set, total band size and weighted sum of band size were determined. Note that these values are "expected" to be observed in the field if the offsets were to be implemented; however, due to variations in demand level, we may not see identical values. The set of offsets that provided the highest expected weighted sum of bandwidth was selected and implemented in the field. For a period of two weeks in August 2013, green durations and start times were recorded to determine the band size in each direction, total band size, and weighted sum of band size. In addition, arterial through travel times were collected before and after offsets were changed using Bluetooth units.

Table 6.6 Site "A" April Two weeks Split Monitor data DBAT Process Result

Cycle Length (sec)	Plan Start (Plan End)	Plan	Offset (sec)*		EB Band Size (sec)	WB Band Size (sec)	Total Band Size (sec)	Weighted Sum of Band (% Change) **	EB (WB) Weight ***
			Int. 2	Int. 3					
170	6:20 (7:30)	April field Plan	3	86	1,547	1,202	2,749	1,325	0.36 (0.64)
		Estimated Optimal Plan****	0	85	1,524	1,229	2,753	1,334 (0.68%)	
140	7:30 (9:30)	April field Plan	135	73	2,445	2,478	4,923	2,463	0.46 (0.54)
		Estimated Optimal Plan****	136	79	2,206	2,723	4,929	2,485 (0.89%)	
120	9:30 (11:30)	April field Plan	108	62	2,458	2,237	4,695	2,365	0.58 (0.42)
		Estimated Optimal Plan****	107	60	2,519	2,241	4,759	2,402 (1.56%)	
130	11:30 (13:30)	April field Plan	118	53	2,411	1,709	4,121	2,140	0.61 (0.39)
		Estimated Optimal Plan****	117	57	2,327	1,908	4,235	2,165 (1.17%)	
150	13:30 (16:30)	April field Plan	4	47	4,948	1,848	6,797	3,913	0.67 (0.33)
		Estimated Optimal Plan****	137	54	4,783	2,774	7,557	4,112 (5.09%)	
220	16:30 (18:30)	April field Plan	211	3	3,080	1,895	4,975	2,735	0.71 (0.29)
		Estimated Optimal Plan****	172	5	3,156	1,853	5,009	2,777 (1.54%)	

* Intersection 1 is the reference intersection. Therefore, its offset is always set at zero

** Objective Function: Two week average directional band size multiplied by directional weight

*** Two week average directional volume ratio

**** Maximized weighted sum of band using two week split monitor

Table 6.6 summarizes the actual and estimated optimal EB, WB, Total, and Weighted sum of bandwidths at Site A. In all time of day plans except for one (shaded), total band size and weighted sum of bandwidths are almost equal. This means that the implemented offsets for these time-of-day plans were quite close to the estimated optimal ones. On the other hand, for the first pm plan, the result of the exhaustive search estimated a significant 760 seconds improvement in total band size, if the offsets were changed from 0, 4, and 47 to 0, 137, and 54 at intersections 1, 2, and 3 respectively. Therefore, after field implementation, we expect to observe a significant increase in total band size as well as considerable decrease in travel time in plan 5. Figure 6-7 shows DBAT exhaustive search results for eight weekdays at Site "A".

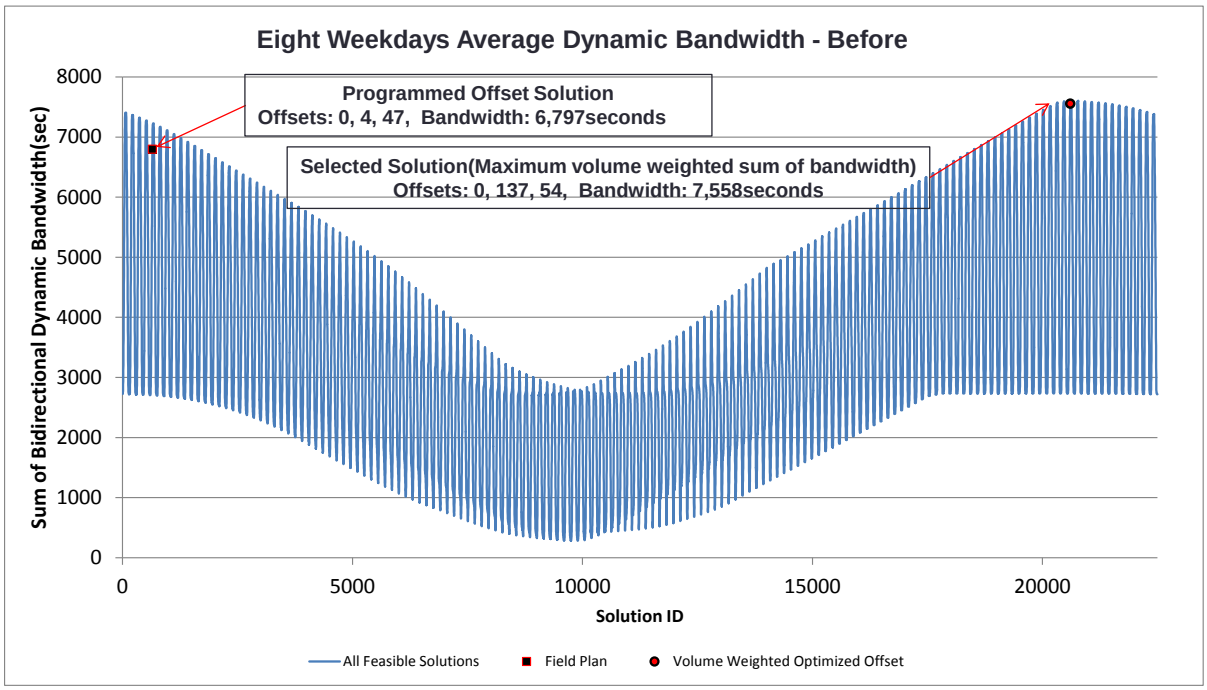


Figure 6-7 DBAT Exhaustive Search Result in Plan 5 (Before)

In order to ascertain that any potential change in arterial travel time is solely due primarily to changes in the offsets and not to different signal operations or demand (volume), before and after detector counts for through movements and coordinated phase's used green times were studied.

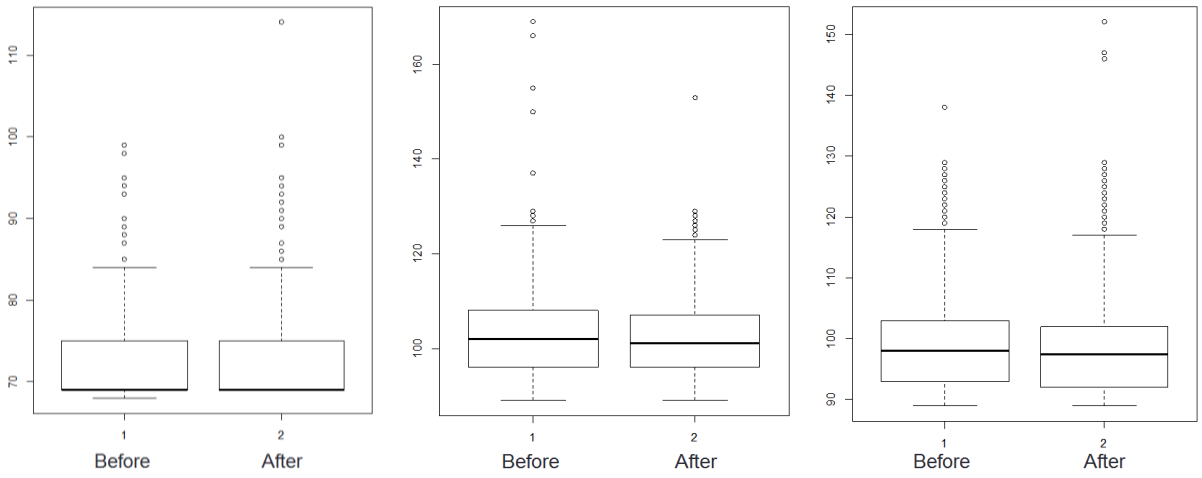
A t-test was used for testing demand difference in before and after conditions. The test was conducted under a significant level of 0.05 and mean difference of 0.4vehicles/minute. Among the six different time of day plans, only the fifth plan shows meaningful bandwidth improvement so 0.4vehicles/minute was decided to detect one vehicle difference in the fifth plan cycle (150 seconds). Table 6.7 summarizes important before and after data. The results show that the average number of detector calls per lane per minute did not significantly change before and after field implementation of the optimal offsets at any intersection. This indicates that through volume level did not appreciably change in the two data collection periods.

Table 6.7 Test Results for Demand Variation Before and After Offset Change

Intersection	Direction	Number of lane	Data Collection Duration (min)		Average number of Detector Calls (veh/min/ln)		Standard Deviation of Detector Calls (veh/min/ln)		Power	p-value
			Before	After	Before	After	Before	After		
Shotwell & US 70	EB	2	2,887	2,890	8.179	8.143	4.449	4.5	0.960	0.379
	WB	3	4,329	4,336	3.206	3.181	4.008	4.018	0.999	0.384
S Moore & US 70	EB	2	2,896	2,883	10.37	10.28	6.07	6.081	0.804	0.285
	WB	3	4,341	4,318	5.459	5.385	3.477	3.757	1	0.169
Robertson & US 70	EB	2	2,895	2,573	9.77	9.663	6.052	5.446	0.821	0.247
	WB	3	4,341	3,856	4.921	4.88	2.73	2.77	1	0.251

In addition, the Kolmogorov-Smirnov test shows no significant change in the used green time distribution before and after field implementation of the optimal offsets, see Table 6.8. Since the distribution of used greens of the coordinated phases is highly correlated with the early return to greens and green extensions, and they both are highly correlated with the demand pattern on non-coordinated phases, the analysis suggests that the demand level at non-coordinated phases did not significantly change either. As a result, there is no strong evidence to conclude that the demand level in the before and after data collection phases are different.

Figure 6-8 Show boxplot result for before and after coordinated phases used green time. The plot illustrate there are no significant difference between before and after used green time.



a) Shotwell Intersection b) Moore Intersection c) Robertson Intersection

Figure 6-8 Boxplot Result for Before and After Used Green Time

Table 6.8 K-S Test Result for Before and After green used time of Coordinated Phase

Intersection	Phase	D	P-Value
Shotwell & US 70	EB (Phase2)	0.041	0.407
	WB (Phase 6)	0.013	0.913
S Moore & US 70	EB (Phase2)	0.026	0.695
	WB (Phase 6)	0.007	0.971
S. Robertson & US 70	EB (Phase2)	0.024	0.998
	WB (Phase 6)	0.034	0.920

Figure 6-9, Figure 6-10 and Figure 6-11 shows Shotwell Dr. intersection's before and after coordinated phase used green time distribution.

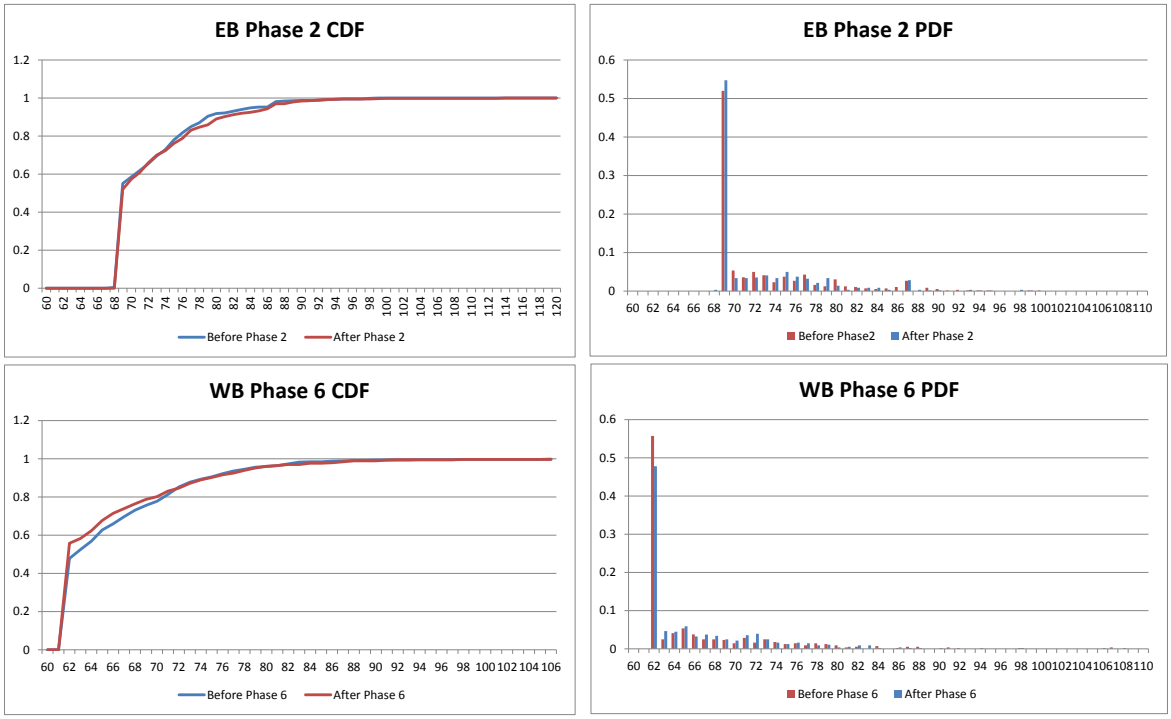


Figure 6-9 Coordinated Phase Used Green Time Comparison (Shotwell Dr. Intersection)

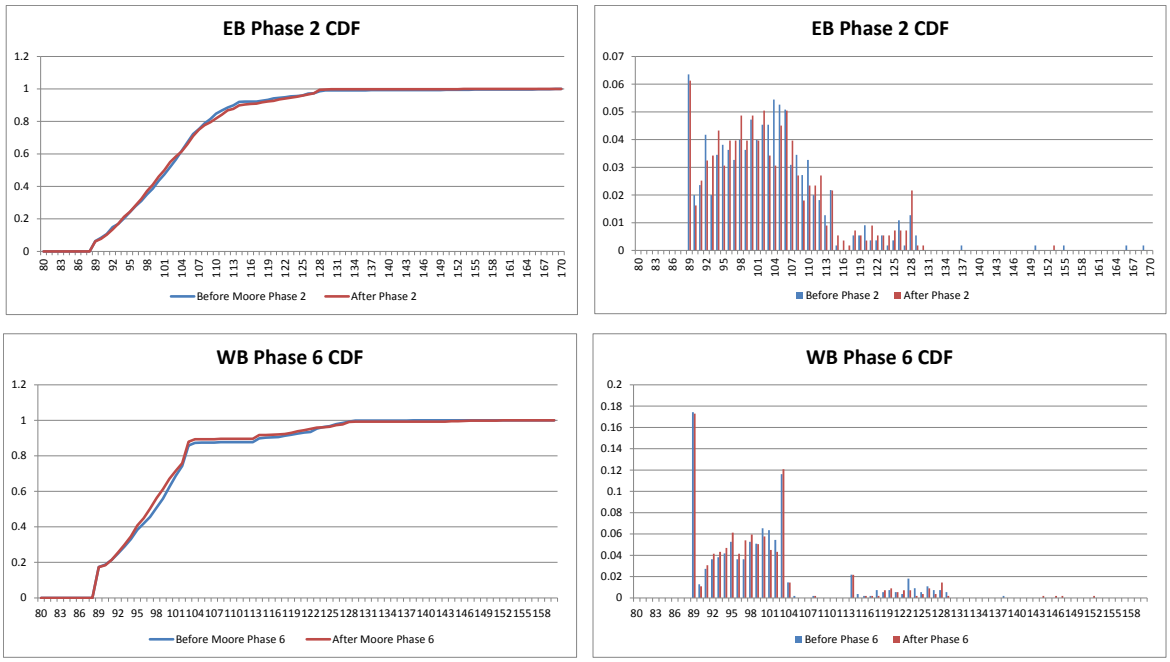


Figure 6-10 Coordinated Phase Used Green Time Comparison (More St. Intersection)

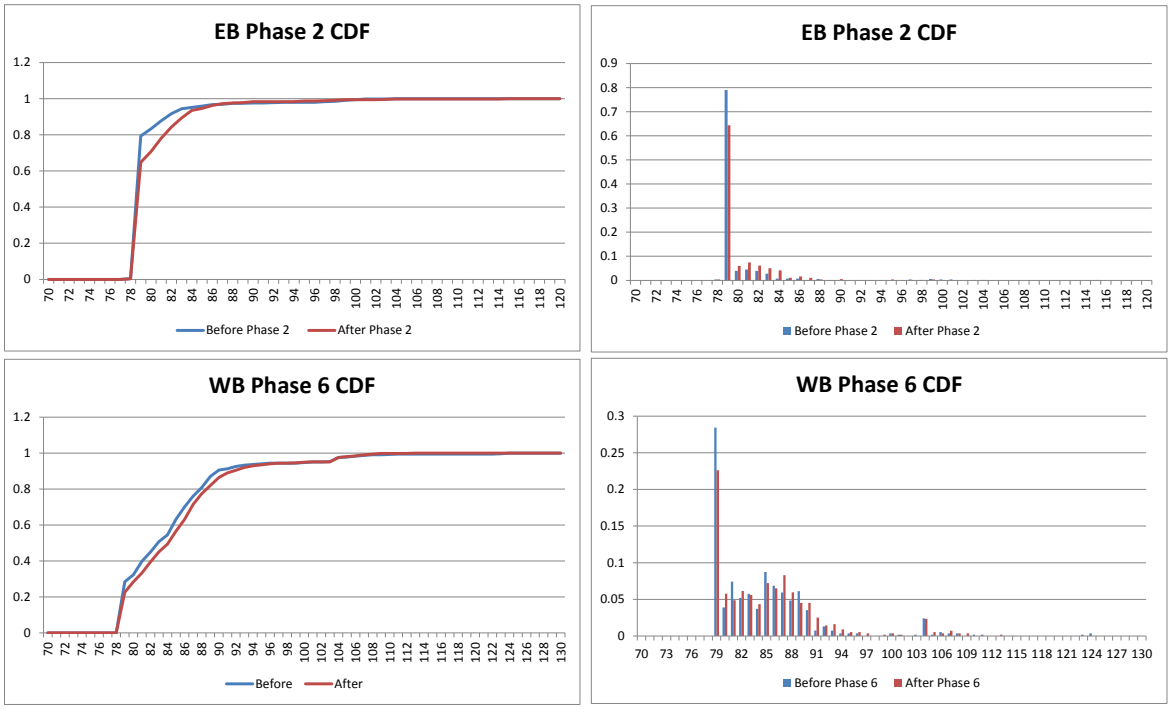


Figure 6-11 Coordinated Phase Used Green Time Comparison (Robertson St. Intersection)

Figure 6-12 and Figure 6-13 presents cumulative travel time distributions for the study site before and after field implementation of the optimal offsets for the 1:30 pm Plan. The cumulative travel time distributions do not show a considerable change in EB travel time, the peak direction (see Figure 6-12); however, they show a significant shift in the distribution towards lower travel times in the westbound direction as a result of using the optimal offsets (see Figure 6-13). This shows that while optimizing the offsets did not significantly change travel time and bandwidth on the major direction, it significantly reduced travel time in the minor direction.

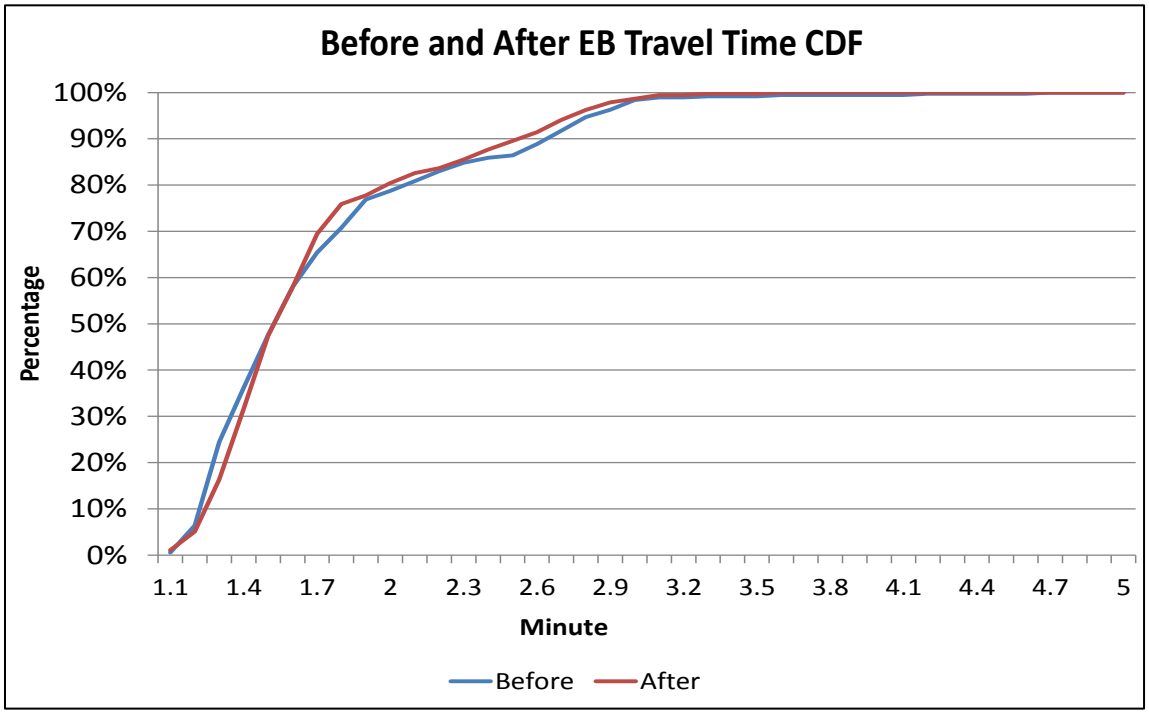


Figure 6-12 EB Before and After Travel Time CDF's for 1:30 pm Plan at Site A

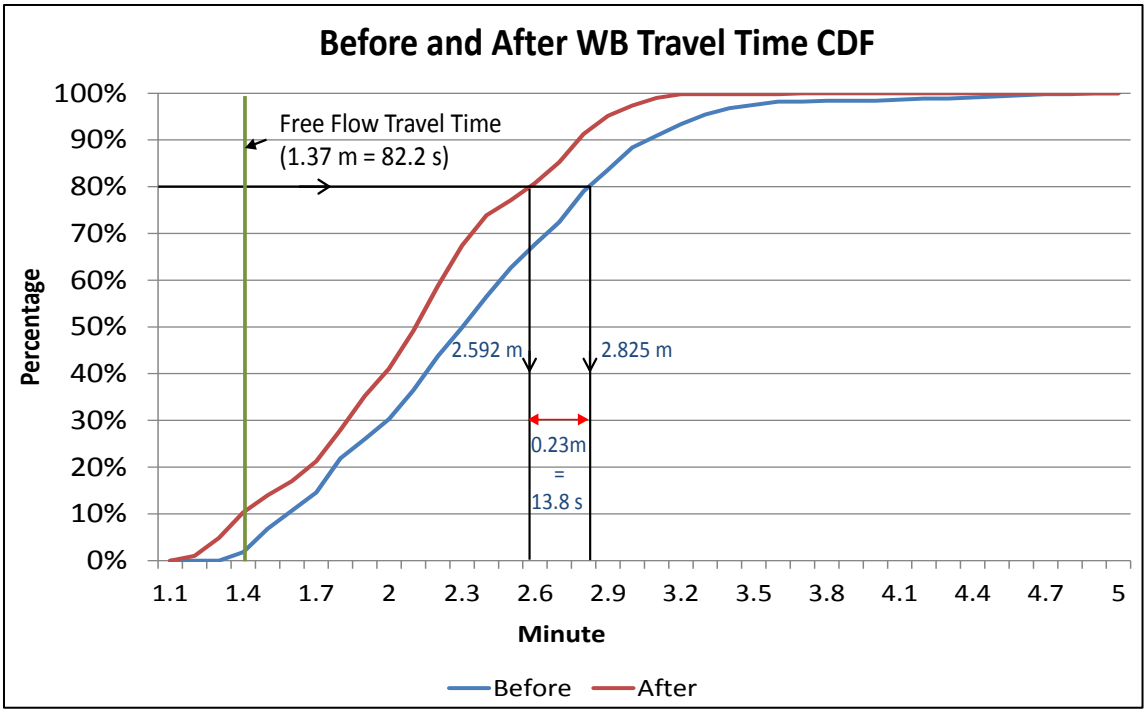


Figure 6-13 WB Before and After Travel Time CDF's for 1:30 pm Plan at Site A

Table 6.9 Before and After Travel Time Observation at Site "A summarizes the before and after directional band size, total band size, weighted sum of band size, average travel time, and standard deviation of travel time in the study site for all time of day plans. As expected, there is not a considerable change in average travel time in most time of day plans. The main reason is that using the optimal set of offsets did not significantly change the "expected" band size for those plans. On the other hand, for the 1:30 pm plan, the optimal offsets reduced average travel time in the westbound direction by 10.5% without creating a significant change in the eastbound average travel time.

Table 6.9 Before and After Travel Time Observation at Site "A"

Cycle Length (sec)	Plan Start (Plan End)	Plan	Offset *		EB Band Size (sec)	WB Band Size (sec)	Total Band Size (sec)	Weighted Sum of Band (Objective Function)	EB Weight Value (WB Weight Value)	Bluetooth Sample Size		Average Travel Time (sec)		STDEV Travel Time (sec)	
			Int. 2	Int. 3						EB	WB	EB	WB	EB	WB
170	6:20 (7:30)	Before	3	86	1,547	1,202	2,749	1,325	0.36 (0.64)	94	297	103	123	33.3	38.9
		After	0	85	1,543	1,235	2,778	1,346		97	263	104	122	32.5	39.1
140	7:30 (9:30)	Before	135	73	2,445	2,478	4,923	2,463	0.46 (0.54)	254	417	104	109	29.4	31.6
		After	136	79	2,223	2,749	4,973	2,507		232	420	107	107	28.8	33.5
120	9:30 (11:30)	Before	108	62	2,458	2,237	4,695	2,365	0.58 (0.42)	223	284	102	110	37.1	35.2
		After	107	60	2,589	2,283	4,873	2,461		222	295	98	107	32.2	34.6
130	11:30 (13:30)	Before	118	53	2,411	1,709	4,121	2,140	0.61 (0.39)	264	315	115	104	37.1	33.5
		After	117	57	2,378	1,890	4,267	2,187		273	348	116	103	35.2	31.9
150	13:30 (16:30)	Before	4	47	4,948	1,848	6,785	3,913	0.67 (0.33)	379	441	103	141	34.6	40.0
		After	137	54	4,802	2,751	7,553	4,117		375	494	102	126	33.8	29.1
220	16:30 (18:30)	Before	211	3	3,080	1,895	4,975	2,735	0.71 (0.29)	398	308	125	125	44.3	41.8
		After	172	5	3,198	1,860	5,058	2,810		326	316	122	125	44.8	42.3

* Intersection 1 is the reference intersection. Therefore, its offset is always set at zero.

Figure 6-7 illustrates before dynamic bandwidth exhaustive search result and it estimate 7,558 seconds bandwidth solution under 0, 137 and 54 second offset condition. Figure 6-14 shows exhaustive search result for after condition. The observed real field dynamic bandwidth is 7,554 second. There are only 4 seconds difference between estimation value and observed value. This supports the robustness of proposing method for dynamic bandwidth optimization.

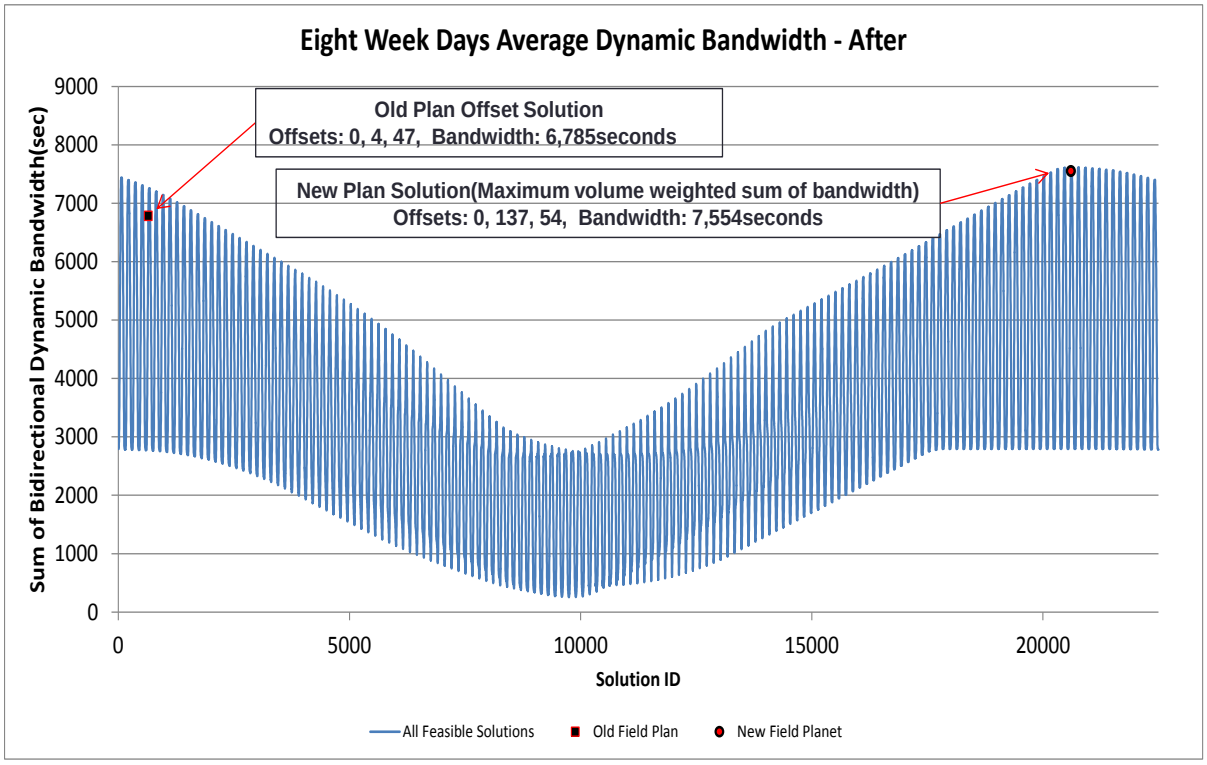


Figure 6-14 DBAT Exhaustive Search Result in Plan 5 (After)

It is very important to note that the optimal offsets were not fine-tuned in the field, yet they still yielded a significant reduction in travel time in the westbound direction without

creating any significant change in other travel times. This is a very important finding and shows that the proposed approach has great potential for eliminating the fine-tuning process, which is time consuming and costly and cannot guarantee an optimal performance especially when the number of intersections increases. Our proposed approach, by analyzing two weeks of used green times could accurately account for patterns that are expected to be observed in early returns to green and green extensions.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

The dynamic bandwidth optimization method is developed and verified through hypothetical network testing and comparisons with field evaluation. At the same time, dynamic bandwidth characteristics and enhanced MAXBAND formulation are also proposed and verified. A summary of the findings and conclusions drawn are as follows:

Enhanced MAXBAND formulation results:

- An enhanced MAXBAND formulation is introduced to overcome the potential for the MAXBAND family of bandwidth optimization MILPs to yield suboptimal solutions under certain conditions.
- Using hypothetical network scenarios, the MAXBAND over constrainted problem is revealed, and the proposed enhanced formulation is tested. Optimization performance of the original and enhanced formulations are compared using two hypothetical test network scenarios and the real world arterial used in Little's 1966 paper.
- The test result supports the notion that the new formulation is consistent with the original MAXBAND formulation in cases where over constraint and suboptimality are not an issue and that it also yields an optimal solution in cases where the original formulation suffers from suboptimality due to over constraint of the bandwidth ratio.

- The modified objective function and constraints in the new formulation are proposed to replace MAXBAND's formulation as well as new editions to the MAXBAND family of optimization models such as MULTIBAND, MULTIBAND-96, and AM-BAND. Although the MILP formulations for MULTIBAND, MULTIBAND-96, and AM-BAND have increasingly developed innovative objective functions and constraints compared to the original MAXBAND MILP, all the MAXBAND-related models have similar constraints incorporating directional bandwidth target value *k*.
- The proposed new formulation overcomes the MAXBAND MILP family over-constrained problem assuming a fixed duration of the effective red and green intervals. However, coordinated arterial green in most modern signal systems is not static but varies dynamically on a cycle-by-cycle basis yielding in turn progression bandwidths that vary from cycle-by-cycle. Therefore, further development of a proposed enhanced MAXBAND formulation as well as all the other arterial performance optimization efforts should be designed to consider the impact of dynamic bandwidths.

Dynamic bandwidth characteristic monitoring results:

- A dynamic bandwidth analysis methodology that provides actual dynamic bandwidth durations using closed loop signal data to assess the performance of semi-actuated coordinated arterial streets was developed.

- The methodology was designed to calculate dynamic bandwidth from signal ATMS data. Traditionally, calculated programmed green bandwidth provides limited insight for evaluating the quality of arterial coordination because it cannot assess the impact of early return to green and green extension for the coordinated phases.
- Detailed analysis at three locations confirmed that real world coordinated green times along signalized arterials have complex multimodal distributions. This finding underscores the value of cycle-by-cycle dynamic bandwidth analysis for both the evaluation and optimization of timing plan offsets. The developed methodology represents an important first step in formalizing and enabling dynamic bandwidth using cycle-by-cycle phase duration data from signal based ATMS.
- The summary of key findings from dynamic bandwidth monitoring include:
    1. Dynamic bandwidth totals and bandwidth efficiencies are considerably larger than the corresponding programmed bandwidth and bandwidth efficiency.
    2. Dynamic bandwidths may include secondary and non-programmed bands.
    3. Dynamic bandwidths distributions are complex and characteristically multi-modal.

Dynamic bandwidth optimization results:

- Two dynamic bandwidth optimization methods are introduced in this dissertation.
- First, an exhaustive search method by developed dynamic bandwidth analysis tool (DBAT) provides complete feasible solution space, at a one second resolution, for the combination of all offsets. This result provides important information to understand the solution space thereby allowing for better arterial coordination.
- The global optimal solution can be detected by DBAT exhaustive search result. However, limitations exist when the analysis arterial consists of many intersections since it exponentially increases the solution space and calculation time.
- Second, this study presents a linear programming (LP) formulation to enable dynamic bandwidth maximization on semi-actuated arterial streets. The methodology relies on the use of archived signal log data regarding phase start and end times in each cycle in both directions. Those data are used to optimize the offsets that maximize the variable system bandwidth across multiple cycles constituting a coordination plan period.
- The formulation is also flexible to be used to optimize signal offsets using programmed (fixed) green durations at each intersection.
- The proposed formulation offers five significant enhancements compared to traditional methods:

1. The formulation is strictly linear (complexity of class P) as opposed to the traditional mixed integer programming formulations (complexity of class NP-hard).
2. It can work with either static or dynamic (cycle varying) green durations.
3. Traditional bandwidth optimization methods have explicit constraints to enforce bandwidth allocation to be proportional to directional demand. However, as longs as the minimum required bandwidth is allocated to each direction, a proportional allocation of bandwidth is not necessary and in fact, can result in reporting sub-optimal solutions. The proposed formulation overcomes this shortcoming.
4. The proposed formulation predicts the maximum proportion of traffic demand that can be served in the bandwidth, a unique attribute absent from other formulations found in the literature.
5. As an added benefit, the formulation enables the analyst to carry out a slack analysis processing step which reports the range of offsets at non-critical intersections within which the optimal objective function value is unaffected.

- The proposed LP model was tested on several hypothetical arterials and on the Euclid Avenue arterial in Cleveland, Ohio, a site that was cited in Little's paper (Little, J. D. C., 1966). The application to hypothetical scenarios illustrates MAXBAND over-constrained formulation, which occurs when the

maximum possible inbound and outbound bandwidth sizes are different. It also shows how the new LP formulation overcomes this problem.

- The proposed LP model, when applied to Euclid Avenue, is shown to provide an identical bandwidth solution to MAXBAND in addition to specifying all possible optimal solution sets. In addition, it also provides all global solutions by slack analysis.
- Finally, the optimal solution was field tested on the US 70 arterial in Clayton, North Carolina using the dynamic green time information provided by the Advanced Traffic Management System (ATMS). The results indicated that the optimal solution improved the sum of dynamic bandwidths by 768 seconds (or about 16 seconds/cycle).
- Furthermore, field measurements of travel time after implementing the proposed optimal solution yielded a 10% decrease in arterial travel time. Even without field fine tuning, the proposed algorithm has the potential to produce very high-quality solutions that are ready for field implementation.

7.2 Future Work

This study explored real world dynamic bandwidth characteristics and developed dynamic bandwidth optimization methods. The proposed methods use signal ATMS data. The following lists future studies and work:

- Since both exhaustive search and LP methods use actual field dynamic green information to search and calculate the optimal global solution, they cannot conduct phase sequence optimization, which is one of the major functions of MAXBAND. Chapter 3 shows how phase sequence is related to coordinated phase dynamic green. For example, a lag phase must use all of its allotted green time since it cannot be terminated until the concurrent phase (one of the coordinated phases) is terminated. This relationship between phase sequence and coordinated phases' dynamic green is one of the future study items to explore. If a significant relationship can be found, it could be possible to develop enhanced LP formulation which could include a phase sequence optimization process.
- Dynamic bandwidth may include secondary bandwidth, but the function of secondary bandwidth is not clear. It could have both positive and negative effects. For example, it may have a negative effect from a safety aspect since the secondary bands are generally narrow. When drivers try to use a small secondary band to avoid stopping, it may increase the possibility of a crash. On the other hand, when secondary bands are safely utilized by road users, it will increase arterial performance from a mobility aspect. From this future

study, a new dynamic bandwidth objective function could be developed, which would either minimize or include the secondary bandwidth for bandwidth maximization.

- Another future study item is related to the exhaustive search method. The exhaustive search method has many positive functions with the only limitation being its calculation time. For example, it can provide all the feasible solutions with corresponding directional bandwidth ratio (k) space which can include or exclude secondary bandwidth. Therefore, a more efficient search algorithm needs to be studied which would include the sensitivity of searching resolution. In addition, the methodology to generate the optimal range of bandwidth and critical intersection information under dynamic green condition also need to be studied. Generally, both LP and exhaustive search methods provide unique (only one) offset solution due to its own characteristic introduced in chapter 5. That means all the intersections are critical intersections under the dynamic green condition. However, the sensitivity of each intersection offset against objective function value may be different from each other. Therefore, it will be useful information to provide the optimal range of bandwidth and more or less critical intersection information by using sensitivity test.
- As mentioned in the literature review chapter, there are three common arterial performance measures: travel time, number of stops, and bandwidth. For the bandwidth, FHWA provides a guideline for bandwidth efficiency and

attainability. However, the guideline may rely on static (programmed) green. If it is true, a new guideline needs to be developed and studied for dynamic bandwidth, since there are many differences between programmed bandwidth and dynamic bandwidth, which is introduced in chapter 6.

- Finally, more variety of real world case studies are good future work. For example, applying dynamic bandwidth optimization method to DDI (Diverging Diamond Interchange) would be a good case study. Offset optimization is very important for DDIs since general DDIs use two phase control while the adjacent intersections generally use more than two phases, so there is always a maximum g over C imbalance issue between the DDI and its adjacent intersection.

REFERENCES

Adaptive Traffic Control Systems: Domestic and Foreign State of Practice, (2010). NCHRP Report 403, Federal Highway Administration, Washington, DC.

Andrews, C.M. and S.M. Elahi (1997). Evaluation of New Jersey Route 18 OPAC/MIST Traffic Control System. Presented at 76th Annual Meeting of the Transportation Research Board, Washington, DC.

Application of UTCS First Generation Control Software in New Orleans, (1978). Final Report, FHWA-RD-78-3, U.S. Department of Transportation, Federal Highway Administration, Washington, DC.

Balke, K., Charara, H., and Parker, R. (2005). Development of a Traffic Signal Performance Measurement System (TSPMS). Texas Transportation Institute. College Station, TX.

Brooks W.D. (1965). Vehicular Traffic Control: Designing Traffic Progression Using a Digital Computer. IBM-Data Processing Division, Kingston N.Y.

Chang, E. C. P., Cohen, S. L., Liu, C., Chaudhary, N. A., and Messer, C. (1988). MAXBAND-86: program for optimizing left-turn phase sequence in multi-arterial closed networks. *Transportation Research Record*, No. 1181: 61-67.

Chang, E. C. P. (1996). Guidelines for Actuated Controllers on Coordinated Systems. *Transportation Research Record*, No. 1554, 61-73.

Corporation of Metropolitan Toronto (1974-1976). Improved Operation of Urban Transportation Systems. Vol. 1 - 3, Toronto, Canada.

Day, C.M., E.J. Smaglik, D.M. Bullock, and J.R. Sturdevant (2008). Quantitative Evaluation of Actuated Coordinated Versus Non-actuated Coordinated Phases, *Transportation Research Record*, No. 2080, Transportation Research Board of the National Academies, Washington, D.C., pp. 8-21.

Day, C.M., R. Haseman, H. Premachandra, T.M. Brennan, J.S. Wasson, J.R. Studevant, and D.M. Bullock (2010). Visualization and Assessment of Arterial Progression Quality Using High Resolution Signal Event Data and Measured Travel Time. Transportation Research Board Annual Meeting, Paper No. 10-0039, Transportation Research Board of the National Academies, Washington, DC.

Dell'Olmo, P. and P.B. Mirchandani. REALBAND: An Approach for Real-Time Coordination of Traffic Flows on a Network. *Transportation Research Record*, No. 1494, Transportation Research Board of the National Academies, Washington, DC, pp. 106-116, 1995.

FHWA Announces Policy on Support of the UTCS-Enhanced software, (1985). CCSAG Newsletter, ITE Journal.

Gartner, N. H.(1983). OPAC: A Demand-Responsive Strategy for Traffic Signal Control. Transportation Research Record 906. Transportation Research Board, National Research Council, Washington DC.

Gartner, N. H., Assmann, S. F., Lasaga, F., and Hou, D. L. (1991). A Multiband Approach to Arterial Traffic Signal Optimization. *Transportation Research Part B*, 25(1), 55-74.

Gartner, N. H. (2001). Optimized Policies for Adaptive Control (OPAC), Session 1: Principles of Operation, presented at the TRB Annual Meeting, Adaptive Traffic Signal Control Systems Workshop, Washington, D. C.

Gartner, N.H., F.J. Pooran, and C.M. Andrews (2002). Optimized Policies for Adaptive Control Strategy in Real-Time Traffic Adaptive Control Systems: Implementation and Field Testing. *Transportation Research Record*, No. 1811, Transportation Research Board of the National Academies, pp. 148-156.

Gartner, N. H., and Stamatiadis, C. (2002). Arterial-Based Control of Traffic Flow in Urban Grid Networks. Math. Comput. Model., 35(5), 657-671.

Girianna, M. and R. F. Benekohal (2004). Using Genetic Algorithms to Design Signal Coordination for Oversaturated Network. *Journal of Intelligent Transportation Systems*, Volume 8, Issue 2, pp 117-129.

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

Highway Capacity Manual (2010). Transportation Research Board, National Research Council, Washington D.C.

Holroyd, J.; Hillier, J.A. (1971). The Glasgow Experiment: PLIDENT and After. RRL Report 384

Hunt, P. B., D. I. Robertson, R. D. Bretherton, and R. I. Winton (1981). SCOOT- A Traffic Method of Coordinating Signals. Laboratory Report No. LR 1014. Transportation and Road Research, Crowthorne, Bershire, England.

Liao, L. C. (1998). A Review of the Optimized Policies for Adaptive Control Strategy (OPAC). California PATH Working Paper, University of California, Berkeley.

Lin, L. T., Tung, L. W., and Ku, H. C. (2010). Synchronized Signal Control Model for Maximizing Progression along an Arterial. *Journal of Transportation Engineering*, 136(8), 727-735.

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

Liu, H.X., and Ma, W. (2009). A virtual vehicle prove model for time-dependent travel time estimation on signalized arterials. *Transportation Research Part C*, 17, 11-26.

Lowrie, P.R. (1982). The Sydney cooperative Adaptive Traffic System – Principles, Methodology, Algorithms. In Proceedings of International Conference on Road Traffic Signaling, Institution of Electrical Engineers, London, UK., Vol. 207, pp. 67-70.

Lowrie, P. R. (1992). SCATS – Sydney Coordinated Adaptive Traffic System – A Traffic Responsive Method of Controlling Urban Traffic. Roads and Traffic Authority, Sydney Australia.

Lu, K., Zeng, X., Li, L., and Xu, J. (2012). Two-way Bandwidth Maximization model with Proration Impact Factor for Unbalanced Bandwidth Demand. *Journal of Transportation Engineering*, 138(5), 527-534.

Luyanda, F., D. Gettman, L. Head, S. Shelby, D. Bullock, and P. Mirchandani (2003). ACS-Lite Algorithmic Architecture: Applying Adaptive Control System Technology to Closed-Loop Systems. *Transportation Research Record*, No. 1856, Transportation Research Board of the National Academies, Washington, DC, pp. 175-184.

MacGowan, J., and Fullerton, I. J. (1979). Development and Testing of Advanced Control Strategies in the Urban Traffic Control System, Public Road, Vol. 43, No. 3.

Manual on Uniform Traffic Control Devices for Streets and Highways (2009). U.S. Department of Transportation, Federal Highway Administration.

McShane, W.R., R.P. Roess and Elena S. P. (1990). Traffic Engineering. Englewood Cliffs, New Jersey: Prentice Hall.

Messer, C. J., R.H. Whitson, C.L. Dudek, and E. J. Romano (1973). A Variable sequence Multi Phase Progression Optimization Program, Highway Research Record 445, pp. 24-33.

Miller, A.J. (1963). A computer control system for traffic network. Proc., 2nd Int. Symp. On Theory of Road Traffic Flow, London, pp. 201-220.

Mirchandani, P. and Head, L. (2001). RHODES: A Real-Time Traffic Signal Control System: Architecture, Algorithms, and Analysis. *Transportation Research Part C*, 9(6), pp 415–432.

Morgan, J.T. and J. D. C Little (1964). Synchronizing Traffic Signals for Maximal Bandwidth. *Operations Research*, Vol. 12, pp. 869-912.

National Traffic Signal Report Card (2007). Technical Report. National Transportation Operations Coalition.

National Traffic Signal Report Card (2012). Technical Report. National Transportation Operations Coalition.

Pengfei, T. L., Peter F., Stephen S. S., and Xiucheng Guo (2011). A Monte Carlo Simulation Procedure to Search for the Most-Likely Optimal Offsets on Arterial Using Cycle-by-Cycle Green Usage Report, (CD-ROM), Transportation Research Board, Washington, D.C.

Raus, J. (1975). Urban Traffic Control/Bus priority system (UTCS/BPS): A Status Report, Public Roads, Vol. 38, No. 4.

Rowe, E. (1991). The Los Angeles Automated Traffic Surveillance and Control (ATSAC) System. *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 1, pp. 16-20.

Sangkey Kim, Hajbabaie, A., Williams, B. M., Rouphail, N., M. (2013) Dynamic Bandwidth Analysis and Optimization for Coordinated Arterial Streets. Transportation Research Board 93th Annual Meeting.

Sharma, A., D.M. Bullock, and J. Bonneson (2006). Input-Output and Hybrid Techniques for Real-Time Prediction of Delay and Maximum Queue Length at a Signalized Intersection. In *Transportation Research Record*, No. 2035, Transportation Research Board of the National Academies, Washington, DC, pp. 25-33.

Sharma, A. and D.M. Bullock (2008). Field Evaluation of Alternative Real-Time Methods for Estimating Delay at Signalized Intersections. Proceedings of the 10th International Conference on Applications of Advanced Technologies in Transportation, Athens, Greece, May 27-31.

Shrank, D., B. Eisele, and T. Lomax,(2007). TTI's 2012 Urban Mobility Report. http://mobility.tamu.edu/ums/. Texas Transportation Institute.

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

Skabardonis, A. (1996). Determination of Timings in Signal Systems with Traffic-actuated Controllers. *Transportation Research Record*, No. 1554, 18-26.

Skabardonis, A. and N. Geroliminis (2008). Real-Time Monitoring and Control on Signalized Arterils. *Journal of Intelligent Transportation Systems*, Volume 12, Issue 2, pp 64-74.

Stamatiadis, C., and Gartner, N. H. (1996). MULTIBAND-96: A Program for Variable Bandwidth Progression Optimization of Multi-Arterial Traffic Networks. *Transportation Research Record*, No. 1554, pp. 9-17.

Stevanovic, A., C. Kergaye, and P.T. Martin (2009). SCOOT and SCATS: A Closer Look Into Their Operations. Transportation Research Board Annual Meeting, Paper No. 09-1672, Transportation Research Board of the National Academies, Washington, DC.

Stockfish, C.R.(1984). The UTCS Experience. Public Road, Vol. 48, No. 1.

The Urban Traffic Control System in Washington, DC. (1974). Federal Highway Administration, U.S. Department of Transportation, Washington, DC.

Tian, Z., Urbanik, T., Messer, C., Balke, K., and Koonce, P. (2003). A System Partition Approach to Improve Signal Timing, (CD-ROM), *Transportation Research Board*, Washington, D.C.

Traffic Signal Control System – Research and Innovation Technology Administration National Transportation Library.
http://ntl.bts.gov/lib/jpodocs/edldocs1/13480/ch3.pdf

Traffic Signal Retiming Practices in the United States (2010). NCHRP Report 409, Federal Highway Administration, Washington, DC.

Traffic Signal Timing Manual (2013). U.S. Department of Transportation FHWA.
http://ops.fhwa.dot.gov/publications/fhwahop08024/chapter9.htm#9.4

Transport Research Laboratory, (2014). How SCOOT Works. Retrieved on October 16, 2009.
http://www.scoot-utc.com/DetailedHowSCOOTWorks.php

Tsay, H. S., and Lin, L. T. (1998). A New Algorithm for Solving the Maximum Progression Bandwidth. *Transportation Research Record*, No. 1194, 15-30.

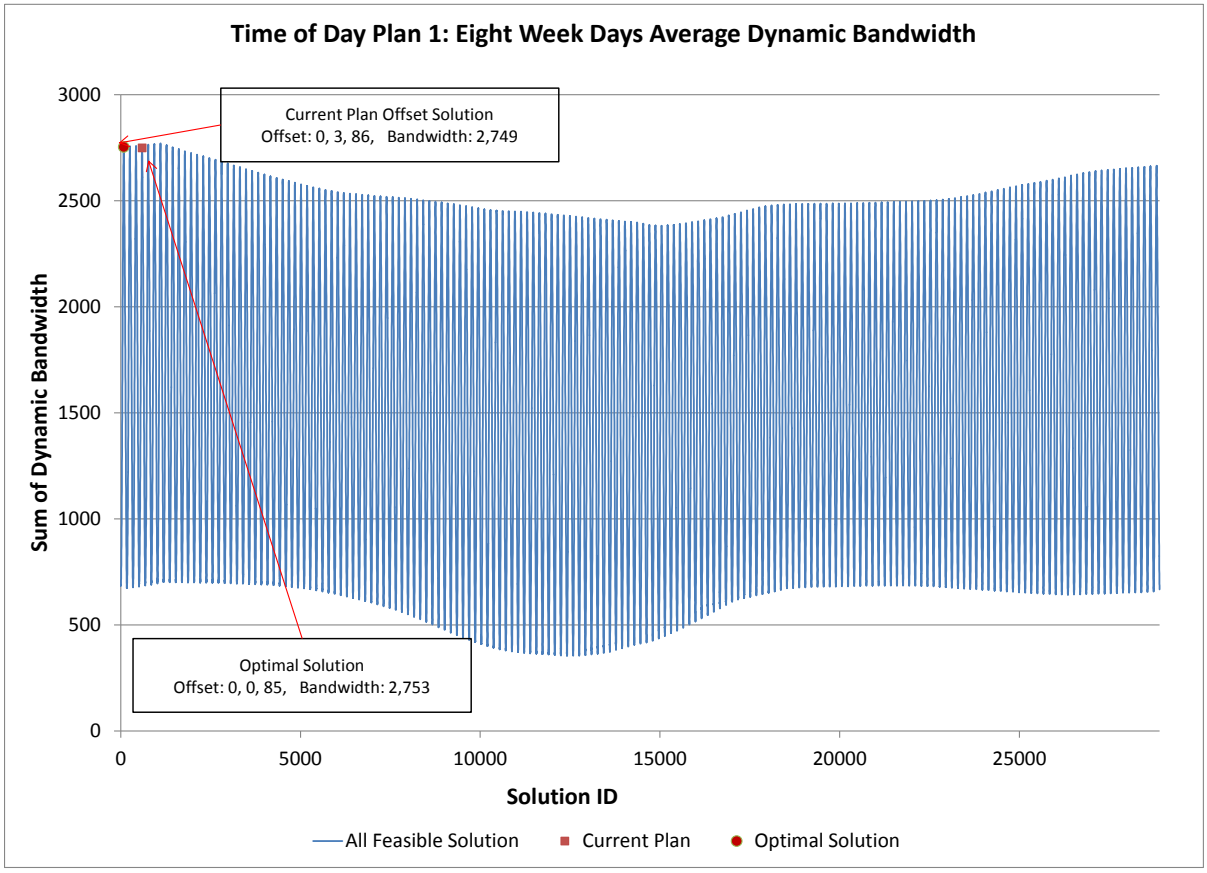
Yin, Y., Li, M., and Skabardonis, A. (2007). Offline offset Refiner for Coordinated Actuated Signal Control Systems1. *Journal of Transportation Engineering*, 133(7), pp 423-432.

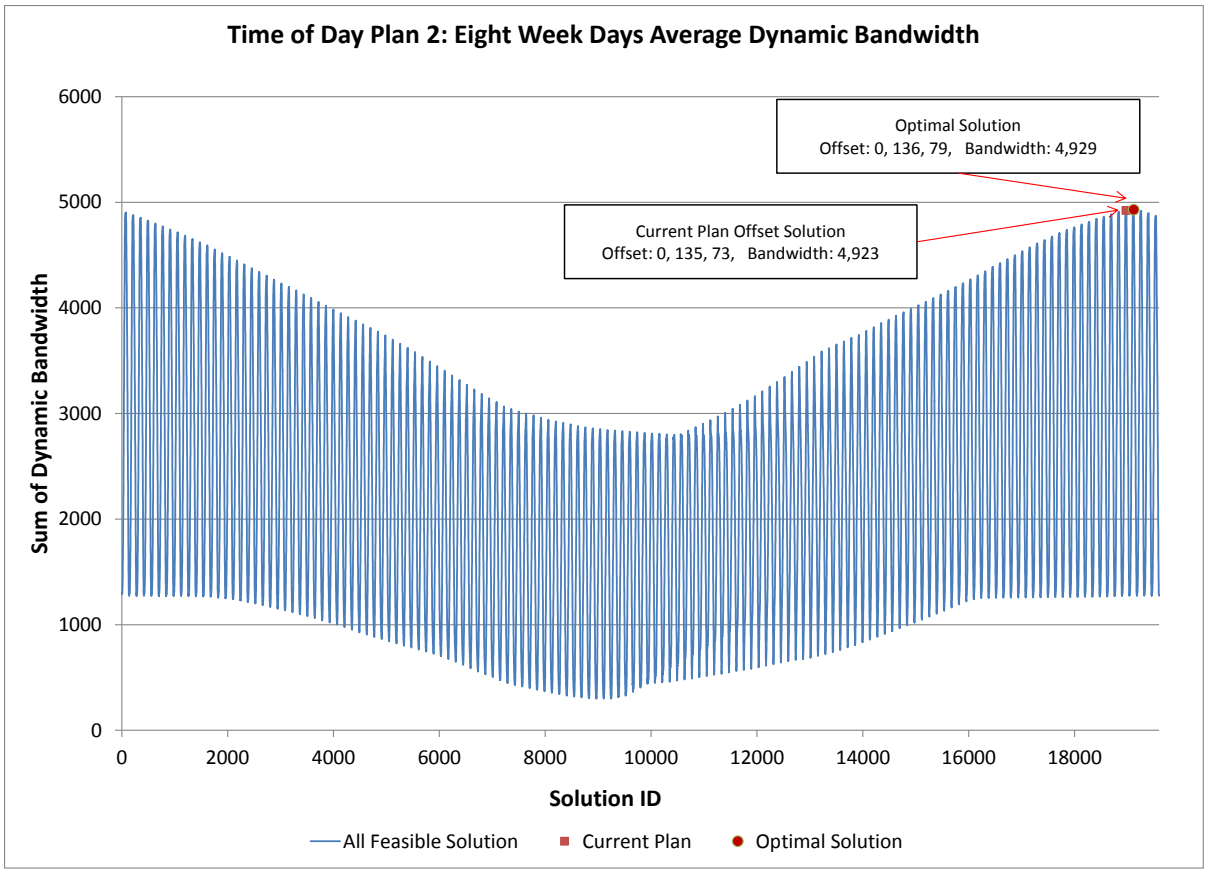
Zheng, J., X. Ma, Yao-Jan, and Y. Wang (2013). Measuring Signalized Intersection Performance in Real-Time with Traffic Sensors. *Journal of Intelligent Transportation Systems*, Volume 17, Issue 4, pp 304-306.

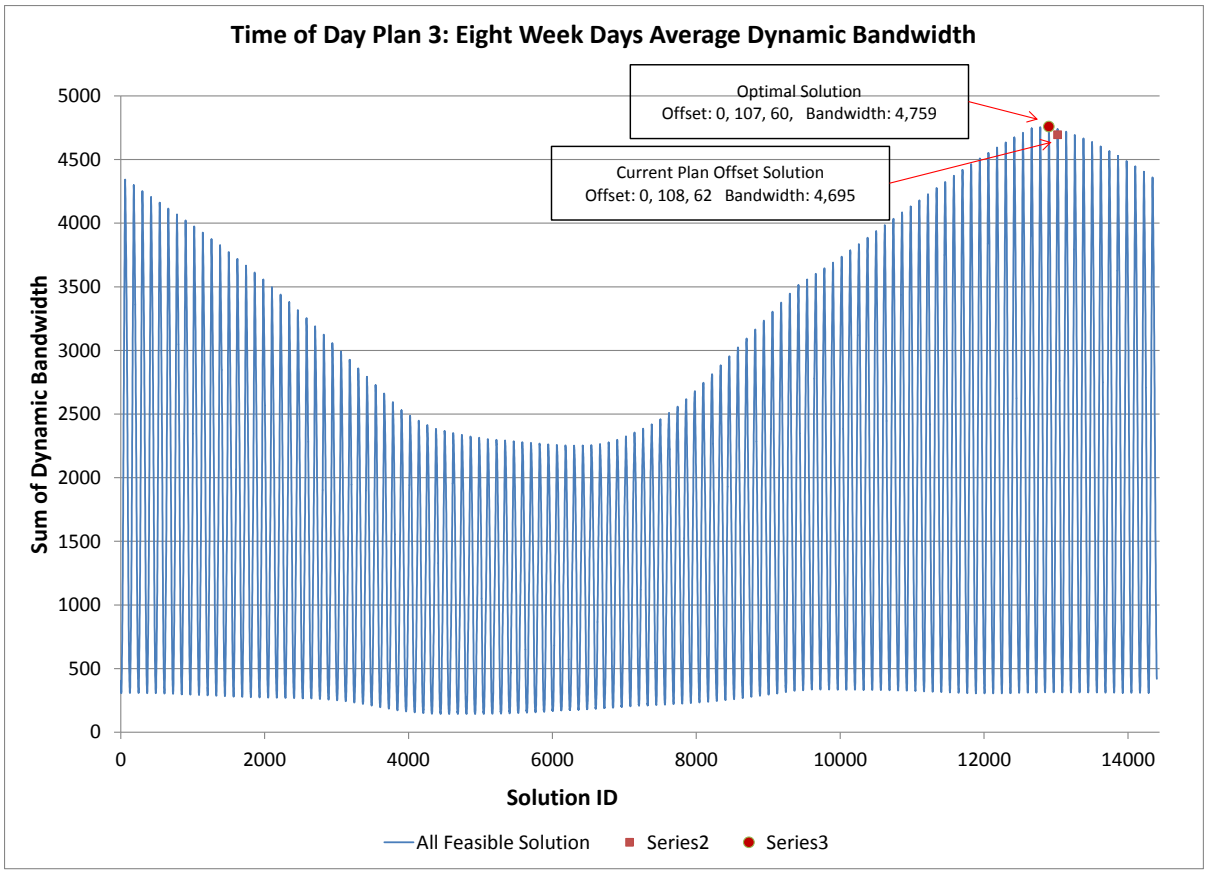
APPENDIX

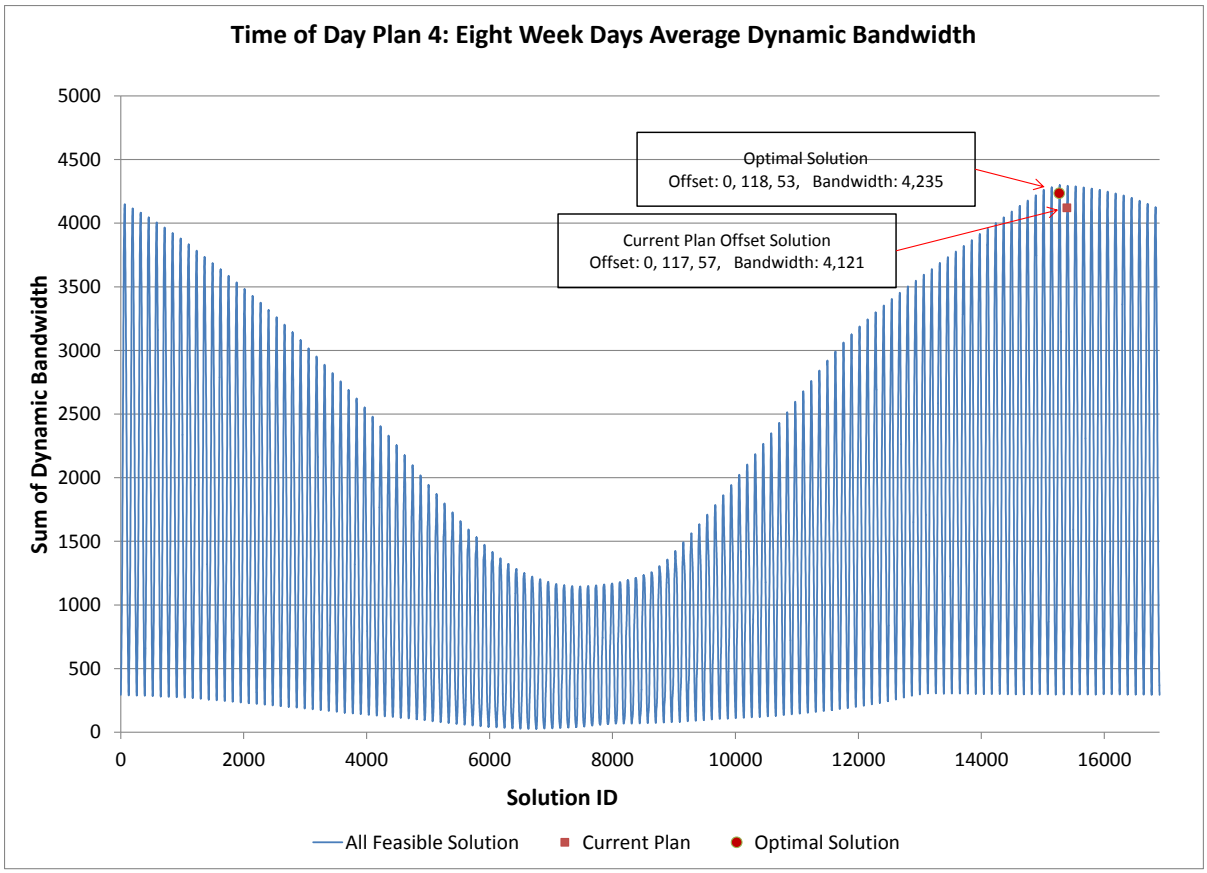
APPENDIX A: SOLUTION OF DYNAMIC BANDWIDTH OPTIMIZATION

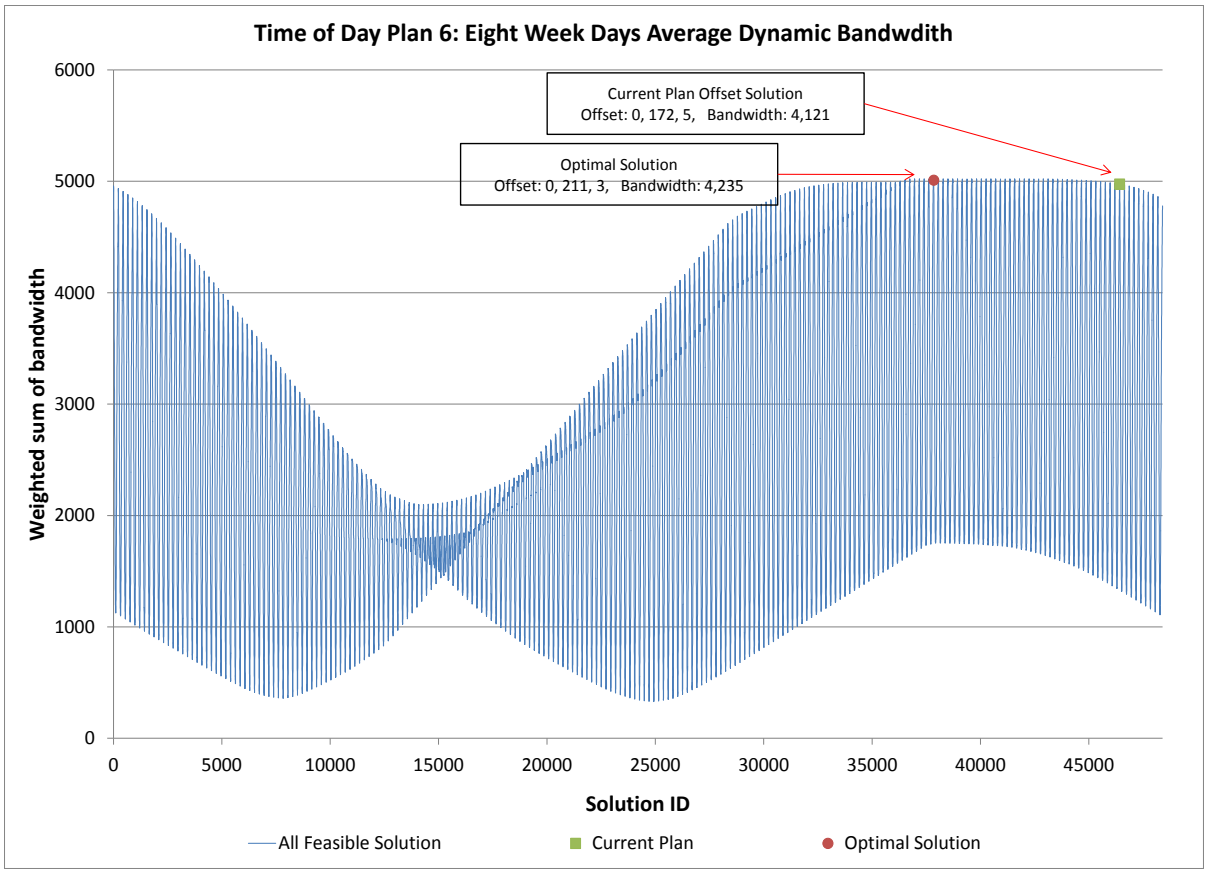
Appendix A provides current offset bandwidth solution with dynamic bandwidth optimization result on the all feasible solution plot for site "A" plan 1, 2, 3, 4 and 6 which correspond to Table 6.6 results.




Time of Day Plan 2: Eight Week Days Average Dynamic Bandwidth
Optimal Solution
Offset: 0, 136, 79, Bandwidth: 4,929
Current Plan Offset Solution
Offset: 0, 135, 73, Bandwidth: 4,923
Sum of Dynamic Bandwidth
6000
5000
4000
3000
2000
1000
0
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
Solution ID
All Feasible Solution
Current Plan
Optimal Solution


Time of Day Plan 3: Eight Week Days Average Dynamic Bandwidth
Optimal Solution
Offset: 0, 107, 60, Bandwidth: 4,759
Current Plan Offset Solution
Offset: 0, 108, 62 Bandwidth: 4,695
Sum of Dynamic Bandwidth
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
14000
Solution ID
All Feasible Solution
Series2
Series3


Time of Day Plan 4: Eight Week Days Average Dynamic Bandwidth
Optimal Solution
Offset: 0, 118, 53, Bandwidth: 4,235
Current Plan Offset Solution
Offset: 0, 117, 57, Bandwidth: 4,121
Sum of Dynamic Bandwidth
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2000
4000
6000
8000
10000
12000
14000
16000
Solution ID
All Feasible Solution
Current Plan
Optimal Solution


Time of Day Plan 6: Eight Week Days Average Dynamic Bandwdith
Current Plan Offset Solution
Offset: 0, 172, 5, Bandwidth: 4,121
Optimal Solution
Offset: 0, 211, 3, Bandwidth: 4,235
Weighted sum of bandwidth
6000
5000
4000
3000
2000
1000
0
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
Solution ID
All Feasible Solution
Current Plan
Optimal Solution

APPENDIX B: DYNAMIC BANDWIDTH ANALYSIS TOOL (DBAT) C++ CODE

// TrafficSimulationDlg.cpp : implementation file

```
#include "stdafx.h"
#include "TrafficSimulation.h"
#include "TrafficSimulationDlg.h"
#include "afxdialogex.h"
#include "BasicExcel.hpp"

#define MAX_INTER_COLUMN 6
#define MAX_NULL_SPACE_COLUMN 1
#define MAX_NULL_SPACE_ROW 2

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

using namespace YExcel;
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialogEx
{
public:
	CAboutDlg();

// Dialog Data
	enum { IDD = IDD_ABOUTBOX };

	protected:
	virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support

// Implementation
protected:
	DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(CAboutDlg::IDD)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
```

```
END_MESSAGE_MAP()


// CTrafficSimulationDlg dialog

CTrafficSimulationDlg::CTrafficSimulationDlg(CWnd* pParent /*=NULL*/)
	: CDialogEx(CTrafficSimulationDlg::IDD, pParent)
{
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CTrafficSimulationDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialogEx::DoDataExchange(pDX);
	DDX_Control(pDX, IDC_EDIT_FILE, m_filepath);
	DDX_Control(pDX, IDC_EDIT_SHEET, m_excelsheet);
	DDX_Control(pDX, IDC_EDIT_NUM_INTERSECTION, m_intersectionNum);
	DDX_Control(pDX, IDC_EDIT_SPEED, m_carspeed);
	DDX_Control(pDX, IDC_EDIT_SECOND_RANGE, m_secRange);
	DDX_Control(pDX, IDC_EDIT_STATUS, m_status_display);
	DDX_Control(pDX, IDC_CHECK_ENABLE_WRITE, m_enableWriting);
	DDX_Control(pDX, IDC_EDIT_RANGE_END, m_editRangeEnd);
	DDX_Control(pDX, IDC_EDIT_RANGE_INTERSECTION,
m_editRangeIntersection);
	DDX_Control(pDX, IDC_EDIT_RANGE_START, m_editRangeStart);
	DDX_Control(pDX, IDC_LIST_RANGE, m_listRange);
	DDX_Control(pDX, IDC_EDIT_FILTER, m_filter);
	DDX_Control(pDX, IDC_EDIT_STEP, m_step);
	DDX_Control(pDX, IDC_EDIT_FACTOR1, m_factor1);
	DDX_Control(pDX, IDC_EDIT_FACTOR2, m_factor2);
	DDX_Control(pDX, IDC_EDIT_FILTER2, m_filter2);
}

BEGIN_MESSAGE_MAP(CTrafficSimulationDlg, CDialogEx)
	ON_WM_SYSCOMMAND()
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
	ON_BN_CLICKED(IDC_BUTTON_START,
&CTrafficSimulationDlg::OnBnClickedButtonStart)
	ON_BN_CLICKED(IDC_BUTTON_SEARCH,
&CTrafficSimulationDlg::OnBnClickedButtonSearch)
	ON_BN_CLICKED(IDC_BUTTON_STOP,
&CTrafficSimulationDlg::OnBnClickedButtonStop)
	ON_BN_CLICKED(IDC_BUTTON_ADD,
&CTrafficSimulationDlg::OnBnClickedButtonAdd)
	ON_BN_CLICKED(IDC_BUTTON_CLEAR,
&CTrafficSimulationDlg::OnBnClickedButtonClear)
```

```
		ON_EN_CHANGE(IDC_EDIT_FACTOR1,
&CTrafficSimulationDlg::OnChangeEditFactor1)
	END_MESSAGE_MAP()


	// CTrafficSimulationDlg message handlers

	BOOL CTrafficSimulationDlg::OnInitDialog()
	{
		CDialogEx::OnInitDialog();

		// Add "About..." menu item to system menu.

		// IDM_ABOUTBOX must be in the system command range.
		ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
		ASSERT(IDM_ABOUTBOX < 0xF000);

		CMenu* pSysMenu = GetSystemMenu(FALSE);
		if (pSysMenu != NULL)
		{
			BOOL bNameValid;
			CString strAboutMenu;
			bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
			ASSERT(bNameValid);
			if (!strAboutMenu.IsEmpty())
			{
		pSysMenu->AppendMenu(MF_SEPARATOR);
		pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
			}
		}

		// Set the icon for this dialog.  The framework does this automatically
		//  when the application's main window is not a dialog
		SetIcon(m_hIcon, TRUE);		// Set big icon
		SetIcon(m_hIcon, FALSE);		// Set small icon

		// TODO: Add extra initialization here

		m_filepath.SetWindowTextA("input_data_form3_.xls");
		m_excelsheet.SetWindowTextA("Data arrange");
		m_intersectionNum.SetWindowTextA("3");//7
		m_carspeed.SetWindowTextA("66"); // feet/sec,
		m_secRange.SetWindowTextA("4"); //159
		m_filter.SetWindowTextA("0.0"); // filter
		m_filter2.SetWindowTextA("0.0"); // filter0
		m_step.SetWindowTextA("1"); // searching step
		m_factor1.SetWindowTextA("1"); // factor
```

```
		m_bStop = FALSE;
		m_nDisplayIndex =1;
		return TRUE;  // return TRUE  unless you set the focus to a control
	}

	void CTrafficSimulationDlg::OnSysCommand(UINT nID, LPARAM lParam)
	{
		if ((nID & 0xFFF0) == IDM_ABOUTBOX)
		{
			CAboutDlg dlgAbout;
			dlgAbout.DoModal();
		}
		else
		{
			CDialogEx::OnSysCommand(nID, lParam);
		}
	}

	// If you add a minimize button to your dialog, you will need the code below
	//  to draw the icon.  For MFC applications using the document/view model,
	//  this is automatically done for you by the framework.

	void CTrafficSimulationDlg::OnPaint()
	{
		if (IsIconic())
		{
			CPaintDC dc(this); // device context for painting
			SendMessage(WM_ICONERASEBKGND,
reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

			// Center icon in client rectangle
			int cxIcon = GetSystemMetrics(SM_CXICON);
			int cyIcon = GetSystemMetrics(SM_CYICON);
			CRect rect;
			GetClientRect(&rect);
			int x = (rect.Width() - cxIcon + 1) / 2;
			int y = (rect.Height() - cyIcon + 1) / 2;

			// Draw the icon
			dc.DrawIcon(x, y, m_hIcon);
		}
		else
		{
			CDialogEx::OnPaint();
		}
	}
```

```
// The system calls this function to obtain the cursor to display while the user drags
//  the minimized window.
HCURSOR CTrafficSimulationDlg::OnQueryDragIcon()
{
	return static_cast<HCURSOR>(m_hIcon);
}

void CTrafficSimulationDlg::OnBnClickedButtonStart()
{
	CString strTemp, strDisplay;
	CTime time;
	int i;
	int nTemp1, nTemp2, nTemp3;
	// TODO: Add your control notification handler code here
	UpdateData(TRUE);

	m_filepath.GetWindowTextA(m_strFilePath);
	m_excelsheet.GetWindowTextA(m_strExcelsheet);
	m_carspeed.GetWindowTextA(strTemp);
	m_dblcarSpeed = atof(strTemp)*5280.0/3600.0;
	m_intersectionNum.GetWindowTextA(strTemp);
	m_nIntersectionNum = atoi(strTemp);
	m_filter.GetWindowTextA(strTemp);
	m_dblFilter = atof(strTemp);
	m_filter2.GetWindowTextA(strTemp);
	m_dblFilter2 = atof(strTemp);
	m_step.GetWindowTextA(strTemp);
	m_nStep = atoi(strTemp);
	m_secRange.GetWindowTextA(strTemp);
	m_nSecondRange = atoi(strTemp);
	m_factor1.GetWindowTextA(strTemp);
	m_dblWeightedFactor_forward = atof(strTemp);
	m_factor2.GetWindowTextA(strTemp);
	m_dblWeightedFactor_backward = atof(strTemp);

	if(m_dblWeightedFactor_forward + m_dblWeightedFactor_backward != 1)
	{
		AfxMessageBox("Wrong Input: weighted factor");
		return;
	}

	for(i=0; i<m_nIntersectionNum; i++)
	{
		m_nRangeStart[i] = 0;
		m_nRangeEnd[i] = m_nSecondRange;
	}
```

```
		for(i=0; i<m_listRange.GetCount(); i++)
		{
			m_listRange.GetText(i, strTemp);
			sscanf(strTemp, "%d, %d, %d", &nTemp1, &nTemp2, &nTemp3);

			if(nTemp1 > 1 && nTemp2 >= 0)
			{
		m_nRangeStart[nTemp1-2] = nTemp2;
		m_nRangeEnd[nTemp1-2] = nTemp3+1;
			}
			else
			{
		AfxMessageBox("Out of range: " +strTemp);
		return;
			}
		}

		m_nEnableWriting = m_enableWriting.GetCheck ();
		m_destFile.Open("traffic.txt", CFile::modeWrite | CFile::modeCreate |
CFile::typeText | CFile::shareDenyNone);
		m_destFile2.Open("simulation_time.txt", CFile::modeWrite | CFile::modeCreate |
CFile::typeText | CFile::shareDenyNone);
		time = CTime::GetCurrentTime();
		strDisplay = time.Format("Starting: %Y-%m-%d %H:%M:%S \n");
		m_status_display.SetWindowTextA(strDisplay);
		UpdateData(FALSE);
		LoadExcel();
		CreateTimeLine();
		m_nDisplayIndex =1;
		Choose_Interval(0,0);
		time = CTime::GetCurrentTime();
		strDisplay += time.Format("Ending: %Y-%m-%d %H:%M:%S\n");
		m_destFile2.WriteString (strDisplay);
		m_destFile.Close();
		m_destFile2.Close();
		m_status_display.SetWindowTextA(strDisplay);
		UpdateData(FALSE);

	}

	void CTrafficSimulationDlg::CreateTimeLine()
	{
		int i;
		for(int i=0; i<MAX_ARRAY; i++)
		{
			arrayTimeLine_forward[i].RemoveAll();
			arrayTimeLine_backward[i].RemoveAll();
```

```
}

int nSection =0;
int nIndex=0;
int nSub=0;
int nStartTime =0;
int nEndTime =0;
int nGreenTime = 0;
int nExtraTime=0;
for(nSection=0; nSection<m_nIntersectionNum; nSection++) // intersection
{
	for(nIndex =0; nIndex< arrayTime[nSection].GetCount(); nIndex++) // items
of each section
	{
//forward
nStartTime = arrayTime[nSection].GetAt(nIndex);
nGreenTime = arrayUsedPhase2[nSection].GetAt(nIndex);
nExtraTime = arrayExtraTimeCP1[nSection].GetAt(nIndex);
if(nIndex >= arrayTime[nSection].GetCount()-1)
	nEndTime = nStartTime+nGreenTime;
else
	nEndTime = arrayTime[nSection].GetAt(nIndex+1);

for(nSub=0; nSub<nEndTime - nStartTime; nSub++)
{
	if(nSub >= nEndTime - nStartTime - nExtraTime)
arrayTimeLine_forward[nSection].Add(TRUE);
	else
	{
if(nSub < nGreenTime)
	arrayTimeLine_forward[nSection].Add(TRUE);
else
	arrayTimeLine_forward[nSection].Add(FALSE);
	}
}

//backward
nStartTime = arrayTime[nSection].GetAt(nIndex);
nGreenTime = arrayUsedPhase6[nSection].GetAt(nIndex); //
nExtraTime = arrayExtraTimeCP2[nSection].GetAt(nIndex);
if(nIndex >= arrayTime[nSection].GetCount()-1)
	nEndTime = nStartTime+nGreenTime;
else
	nEndTime = arrayTime[nSection].GetAt(nIndex+1);

for(nSub=0; nSub<nEndTime - nStartTime; nSub++)
{
```

```
                if(nSub >= nEndTime - nStartTime - nExtraTime)
        arrayTimeLine_backward[nSection].Add(TRUE);
                else
                {
        if(nSub < nGreenTime)
                arrayTimeLine_backward[nSection].Add(TRUE);
        else
                arrayTimeLine_backward[nSection].Add(FALSE);
                }

        }
                }
        }
}

BOOL CTrafficSimulationDlg::Choose_Interval(int nInterSection, int nSectionInterval)
{
        int i;
        int nPredictTime=0;
        int nTimeLineIndex=0;
        CString strDisplay;
        BOOL bfail = FALSE;
        int nCount =0;

        if(nInterSection != -1)
                m_nSectionInterval[nInterSection] = nSectionInterval;

        if(nInterSection+1 >= m_nIntersectionNum)
        {
                m_arrayIndividual_forward.RemoveAll();
                m_arrayIndividual_backward.RemoveAll();
                // calling
                m_nBandwidth1 =0;
                m_nBandwidth2 =0;
                m_nBandwidth_count1 =0;
                m_nBandwidth_count2 =0;
                m_nCurGroupCheck = 1;
                m_nPreGroupCheck = 0;
                nCount = arrayTimeLine_forward[0].GetCount();
                for(i=0; i<nCount; i++)
                {
        if(arrayTimeLine_forward[0].GetAt(i) == TRUE)
        {
                Analysis_Intersection_forward(0,i+m_nSectionInterval[0]);
        }
        else
                m_nCurGroupCheck++;
```

```
                    }
                    AddData(&m_arrayIndividual_forward, m_nBandwidth1);
                    MakeArray_Diff(&m_arrayIndividual_forward, &m_arrayDiff_forward, 0);

                    m_nCurGroupCheck = 1;
                    m_nPreGroupCheck = 0;

                    nCount = arrayTimeLine_backward[m_nIntersectionNum-1].GetCount();
                    for(i=0; i<nCount; i++)
                    {
            if(arrayTimeLine_backward[m_nIntersectionNum-1].GetAt(i) == TRUE)
            {

        Analysis_Intersection_backward(0,i+m_nSectionInterval[m_nIntersectionNum-1]);
            }
            else
                    m_nCurGroupCheck++;
                    }
                    AddData(&m_arrayIndividual_backward, m_nBandwidth2);
                    MakeArray_Diff(&m_arrayIndividual_backward, &m_arrayDiff_backward,
1);
                    DisplayStatus(nInterSection+1);

                    return FALSE;
            }

            for(i=m_nRangeStart[nInterSection]; i<m_nRangeEnd[nInterSection]; i+=m_nStep)
            {
                    Choose_Interval(nInterSection+1, i);
            }

            return TRUE;
        }



        BOOL CTrafficSimulationDlg::Analysis_Intersection_forward(int nInterSection, int
nCarPreSecTime)
        {
            int i;
            int nPredictTime=0;
            int nTimeLineIndex=-1;
            CString strDisplay;
            BOOL bfail = FALSE;
            int nCount =0;
```

```
		if(nInterSection !=0)
		{
			nPredictTime = (int)(arrayOffset[nInterSection].GetAt(0)/m_dblcarSpeed +
0.9999999) + nCarPreSecTime;
			nTimeLineIndex = nPredictTime - m_nSectionInterval[nInterSection];
		}
		else
		{
			nPredictTime = 0 + nCarPreSecTime;
			nTimeLineIndex = nPredictTime - m_nSectionInterval[nInterSection];
		}

		nCount = arrayTimeLine_forward[nInterSection].GetCount();
		if(nTimeLineIndex >= 0 && nTimeLineIndex < nCount)
		{
			if(arrayTimeLine_forward[nInterSection].GetAt(nTimeLineIndex) ==
FALSE)
			{
		bfail = TRUE;
			}
		}
		else
			bfail = TRUE;

		if(bfail == TRUE)
		{
			// stop while working
			//DisplayStatus(nInterSection+1);
			m_nCurGroupCheck++;
			return	FALSE;
		}

		 //testing
		/*
		if(m_nSectionInterval[0] == 0 && m_nSectionInterval[1] == 10 &&
m_nSectionInterval[2] == 20 &&
			m_nSectionInterval[3] == 30)
		{
			if(nInterSection == 0) //m_nIntersectionNum-1)
		int test = 3;
		}*/

		if(nInterSection+1 >= m_nIntersectionNum)
		{
			// completing
			m_nBandwidth1++;
```

```
                    if(m_nCurGroupCheck != m_nPreGroupCheck)
                    {
                AddData(&m_arrayIndividual_forward, m_nBandwidth1-1);

                m_nBandwidth_count1++;
                m_nPreGroupCheck = m_nCurGroupCheck;
                    }
                    return TRUE;
                }


                Analysis_Intersection_forward(nInterSection+1, nPredictTime);
                return TRUE;
        }

        BOOL CTrafficSimulationDlg::Analysis_Intersection_backward(int nInterSection, int
nCarPreSecTime)
        {
                int i;
                int nPredictTime=0;
                int nTimeLineIndex=0;
                CString strDisplay;
                BOOL bfail = FALSE;

                if(nInterSection !=0)
                {
                        nPredictTime = (int)(arrayOffset[m_nIntersectionNum-nInterSection-
1].GetAt(1)/m_dblcarSpeed + 0.9999999) + nCarPreSecTime;
                        nTimeLineIndex = nPredictTime -
m_nSectionInterval[m_nIntersectionNum-nInterSection-1];
                }
                else
                {
                        nPredictTime = 0 + nCarPreSecTime;
                        nTimeLineIndex = nPredictTime -
m_nSectionInterval[m_nIntersectionNum-nInterSection-1];
                }
                /* //testing
                if(m_nSectionInterval[0] == 0 && m_nSectionInterval[1] == 20 &&
m_nSectionInterval[2] == 40 &&
                        m_nSectionInterval[3] == 60)
                {
                        if(nInterSection ==0 && nCarPreSecTime == 0)
                int test = 3;
                }*/
```

```
		if(nTimeLineIndex >= 0 && nTimeLineIndex <
arrayTimeLine_backward[m_nIntersectionNum-nInterSection-1].GetCount())
		{
			if(arrayTimeLine_backward[m_nIntersectionNum-nInterSection-
1].GetAt(nTimeLineIndex) == FALSE)
			{
		bfail = TRUE;
			}
		}
		else
			bfail = TRUE;

		if(bfail == TRUE)
		{
			// stop while working
			//DisplayStatus(nInterSection+1,0);
			m_nCurGroupCheck++;
			return	FALSE;
		}



		if(nInterSection+1 >= m_nIntersectionNum)
		{
			// completing
			m_nBandwidth2++;

			if(m_nCurGroupCheck != m_nPreGroupCheck)
			{
		AddData(&m_arrayIndividual_backward, m_nBandwidth2-1);

		m_nBandwidth_count2++;
		m_nPreGroupCheck = m_nCurGroupCheck;
			}
			return TRUE;
		}

		Analysis_Intersection_backward(nInterSection+1, nPredictTime);
		return TRUE;
	}

	void CTrafficSimulationDlg::DisplayStatus(int nInterSection)
	{
		int i,j;
		int nBandW;
		CString strTemp="";
```

```
		CString strDisplay= "";
		int nSum1, nSum2;
		nSum1= GetSum(&m_arrayDiff_forward);
		nSum2 = GetSum(&m_arrayDiff_backward);
		strDisplay.Format("ID: %-5d: ", m_nDisplayIndex++);

		for(int i=0; i<nInterSection; i++)
		{
			strTemp.Format("%3d", m_nSectionInterval[i]);

			if(i< nInterSection-1)
		strTemp += " - ";

			strDisplay += strTemp;
		}

		strTemp.Format(": %5d,  %5d, %5d, %5d, %5.2f, %5.2f %3.4f, %3.4f, %3.4f, %3.4f,
 %d, %3.4f, %0.4f, %0.4f",
			nSum1, nSum2,
			m_arrayDiff_forward.GetCount(), m_arrayDiff_backward.GetCount(),
			GetMean(&m_arrayDiff_forward), GetMean(&m_arrayDiff_backward),
			(nSum1*100.0)/(m_nSecondRange*m_nExcelRow*1.0),
(nSum2*100.0)/(m_nSecondRange*m_nExcelRow*1.0),
			GetStandardDeviation(&m_arrayDiff_forward),
GetStandardDeviation(&m_arrayDiff_backward),
			nSum1 + nSum2,
			(nSum1*m_dblWeightedFactor_forward+
nSum2*m_dblWeightedFactor_backward),
			m_dblWeightedFactor_forward, m_dblWeightedFactor_backward
			);
		strDisplay += strTemp;

		if(m_nEnableWriting == 1)
		{
			strDisplay += " ,!! ";
			//individual data forward
			for(i=0;i<m_arrayDiff_forward.GetCount(); i++)
			{
		nBandW = m_arrayDiff_forward.GetAt(i);
		strTemp.Format(",%d", nBandW);
		strDisplay += strTemp;
			}

			strDisplay += " ,!! ";
			//individual data backward
			for(i=0;i<m_arrayDiff_backward.GetCount(); i++)
			{
```

```
                nBandW = m_arrayDiff_backward.GetAt(i);
                strTemp.Format(",%d", nBandW);
                strDisplay += strTemp;
                        }
                }

                strDisplay += "\n";
                m_destFile.WriteString (strDisplay);
                m_nNumDisplay++;
        }


        BOOL CTrafficSimulationDlg::LoadExcel()
        {
                CString strTemp;
                double dblTemp;
                int nTemp;
                int i;
                int nCheckRow =0;
                int nRow, nCol, nIntersection;
                BasicExcel e;
                m_nExcelRow = 0;
                for(int i=0; i<MAX_ARRAY; i++)
                {
                        arrayTime[i].RemoveAll();
                        arrayOffset[i].RemoveAll();
                        arrayExtraTimeCP1[i].RemoveAll();
                        arrayExtraTimeCP2[i].RemoveAll();
                        arrayUsedPhase2[i].RemoveAll();
                        arrayUsedPhase6[i].RemoveAll();
                }

                // Load a workbook with one sheet, display its contents and save into another file.
                e.Load(m_strFilePath);
                BasicExcelWorksheet* sheet1 = e.GetWorksheet(m_strExcelsheet);
                if (sheet1)
                {
                        size_t maxRows = sheet1->GetTotalRows();
                        size_t maxCols = sheet1->GetTotalCols();
                        cout << "Dimension of " << sheet1->GetAnsiSheetName() << " (" <<
maxRows << ", " << maxCols << ")" << endl;

                        m_nExcelRow = maxRows;

                        for(nIntersection = 0; nIntersection<m_nIntersectionNum; nIntersection++)
                        {
                for(nCol = 0; nCol<MAX_INTER_COLUMN; nCol++)
```

```
		{
			for(nRow = 0; nRow<maxRows; nRow++)
			{
		nCheckRow = 0;
		BasicExcelCell* cell = sheet1->Cell
			(nRow+MAX_NULL_SPACE_ROW,
nIntersection*(MAX_INTER_COLUMN+MAX_NULL_SPACE_COLUMN)+nCol);
		switch (cell->Type())
		{
		case BasicExcelCell::UNDEFINED:
			break;
		case BasicExcelCell::INT:
			nTemp = cell->GetInteger();
			nCheckRow =1;
			break;
		case BasicExcelCell::DOUBLE:
			dblTemp = cell->GetDouble();
			nTemp = dblTemp;
			nCheckRow = 1;
			break;
		case BasicExcelCell::STRING:
			strTemp.Format("%10s", cell->GetString());
			break;
		case BasicExcelCell::WSTRING:
			strTemp.Format("%10s", cell->GetWString());
			break;

		}

		switch(nCol)
		{
		case 0: // Time
			if(nCheckRow >0)
		arrayTime[nIntersection].Add((int)(dblTemp*24*60*60+0.5));
			else
			{
		maxRows = nRow;
		m_nExcelRow = nRow;
			}
			break;
		case 1: // Offset
			arrayOffset[nIntersection].Add(nTemp);
			break;
		case 2: // ExtraTimeCP1
			arrayExtraTimeCP1[nIntersection].Add(nTemp);
			break;
		case 3: // ExtraTimeCP2
```

```
            arrayExtraTimeCP2[nIntersection].Add(nTemp);
            break;
case 4: // UsedPhase2
            arrayUsedPhase2[nIntersection].Add(nTemp);
            break;
case 5: // UsedPhase6
            arrayUsedPhase6[nIntersection].Add(nTemp);
            break;

}
            }
}
            }
}


/*
            for (size_t r=0; r<maxRows; ++r)
            {
printf("%10d", r+1);
for (size_t c=0; c<maxCols; ++c)
{
            BasicExcelCell* cell = sheet1->Cell(r,c);
            switch (cell->Type())
            {
case BasicExcelCell::UNDEFINED:
            printf("          ");
            break;

case BasicExcelCell::INT:
            printf("%10d", cell->GetInteger());
            break;

case BasicExcelCell::DOUBLE:
            printf("%10.6lf", cell->GetDouble());
            break;

case BasicExcelCell::STRING:
            printf("%10s", cell->GetString());
            break;

case BasicExcelCell::WSTRING:
            wprintf(L"%10s", cell->GetWString());
            break;
            }
}
cout << endl;
```

```
        }
}
cout << endl;
e.SaveAs("example2.xls");

// Create a new workbook with 2 worksheets and write some contents.
e.New(2);
e.RenameWorksheet("Sheet1", "Test1");
BasicExcelWorksheet* sheet = e.GetWorksheet("Test1");
BasicExcelCell* cell;
if (sheet)
{
        for (size_t c=0; c<4; ++c)
        {
cell = sheet->Cell(0,c);
cell->Set((int)c);
        }

        cell = sheet->Cell(1,3);
        cell->SetDouble(3.141592654);
        sheet->Cell(1,4)->SetString("Test str1");
        sheet->Cell(2,0)->SetString("Test str2");
        sheet->Cell(2,5)->SetString("Test str1");
        sheet->Cell(4,0)->SetDouble(1.1);
        sheet->Cell(4,1)->SetDouble(2.2);
        sheet->Cell(4,2)->SetDouble(3.3);
        sheet->Cell(4,3)->SetDouble(4.4);
        sheet->Cell(4,4)->SetDouble(5.5);
        sheet->Cell(4,4)->EraseContents();
}

sheet = e.AddWorksheet("Test2", 1);
sheet = e.GetWorksheet(1);
if (sheet)
{
        sheet->Cell(1,1)->SetDouble(1.1);
        sheet->Cell(2,2)->SetDouble(2.2);
        sheet->Cell(3,3)->SetDouble(3.3);
        sheet->Cell(4,4)->SetDouble(4.4);
        sheet->Cell(70,2)->SetDouble(5.5);
}
e.SaveAs("example3.xls");

// Load the newly created sheet and display its contents
e.Load("example3.xls");

size_t maxSheets = e.GetTotalWorkSheets();
```

```
		cout << "Total number of worksheets: " << e.GetTotalWorkSheets() << endl;
		for (size_t i=0; i<maxSheets; ++i)
		{
			BasicExcelWorksheet* sheet = e.GetWorksheet(i);
			if (sheet)
			{
		size_t maxRows = sheet->GetTotalRows();
		size_t maxCols = sheet->GetTotalCols();
		cout << "Dimension of " << sheet->GetAnsiSheetName() << " (" << maxRows << ",
" << maxCols << ")" << endl;

		if (maxRows>0)
		{
			printf("          ");
			for (size_t c=0; c<maxCols; ++c) printf("%10d", c+1);
			cout << endl;
		}

		for (size_t r=0; r<maxRows; ++r)
		{
			printf("%10d", r+1);
			for (size_t c=0; c<maxCols; ++c)
			{
		cout << setw(10) << *(sheet->Cell(r,c)); // Another way of printing a cell content.

			}
			cout << endl;
		}
		if (i==0)
		{
			ofstream f("example4.csv");
			sheet->Print(f, ',', '\"');    // Save the first sheet as a CSV file.
			f.close();
		}
			}
			cout << endl;
		}
		*/
		return TRUE;
	}



	void CTrafficSimulationDlg::OnBnClickedButtonSearch()
	{
		// TODO: Add your control notification handler code here
		char pbuf[1024];
```

```
		int nStart = 0 , nstrCount = 0;
		int nCount = 0  , nDataLen = 0 , nRS = 0;
		CString szBuf = _T("") , szStr = _T("");
		CFile cfile;
		char szFilter[] = "Excel file(*.xls)|*.xls||";
		CString strFilePath;
		CFileDialog fileDlg(TRUE, NULL, NULL,
			OFN_HIDEREADONLY, szFilter);
		if(IDOK == fileDlg.DoModal())
		{
			m_strFilePath = fileDlg.GetPathName();
			m_filepath.SetWindowTextA (m_strFilePath);
		}
		else return;
	}

	void CTrafficSimulationDlg::OnBnClickedButtonStop()
	{
		// TODO: Add your control notification handler code here
		m_bStop = TRUE;
	}

	void CTrafficSimulationDlg::MakeArray_Diff(CArray<int, int>* pSrcArray, CArray<int,
int>* pDesArray, int nDirection)
	{
		int i=0;
		double sum = 0;
		int nValue =0;
		double dblFilter =0;
		if(nDirection == 0)
			dblFilter = m_dblFilter;
		else
			dblFilter = m_dblFilter2;

		pDesArray->RemoveAll();

	   for(i = 0; i < pSrcArray->GetCount()-1; i++)
		{
			nValue = pSrcArray->GetAt(i+1) - pSrcArray->GetAt(i);
			if(dblFilter < nValue)
		pDesArray->Add(nValue);
		}

	}

	double CTrafficSimulationDlg::GetMean(CArray<int, int>* pArray)
	{
```

```
	int i=0;
	double sum = 0;

   for(i = 0; i < pArray->GetCount(); i++)
		sum += pArray->GetAt(i);

   return sum/pArray->GetCount();

}

double CTrafficSimulationDlg::GetSum(CArray<int, int>* pArray)
{
	int i=0;
	double sum = 0;

   for(i = 0; i < pArray->GetCount(); i++)
		sum += pArray->GetAt(i);

   return sum;

}

double CTrafficSimulationDlg::GetStandardDeviation(CArray<int, int>* pArray)
{
   double temp = 0;
	int i=0;

	double mean =0;

	mean = GetMean(pArray);

   for(i = 0; i < pArray->GetCount(); i++)
   {
		temp += (pArray->GetAt(i) - mean) * (pArray->GetAt(i) - mean) ;
   }

   return sqrt(temp / (pArray->GetCount()));

}



void CTrafficSimulationDlg::OnBnClickedButtonAdd()
{
	// TODO: Add your control notification handler code here
	CString strData;
	CString strTemp1, strTemp2, strTemp3;
```

```
	m_editRangeIntersection.GetWindowTextA(strTemp1);
	m_editRangeStart.GetWindowTextA(strTemp2);
	m_editRangeEnd.GetWindowTextA(strTemp3);
	strData.Format("%s, %s, %s", strTemp1, strTemp2, strTemp3);
	m_listRange.AddString(strData);

}


void CTrafficSimulationDlg::OnBnClickedButtonClear()
{
	// TODO: Add your control notification handler code here
	m_listRange.ResetContent();
}

void CTrafficSimulationDlg::AddData(CArray<int, int> * pArray, int nValue)
{
	// filtering

	pArray->Add(nValue);

}

void CTrafficSimulationDlg::OnChangeEditFactor1()
{
	// TODO:  If this is a RICHEDIT control, the control will not
	// send this notification unless you override the CDialogEx::OnInitDialog()
	// function and call CRichEditCtrl().SetEventMask()
	// with the ENM_CHANGE flag ORed into the mask.

	// TODO:  Add your control notification handler code here
	CString strTemp;
	double dblFactor1;
	m_factor1.GetWindowTextA(strTemp);
	dblFactor1 = atof(strTemp);
	strTemp.Format("%f", 1-dblFactor1);
	m_factor2.SetWindowTextA(strTemp);
}
```

// TrafficSimulationDlg.h : header file

```
#pragma once

#define MAX_ARRAY 20

// CTrafficSimulationDlg dialog
class CTrafficSimulationDlg : public CDialogEx
```

```
	{
	// Construction
	public:
		CTrafficSimulationDlg(CWnd* pParent = NULL);	// standard constructor
		BOOL Analysis_Intersection_forward(int nInterSection, int nTime);
		BOOL Analysis_Intersection_backward(int nInterSection, int nTime);
		void CreateTimeLine();
		void DisplayStatus(int nInterSection);
		BOOL Choose_Interval(int nInterSection, int nSectionInterval);
		double GetStandardDeviation(CArray<int, int>* pArray);
		double GetMean(CArray<int, int>* pArray);
		double GetSum(CArray<int, int>* pArray);
		void AddData(CArray<int, int> * pArray, int nValue);
		void MakeArray_Diff(CArray<int, int>* pSrcArray, CArray<int, int>* pDesArray,
int nDirection);


		CString m_strFilePath;
		CString m_strExcelsheet;
		int m_nIntersectionNum;
		double m_dblcarSpeed;
		int m_nSecondRange;
		BOOL m_bStop;
		int m_nNumDisplay;
		int m_nDisplayIndex;
		CStdioFile m_destFile;
		CStdioFile m_destFile2;
		CArray<int, int> arrayTime[MAX_ARRAY];
		CArray<int, int> arrayOffset[MAX_ARRAY];
		CArray<int, int> arrayExtraTimeCP1[MAX_ARRAY];
		CArray<int, int> arrayExtraTimeCP2[MAX_ARRAY];
		CArray<int, int> arrayUsedPhase2[MAX_ARRAY];
		CArray<int, int> arrayUsedPhase6[MAX_ARRAY];
		CArray<BOOL, BOOL> arrayTimeLine_forward[MAX_ARRAY];
		CArray<BOOL, BOOL> arrayTimeLine_backward[MAX_ARRAY];
		double m_dblTime[MAX_ARRAY];
		double m_dblSpeed[MAX_ARRAY];
		double m_dblDist[MAX_ARRAY];
		int m_nSectionInterval[MAX_ARRAY];
		int m_nSectionGroup[MAX_ARRAY];
		int m_nBandwidth1;
		int m_nBandwidth2;
		int m_nBandwidth_count1;
		int m_nBandwidth_count2;
		int m_nExcelRow;
		CArray<int, int> m_arrayIndividual_forward;
		CArray<int, int> m_arrayIndividual_backward;
```

```
	CArray<int, int> m_arrayDiff_forward;
	CArray<int, int> m_arrayDiff_backward;
	int m_nRangeStart[MAX_ARRAY];
	int m_nRangeEnd[MAX_ARRAY];
	int m_nCurGroupCheck;
	int m_nPreGroupCheck;
	int m_nEnableWriting;
	double m_dblFilter;
	double m_dblFilter2;
	int m_nStep;
	double m_dblWeightedFactor_forward;
	double m_dblWeightedFactor_backward;

	BOOL LoadExcel();
// Dialog Data
	enum { IDD = IDD_TRAFFICSIMULATION_DIALOG };

	protected:
	virtual void DoDataExchange(CDataExchange* pDX);   // DDX/DDV support


// Implementation
protected:
	HICON m_hIcon;

	// Generated message map functions
	virtual BOOL OnInitDialog();
	afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
	afx_msg void OnPaint();
	afx_msg HCURSOR OnQueryDragIcon();
	DECLARE_MESSAGE_MAP()
public:
	afx_msg void OnBnClickedButtonStart();
	afx_msg void OnBnClickedButtonSearch();
	CEdit m_filepath;
	CEdit m_excelsheet;
	CEdit m_intersectionNum;
	CEdit m_carspeed;
	CEdit m_secRange;
	CEdit m_status_display;
	afx_msg void OnBnClickedButtonStop();
	CButton m_enableWriting;
	afx_msg void OnBnClickedButtonAdd();
	CEdit m_editRangeEnd;
	CEdit m_editRangeIntersection;
	CEdit m_editRangeStart;
	CListBox m_listRange;
```

```
		afx_msg void OnBnClickedButtonClear();
		CEdit m_filter;
		CEdit m_step;
		afx_msg void OnChangeEditFactor1();
		CEdit m_factor1;
		CEdit m_factor2;
		CEdit m_filter2;
	};

	// TrafficSimulation.cpp : Defines the class behaviors for the application.
	//

	#include "stdafx.h"
	#include "TrafficSimulation.h"
	#include "TrafficSimulationDlg.h"

	#ifdef _DEBUG
	#define new DEBUG_NEW
	#endif


	// CTrafficSimulationApp

	BEGIN_MESSAGE_MAP(CTrafficSimulationApp, CWinApp)
		ON_COMMAND(ID_HELP, &CWinApp::OnHelp)
	END_MESSAGE_MAP()


	// CTrafficSimulationApp construction

	CTrafficSimulationApp::CTrafficSimulationApp()
	{
		// support Restart Manager
		m_dwRestartManagerSupportFlags =
AFX_RESTART_MANAGER_SUPPORT_RESTART;

		// TODO: add construction code here,
		// Place all significant initialization in InitInstance
	}


	// The one and only CTrafficSimulationApp object

	CTrafficSimulationApp theApp;


	// CTrafficSimulationApp initialization
```

```
BOOL CTrafficSimulationApp::InitInstance()
{
	// InitCommonControlsEx() is required on Windows XP if an application
	// manifest specifies use of ComCtl32.dll version 6 or later to enable
	// visual styles.  Otherwise, any window creation will fail.
	INITCOMMONCONTROLSEX InitCtrls;
	InitCtrls.dwSize = sizeof(InitCtrls);
	// Set this to include all the common control classes you want to use
	// in your application.
	InitCtrls.dwICC = ICC_WIN95_CLASSES;
	InitCommonControlsEx(&InitCtrls);

	CWinApp::InitInstance();


	AfxEnableControlContainer();

	// Create the shell manager, in case the dialog contains
	// any shell tree view or shell list view controls.
	CShellManager *pShellManager = new CShellManager;

	// Standard initialization
	// If you are not using these features and wish to reduce the size
	// of your final executable, you should remove from the following
	// the specific initialization routines you do not need
	// Change the registry key under which our settings are stored
	// TODO: You should modify this string to be something appropriate
	// such as the name of your company or organization
	SetRegistryKey(_T("Local AppWizard-Generated Applications"));

	CTrafficSimulationDlg dlg;
	m_pMainWnd = &dlg;
	INT_PTR nResponse = dlg.DoModal();
	if (nResponse == IDOK)
	{
		// TODO: Place code here to handle when the dialog is
		//  dismissed with OK
	}
	else if (nResponse == IDCANCEL)
	{
		// TODO: Place code here to handle when the dialog is
		//  dismissed with Cancel
	}

	// Delete the shell manager created above.
	if (pShellManager != NULL)
```

```
	{
		delete pShellManager;
	}

	// Since the dialog has been closed, return FALSE so that we exit the
	//  application, rather than start the application's message pump.
	return FALSE;
}
```