ABSTRACT

KARMAKAR, NABARUNA. Data-Driven Optimization and Decision Making for Signal Timing on Coordinated Arterial Streets. (Under the direction of Drs. Michael G. Kay and Billy M. Williams).

Traffic signals aim at improving overall flow of traffic and increasing safety at busy intersections. Coordinated signal systems are designed to reduce driver frustration and traffic congestion on major arterials by developing timing plans that minimize the number of stops and maximize green bands, also known as arterial bandwidth. Offset optimization helps improve coordination by synchronizing the start of green indication for the intersections along arterial streets. Most traditional optimization methods have been developed based on programmed greens and reds. However, in most semi-actuated signal systems, uncoordinated phases may be skipped or terminated early and the remaining green time is allocated back to the coordinated phases. This makes the cycle-by-cycle bandwidths dynamic, and offset optimization under dynamic bandwidth is a relatively new area of research. Using information regarding start and end of actual field observed green times from archived signal controller logs, it is possible to enumerate all possible offset combinations and evaluate the bandwidth for each combination. However, exhaustive search can become extremely impractical as the size of the problem grows. This research is aimed at finding alternatives to exhaustive search algorithm to generate the solution space for this problem and finding the optimal offset combination.

First, a method is proposed to estimate the entire solution space for dynamic bandwidth optimization on arterials, using Locally Weighted Projection Regression (LWPR), by using a percentage of the input-output combinations from the exhaustive search method to train the model. The performance of this algorithm in terms of accuracy and robustness of estimating optimal solution is evaluated using two study sites and two training data sampling strategies.

Test results indicate that LWPR worked well in terms of average error in prediction, with 10% randomly sampled training data for a three-intersection site and 30% random sampling for a four-intersection site. Training the model with every nth input-output pairs provides the model with more information, enabling it to provide a higher accuracy with lesser percentage of training data. This technique may prove to be more computationally effective when systems with more than four signals are considered and when augmented with parallel computing.

Next, a new mixed integer linear programming (MILP) model formulation is discussed, which takes the actual used green times on the major through phases on an arterial as input to find the optimal offset combination to maximize two-way bandwidth on a major arterial. The results of the MILP were compared against MAXBAND formulation, using some hypothetical scenarios. It was also tested on a real-world three-intersection site and the results were found to be comparable to the results of the exhaustive search algorithm. More rigorous testing of this model using real-world signal log data is required to make it more robust. The model is flexible enough to include further extensions such as incorporating queue clearance, secondary bandwidths, etc.

While traffic signals play a critical role in arterial traffic control, transportation agencies frequently delay plans for proactive signal maintenances due to budget and staff limitations. The final contribution of this research is an intuitive methodology for calculating signal performance metrics using archived cycle-by-cycle signal controller split monitor logs. This method provides exact timing recommendations for signal re-timing for each movement at an intersection, backed by insights provided by a large amount of more readily available data. Using a major arterial street in Garner, North Carolina as a case study, this methodology is demonstrated and quantitative insights into the effectiveness of the existing signal timing plans at an intersection level are produced. Notably, this method requires that an analyst use expert judgement, thus the development of threshold values for data can further automate its application.

Data-Driven Optimization and Decision Making for Signal Timing on Coordinated Arterial Streets

by
Nabaruna Karmakar

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

Industrial Engineering

Raleigh, North Carolina

2019

APPROVED BY:

Michael G. Kay Committee Co-Chair	Billy M. Williams Committee Co-Chair
Yahya Fathi	Nagui M. Rouphail

DEDICATION

To my family and friends

BIOGRAPHY

Nabaruna Karmakar was born in Kolkata, India on February 2, 1990 and spent most of her school years in Mumbai, India. She earned a bachelor’s degree in Electrical and Electronics Engineering from Manipal University, India in 2012. In 2013, she joined North Carolina State University in Raleigh, NC, to pursue a Master’s in Industrial Engineering. She started working with the Institute of Transportation Research and Education (ITRE) at NC State University as a Graduate Research Assistant in 2014, when she embarked upon her journey to becoming a PhD in Industrial Engineering. Her interests include building optimization models for real-world problems and using cutting edge technology for data analytics. She has been working at SAS Institute in Cary, NC with the OR Center of Excellence as an Operations Research Intern since June 2018 and has taken on a full-time position as Operations Research Specialist in January 2019.

ACKNOWLEDGMENTS

I would like to thank my advisors Dr. Billy Williams and Dr Michael Kay for their support and guidance in my research. I appreciate all the valuable advice that they gave me to help me get through this rigorous process of finishing a PhD. I would also like to thank Dr. Nagui Rouphail and Dr. Yahya Fathi, my other committee member, for evaluating my work and helping me improve my dissertation.

I would like to thank all my colleagues at Institute of Transportation Research and Education (ITRE) for being wonderful mentors. I would like to thank Dr. Sangkey Kim, who got me interested in this research and encouraged me to take his research on dynamic bandwidth forwards. I would also like to thank Mr. Thomas Chase, at ITRE, who taught me a lot about signal system performance and without whose help, this dissertation would not have been accomplished. I also want to appreciate Chang Baek, at NCDOT for helping me collect the data for my dissertation.

I would like to acknowledge my mentor and manager at SAS Institute, Natalia Summerville and Ivan Oliveira for giving me the opportunity to apply my skills in an industry setting and a great start to my career. I eagerly look forward to working with their talented team and solving real-world optimization problems.

I would like to thank all my friends for standing by me and making Raleigh a home away from home. I would like to acknowledge the endless support and love I received from my family and from my dog, Boomer. I would like to thank my sister for being a great role model, by becoming the first person in our family to embark upon the journey to becoming a PhD. Last, but not least, I want to thank Michael for being a constant cheerleader and supporting me in every possible way.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Background and Motivation

Increasing traffic volumes place heavy demands on urban streets in already overcrowded areas, necessitating the need for effective traffic control. Traffic signals are one of the more familiar types of traffic control on arterial streets, which aim at improving overall flow of traffic and increase safety at busy intersections. In a typical urban area, they might direct the movement of as many as 100,000 vehicles per day (Koonce and Rodegerdts, 2013). However, when configured poorly, they cause delay and driver frustration, which can further lead to an increased probability of crashes. Despite their critical role in arterial traffic control, proactive management and maintenance of traffic signals are frequently delayed or cancelled, due to insufficient budget and staff. As a result, more than half of the signals in the U.S. need repair, replacement, or upgrading (Koonce and Rodegerdts, 2013). In 2012, the National Traffic Signal Report Card, released by the National Transportation Operations Coalition, which consisted of the composite consolidated responses of 241 agencies across the United States and Canada, reported that the national score had improved slightly to an overall D+ from a D in 2007 (Denney et al., 2012).

To reduce delays at signalized intersections, a system of signals that are close to each other need to be coordinated so that cars arriving at the first intersection can traverse the entire system of signals, without having to stop. Enabling better progression through a major corridor not only reduces congestion and fuel emissions, but also enhances road safety. The benefits of coordinated signal timing are manifold and are mainly rooted in the fact that it minimizes the number of stops and maximizes green bands, also known as *arterial bandwidth*. Apart from increasing traffic flow and reducing peak-hour delay, coordinated signal timing can also be optimized for slower speeds, creating an uninterrupted flow for bicyclists or for a pedestrianfriendly downtown. However, there are many pitfalls that must be avoided if improved signal timing plans are to be properly developed and successfully implemented (Buckholz, 1993).

Each traffic signal operates under a unique time of day plans and different timing parameters. In a coordinated system, the signals are configured by tuning parameters such as cycle length, splits and offsets. Much research, as will be summarized in the literature review in Chapter 3, has focused on developing mathematical models to maximize bandwidth by computing optimal parameters of traffic signal controllers, such as cycle length, start of green, and offsets between intersections. Traditional methods related to offset optimization have assumed fixed green times at each intersection, and hence, a repeating traffic pattern where no movements are skipped. These studies do not consider the stochastic nature of demand across time of day plans and for different movements at each intersection.

Moreover, most current traffic signals operate on a semi-actuated mode of control. Semi-actuated traffic signal control uses detection only for the minor movements at an intersection, while the phases associated with the major-road through movements are operated as non-actuated (Koonce and Rodegerdts, 2013). Under this type of control, vehicle actuation on uncoordinated phases may cause the phase to skip or gap-out, making it end earlier and allocating the remaining green time to the coordinated phases on the major streets. Then the coordinated phases would start earlier than planned, which may disrupt progression. Hence, the green times for coordinated phases are dynamic and may include additional green time, unused by the non-coordinated movements. This further makes the green bandwidth across cycles on such a corridor stochastic and the use of programmed bandwidth optimization techniques become less accurate.

Offset optimization under dynamic bandwidth is a relatively new area of research, since most traditional mathematical models have focused on programmed green times. Data regarding the start and end of green times for the various phases on a coordinated corridor can be gathered from signal controller log. This data can serve as a valuable input to develop new mathematical models to maximize progression on signalized corridors, considering the stochastic nature of actual used green times. There is need for a robust optimization technique that can find optimal offset combinations and maximize the cycle-by-cycle bandwidths. It should be scalable and efficient enough to conduct analysis for corridors having many signalized intersections.

Optimization of coordinated signal timing plans alone is not enough to ensure reliable system performance on urban streets. With growth in population and transportation activities, maintenance and re-timing of traffic signals become a key factor in urban areas. Advanced Traffic Management Systems (ATMS) enable the logging of various useful information that can give insights into the performance of arterial streets. The types of information, however, are available in various inconsistent formats and differ between state DOTs and agencies. With the advent of Automated Traffic Signal Performance Measures (ATSPMs), state DOTs have a new of continuously monitoring performance of traffic signals, using high-resolution data and base their re-timing efforts on performance-based metrics. However, this approach requires the use of high-resolution data from vehicle detectors, which are expensive to install and maintain, in addition to certain data acquisition equipment and communication infrastructure. Moreover, state DOTs and traffic agencies not only lack resources, but often lack a means of prioritizing re-timing projects to optimize the resources they have. Hence, signal re-timing project planning needs to be more data-driven, defensible and efficient than is currently the state of practice in the United States.

1.2. Traffic Signal Systems – Concepts and Definitions

Chapter 19 of the Highway Capacity Manual (HCM, 2010) discusses the concepts used to describe traffic signal operations. It broadly classifies traffic signal controllers as having *pre-timed control* and *actuated control.* A pre-timed controller may operate in a coordinated or non-coordinated state. An actuated controller can be further classified as *fully actuated, semi-actuated, or coordinated-actuated* signal systems.

Coordinated-actuated control is a variation of semi-actuated operation. It uses the controller's force-off settings to constrain the non-coordinated phases associated with the minor movements so that the coordinated phases are served at the appropriate time during the signal cycle and progression for the major movements is maintained. This type of control is associated with a constant cycle length.

Signalized intersections that are close to one another on the same street are often operated as a *coordinated signal system*, in which specific phases or movements at each intersection are operated on a common time schedule to permit the continuous flow of the associated movements at a planned speed. The signals in a coordinated system typically operate by using pre-timed or coordinated-actuated control, and the coordinated phases typically serve the major-street through movements. This functionality can profoundly enhance the quality of service of the facility. A semi-actuated controller does not allocate a green signal to side streets if there is no demand for it.

Phasing refers to the method by which a traffic signal accommodates the various directional movements at an intersection in an efficient manner. The Manual on Uniform Traffic Control Devices for Streets and Highways (MUTCD), published by the Federal Highway Administration (FHWA), defines a signal phase as the right-of-way, yellow change, and red clearance intervals in a cycle that are assigned to an independent traffic movement or combination of traffic movements.

The National Electrical Manufacturers Association (NEMA) signal phasing refers to a generally accepted rule for numbering the eight standard vehicle phases of an intersection, where numbers are used to designate the left turn phases while even numbers are used to designate the through phases. Typically, a pedestrian movement is associated with the concurrent vehicular phase running parallel and adjacent to it. Figure 1-1 below shows the phase numbering for different vehicular and pedestrian movements at a four-leg intersection.

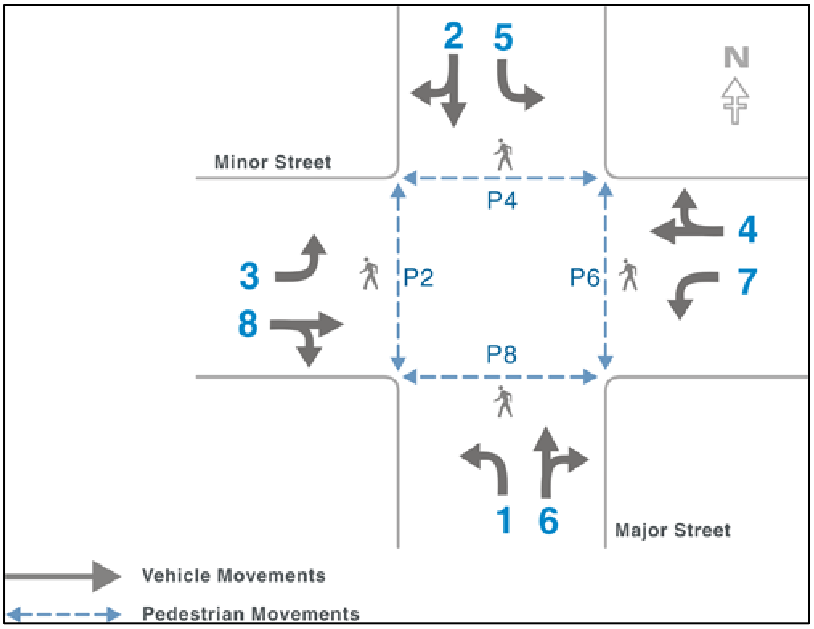


Figure 1-1 - Typical Vehicular and Pedestrian Movements at a four-leg intersection (Traffic Signal Timing Manual, 2013)

A *cycle* is defined as the total time to complete one sequence of signalization for all movements or phases at an intersection. In an actuated controller unit, the cycle is a complete sequence of all signal indications. A *split* refers to the amount of time in a cycle length given to a particular phase and includes the green, yellow and red clearance time of a particular phase. The sum of all the splits, in seconds, should equal the cycle length.

Offset reference points help structure the relationship among coordinated intersections by defining the point in time when the cycle begins timing. *Offset* is defined as the time from a reference point, such as the start of green or yellow of the coordinated phase at an intersection, to the same reference point at the other intersections. *Offset optimization* is the science of selecting offsets for a corridor of signalized intersections to allow platoons of vehicles to move through the corridor without stopping at red lights.

A semi-actuated controller allows skipping or gapping out of minor phases, when there is no demand on side streets and re-allocates that unused green time to the major street in the next cycle. This is called an *early return to green* and can vary between different cycles and different time of day plans. The early-return-to-green problem is particularly evident when low volumes at the upstream intersection simultaneously occur with high volumes at the downstream intersection.

Traffic Signal Coordination can be defined as synchronizing two or more nearby signals on a certain street so that vehicles can progress along the street with minimal stops and delays. This entails the designation of a phase or multiple phases as the coordinated phases, typically the major streets which experience consistent high demand. Coordinated phases are distinguished from uncoordinated phases because they always receive a minimum amount of assigned green time every cycle. A common cycle length helps maintain the guaranteed green interval and maintain the coordinated relationship between intersections.

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.

A *Time-Space* diagram is a chart that plots ideal or hypothetical vehicle platoon trajectories through a series of signalized intersections. Figure 1-2 shows how vehicles would progress through intersections through the green-bands in each direction at the posted speed limit, represented by the blue lines.

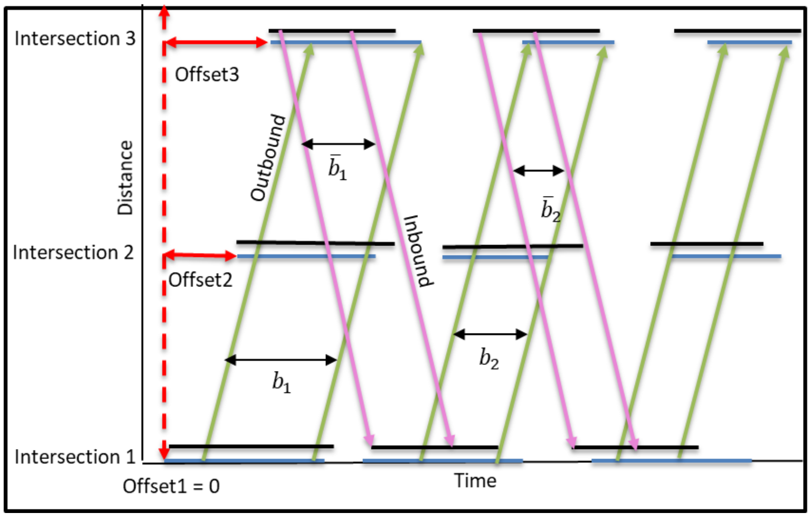


Figure 1-2 - Time-Space Diagram of a Coordinated Timing Plan on a system of 3 intersections

1.3. Problem Definition

The conventional approach to developing a signal timing plan involves a very rigorous process of data collection, evaluation, installation, and manual field-tuning. The manual tuning of offsets for coordination can lead to selection of sub-optimal offset combinations for the corridor, if the cycles observed during fine-tuning are not representative of normal conditions. The process of data collection before and after every fine-tuning effort is labor intensive and expensive. For coordinating a system of signals, the offsets of the start of green times at successive signalized intersections, is used as the decision variable to maximize the bandwidth across the corridor. Most studies have used programmed or allotted green time on coordinated phases as a basis for calculating bandwidth, which assumes maximum and uniform utilization of allotted green on all phases.

A recent study has established that bandwidth calculated based on static green times differ greatly from dynamic bandwidth (Kim, 2014). A key contribution of this research was the development of the Dynamic Bandwidth Allocation Tool (DBAT), an exhaustive search algorithm that calculates the bandwidth for a system of intersections 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. Using information regarding start and end of green times from signal controller log, it enumerates all possible offset combinations, with the first intersection as the reference point, and evaluates the bandwidth for each combination. While it ensures finding the globally optimal solution, the calculation time for an exhaustive search grows exponentially as the size of the problem grows. Hence, it becomes impractical to use this algorithm for arterials with higher number of intersections. However, knowledge of the entire solution space is desired to analyze the reliability and robustness of different solutions. It helps an analyst investigate the neighborhood of the global optimal solution and find one that is most effective for the road under study. Hence, alternatives to using explicit enumeration holds the promise for a more computationally effective method for estimating the entire solution space of this problem.

Another outcome of the dynamic bandwidth research was the formulation of a linear programming model to find the optimal offset combination to maximize the two-way bandwidth on a signalized corridor (Kim, 2014). This formulation uses the cycle-by-cycle dynamic green times from signal controller log for through phases on major streets as input and maximize the two-way bandwidth by ensuring that the phases get the required amount of green time in each cycle. However, the Linear Programming model cannot be extended to some extreme cases, such as including a greater number of intersections or reducing progression speeds. For the model to provide a reliable solution, it should be able to handle large amounts of signal controller data for a given time-of-day plan and be scalable to corridors with large number of intersections. Hence, there is need for a more robust maximization technique to find the global optimal offsets.

Furthermore, re-timing of signal systems is a process that can greatly increase the flow of traffic through signalized corridors and requires constant monitoring of various sources of field data. It helps in designing effective signal timing to meet changing travel patterns and characteristics. ATSPMs present a promising platform for analysts to see trends in different signal performance metrics, but they are still in the investigative phases and have not been widely adopted by DOTs and agencies. They also require installation of very expensive equipment for collection of high-resolution detector data. Decisions regarding selection of signals for re-timing are still based on ad-hoc simulation runs and, in most cases, solely citizen complaints. High resolution signal controller data, augmented with other sources, such Bluetooth or Vehicle Probe-based Travel Time Data, can be used to establish enhanced performance metrics. This data fusion provides the potential for mining through the information provided by the different data sources and developing decision making models that help identify signal systems in need of re-timing.

1.4. Outline of the Dissertation

This dissertation is divided into four other main chapters, which include three different mathematical approaches to solving the issue of optimization and re-timing decision making for signalized intersections and a final chapter which presents conclusion and direction for future research.

The second chapter tackles the problem of estimating the entire solution space of the dynamic bandwidth optimization, using locally weighted regression techniques. It includes a literature review for various machine learning algorithms that can be used in the context of function approximation and finally discusses the performance and results of using Locally Weighted Projection Regression (LWPR) algorithm to estimate the search space. The third chapter contains a detailed literature review of mathematical models developed for the bandwidth maximization problem, pertaining mostly to programmed bandwidths. It presents a new Mixed Integer Linear Programming (MILP) formulation to solve the problem of maximizing dynamic bandwidth on major arterials, by using cycle-by-cycle actual used green times for each of the major approaches from the signal controller split monitor data as input. The fourth chapter presents a novel approach for prioritizing signalized intersections for re-timing. By calculating and visualizing signal performance metrics using signal controller log data, a real-world case study is used to demonstrate how the method can intuitively assist traffic engineers in assessing which signal systems can benefit from re-timing. Finally, the fifth chapter summarizes the key findings of the research and leads the way for future research.

1.5. References

Buckholz, J. (1993). The 10 major pitfalls of coordinated signal timing. ITE journal, 63(8), 26-29.

Denney Jr, R. W., Curtis, E., & Olson, P. (2012). The national traffic signal report card. ITE Journal, 82(6), 22-26.

HCM (2010). Highway capacity manual. Washington, DC, 2.

Kim, S., Hajbabaie, A., Williams, B. M., & Rouphail, N. M. (2014). Development of near real time performance measurements for closed-loop signal systems (CLS) using historical traffic data from existing loop detectors and signal timing data (No. FHWA/NC/2012-12). North Carolina. Dept. of Transportation.

Koonce, P., & Rodegerdts, L. (2013). Traffic signal timing manual (No. FHWA-HOP-08-024). United States. Federal Highway Administration.

2. USING LOCALLY WEIGHTED PROJECTION REGRESSION TO ESTIMATE SOLUTION SEARCH SPACE FOR DYNAMIC BANDWIDTH ANALYSIS ON ARTERIAL STREETS

2.1. Introduction

Traffic signals are designed to not only enable better progression through arterial streets with heavy demand patterns, but also reduce congestion and fuel emissions. A well-timed traffic signal system can also help in enhancing road safety. To reduce delays along a corridor, a system of signals that are close to each are usually coordinated such that platoons of cars arriving at the first intersection can traverse the entire system of signals, without having to stop. The benefits of coordinated signal timing are manifold and are mainly rooted in the fact that it minimizes the number of stops and maximizes green bands, also known as arterial bandwidth.

Some of the tuning parameters used to configure timing plans in a coordinated system are cycle length and offsets. Offset is defined as the time from a reference point, such as the start of green or yellow of the coordinated phase at an intersection to the same reference point at the other intersections in the system. Offset optimization is the process of selecting offsets for a corridor of signalized intersections to maximize the amount of time platoons of vehicles can move through the corridor without stopping at red lights.

Traditional methods related to offset optimization have assumed fixed green times, resulting in programmed bandwidths at each intersection, and hence, a repeating traffic pattern where no movements are skipped. These studies do not consider the extra green time that is donated by the minor approaches at an intersection to the through movements on the major streets when they have no demand for it, making the bandwidths dynamic. Offset optimization under dynamic bandwidth is a relatively new area of research. Some studies related to dynamic offset optimization focus heavily on detector data, requiring additional expensive hardware or software upgrades (Day and Bullock, 2011; Gettman et al., 2007). Data regarding the start and end of green times for the various phases on a coordinated corridor, gathered from archived signal controller log, can serve as a more readily available data source for developing bandwidth maximization algorithms. A recent study has developed the Dynamic Bandwidth Allocation Tool (DBAT), an exhaustive search algorithm that calculates the bandwidth for a system of intersections, by counting the number of seconds that would accommodate a through-vehicle trajectory traveling at a specified progression speed without encountering a red indication at any of the intersections (Kim, 2014). Using information regarding start and end of green times from archived signal controller logs, it enumerates all possible offset combinations with the first intersection as the reference point and evaluates the bandwidth for each combination.

While such an exhaustive search algorithm ensures a globally optimal solution, its computational complexity grows exponentially as the size of the problem grows, making it impractical to use this algorithm for arterials with higher number of intersections. The total number of feasible solutions at a search interval of every second is given by the following equation:

$$Number\ of\ feasible\ solutions = (Cycle\ Length)^{i-1} \tag{2.1}$$

where,

i = Number of intersections in the corridor

In practice, the distribution of sufficient bandwidth in both directions of travel on a major arterial is correlated with system performance as well as road safety. Distribution of bandwidth should be considered from two different perspectives – reliability and robustness. The reliability perspective considers how bandwidth varies cycle by cycle as demand changes, while robustness considers how the bandwidth may vary with factors such as the progression speed, etc. The global optima cannot always guarantee a reliable system performance in the long run. Knowledge of the entire solution space helps an analyst investigate reliability and robustness of offset combinations in the neighborhood of the global optimal solution. Finding alternatives to explicit enumeration holds the promise for a more computationally effective method for estimating the entire solution space for the dynamic bandwidth optimization problem.

In problems where the method to compute the underlying function is computationally intensive, it is worthwhile to replace it with a simpler, more efficient approximating function. The problem of optimizing dynamic bandwidth across signalized intersections is not a trivial problem, and the underlying function is not linear. The solution space does not depend linearly on any of the input parameters and requires a much more complex method than linear regression models. This can be accomplished using Machine Learning techniques which provide systems the ability to learn from samples of data without relying on rules-based programming.

This chapter proposes a method to estimate the entire solution space for dynamic bandwidth optimization on arterials, which is substantially faster than exhaustive search. It includes a literature review on existing machine learning techniques used for function approximation. A methodology to train a Machine Learning algorithm using a percentage of the input-output combinations from the exhaustive search method to estimate the solution space of all offset combinations is explained. Two different sampling techniques, random sampling and n^{th} solution sampling are explored using two real-world case studies. The performance of the two sampling techniques in terms of accuracy and robustness of estimating optimal solution is evaluated using two study sites.

2.2. Literature Review

Function approximation is applied when it is too costly either in terms of time or complexity to compute the true function or when this function is unknown and just a rough idea of its main properties is needed. This is usually done by computing the approximated function at a few points and predicting a response for the other values. Approximation problems can be largely classified as local approximation and global approximation.

Regression analysis is a statistical process of estimating a relationship between a set of input variables and their outputs, usually using least squares fitting. Orthogonal least-squares fitting is a good method to avoid the unsymmetrical treatment of the data and effects of regression toward the mean (de Julian-Ortiz et al, 2010). However, in real world applications, a simple linear regression cannot capture the dynamic nature and the non-linearity of the processes. As the availability of large-scale datasets increases regularly, so does the challenge in dealing with them. Simple regression techniques suffer from the potentiality of overfitting when working with large datasets.

Non-parametric regression is a category of regression analysis in which the predictor is constructed according to information derived from the data, using larger sample sizes than regression based on parametric models. Over the past few decades, many non-parametric Machine Learning algorithms have been developed, which focus on supervised learning techniques using training data sets and cross validating against a test data set to achieve a higher accuracy in model prediction. It should be noted that there is no single machine learning technique that will work best for all datasets; although given a particular problem, some methods can significantly outperform others (Vink & Haan, 2012). Many popular approaches include Neural Networks (Atkeson and Schaal, 1995), Support Vector Machines (Chang & Lin, 2011), and Gaussian Process Regression (Nguyen-Tuong et al., 2009; Williams & Rasmussen, 1996). These algorithms are all global learning systems, which try to minimize an overall loss function, such as expected sum squared errors.

Park et al. (2014) showed the superiority of non-parametric models over parametric models for predicting options prices especially on KOSPI 200 Index options. Their experimental comparisons between state-of-art parametric no-arbitrage models and non-parametric machine learning models showed that the Gaussian process regression method delivers the overall outstanding performance in prediction accuracy as well as in its capability to provide the predictive distribution of option prices (Park et al, 2014). Gaussian process regression methods are easy to apply and yield higher accuracy in predictions but suffer from very high computational cost.

While global learning algorithms can be understood in terms of minimizing a global loss function such as the expected sum squared error, local learning systems split up the global learning problem into many simpler learning problems, which become useful in handling real world data with higher dimensionality. Atkeson et al. (1997) provided an extensive survey which discusses distance functions, smoothing parameters, weighting functions, local model structures, regularization of the estimates and bias, assessing predictions, handling noisy data and outliers, improving the quality of predictions by tuning fit parameters, interference between old and new data, implementing locally weighted learning efficiently, and applications of locally weighted learning.

All locally weighted regression refers to supervised learning of continuous functions (otherwise known as function approximation or regression) by means of spatially localized algorithms, which are often discussed in the context of kernel regression, nearest neighbor methods, or lazy learning (Atkeson et al., 1997). Locally weighted regression was created as a nonparametric memory-based learning method that is computationally efficient, can learn from very large amounts of data and add data incrementally. Ting et al. (1994) demonstrated in their paper that locally weighted regression methods are often preferred over global methods since they are well suited for dealing with non-stationary input data sets. Many real-world applications are dynamic in nature and require a more adaptive form of learning, where the training examples are fed to the algorithms over time, i.e., the learning process is incremental.

Locally Weighted Projection Regression (LWPR) is a state-of-the-art algorithm that was designed for incremental online learning with high dimensional input space (Vijaykumar et al., 2005). It employs nonparametric regression with locally linear models, each of which performs regression analysis with a small number of univariate regressions in selected directions in input space in the spirit of partial least squares regression, which accounts for faster computations. Many applications of LWPR have been published over the years, especially in robotic control (Pontes & Santos, 2012; Vijaykumar et al., 2005). Applications of this algorithm also include other areas such as vehicle tracking using segmented video images (Basher, 2015), prediction of hydraulic phenomenon (Agarwal, 2010), measurement of soil properties (Christy & Dyer, 2006) etc. The application of this algorithm to different domains and its accuracy in prediction shows its usefulness and versatility.

Inspired by the speed and accuracy of locally weighted regression, Nguyen-Tuong et al. (2009) proposed a method to speed-up the training and prediction process by partitioning the training data in local regions and learning an independent Gaussian process model. They compared the performance of this localized GPR method to Locally Weighted Projection Regression (Vijaykumar et al., 2005) and found that LWPR performs best in terms of computational complexity, even with large training datasets.

2.3. Methodology

2.3.1. Description of Locally Weighted Projection Regression

The algorithm selected for estimating the whole solution space of the dynamic bandwidth problem is called Locally Weighted Projection Regression (LWPR). It is a freely available open source software, which was developed for incremental learning and approximation of nonlinear functions, particularly useful in cases of dimensionality reduction. Even though the process of estimating the solution space for the dynamic bandwidth combinatorial problem does not require online learning, it provides a more computationally efficient way than total enumeration. The computational complexity of LWPR is O (n), i.e., it is linear with the number of inputs.

Locally Weighted Projection Regression is used to estimate complex nonlinear functions underlying the input data and employs a non-parametric regression with locally linear models. When compared to global function approximators like multi-layer neural networks, LWPR has the tremendous advantage that its local models learn independently and without interference. Local learning systems split up the global learning problem into many simpler learning problems, which become useful in handling real world data. This enables LWPR to learn rapidly using second order learning methods based on incremental training and adjust the kernel weights based on local information only. Locally weighted partial least squares regression is used for dimensionality reduction within each local mode.

LWPR is particularly suited for regression problem, in which the function to be learnt is non-linear and which have large amounts of training data available, at least more than 2000. The properties of LWPR are that it learns rapidly with second-order learning methods based on incremental training, uses statistically sound stochastic leave-one-out cross validation for learning without the need to memorize training data, adjusts its weighting kernels based on only local information in order to minimize the danger of negative interference of incremental learning, has a computational complexity that is linear in the number of inputs, and can deal with a large number of possibly redundant inputs.

2.3.2. Implementation of Locally Weighted Projection Regression

The methodology used in this paper is displayed schematically in Figure 2-1.

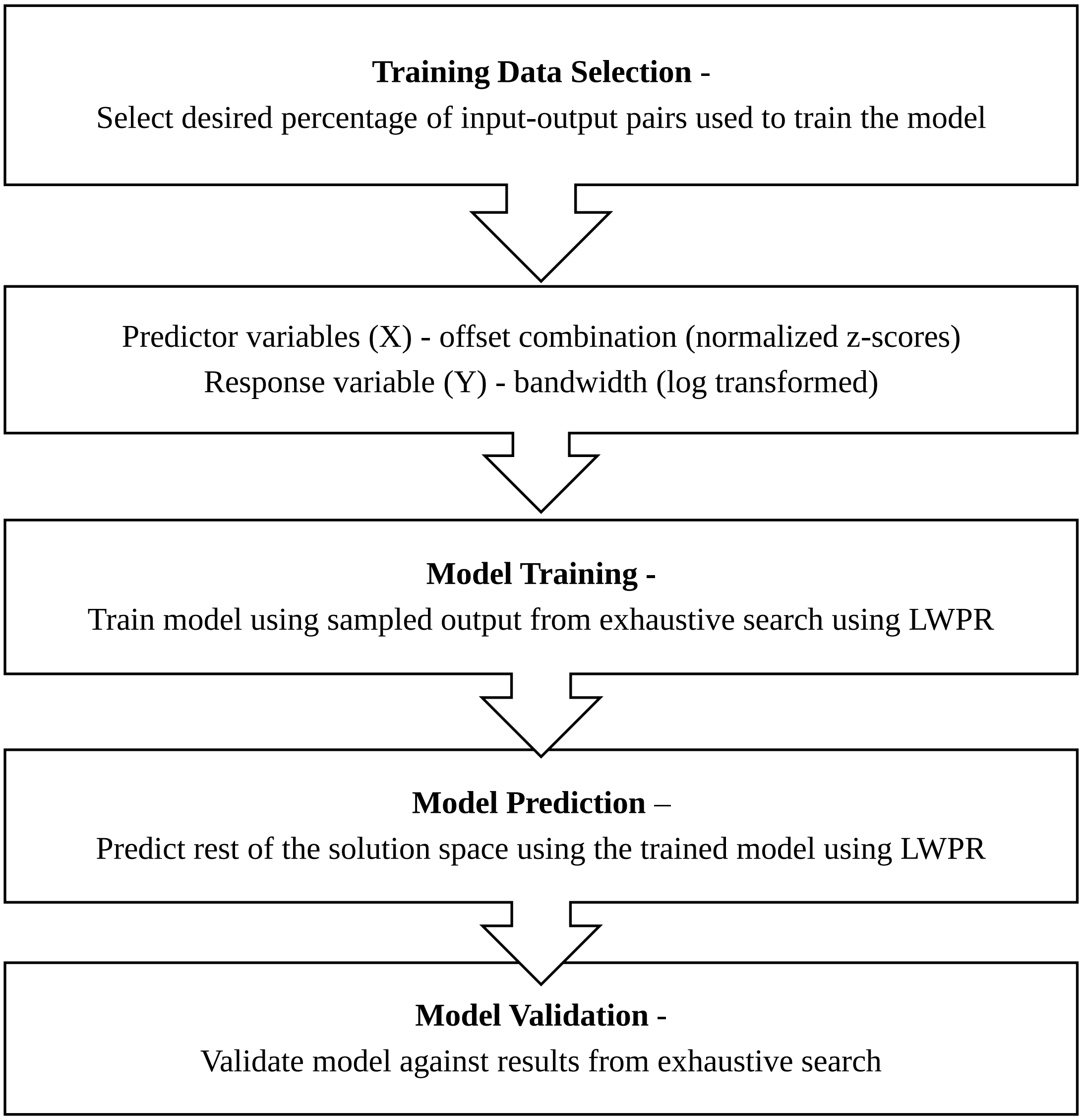

Training Data Selection -
Select desired percentage of input-output pairs used to train the model
Predictor variables (X) - offset combination (normalized z-scores)
Response variable (Y) - bandwidth (log transformed)
Model Training -
Train model using sampled output from exhaustive search using LWPR
Model Prediction –
Predict rest of the solution space using the trained model using LWPR
Model Validation -
Validate model against results from exhaustive search

Figure 2-1 - Methodology for running Locally Weighted Projection Regression

The first step in the methodology is the selection of the training data set. A percentage of the input-output pairs from the exhaustive search results of the Dynamic Bandwidth Allocation Tool (DBAT) is selected as the training dataset. The input variables or the predictor variables are the offset combinations, while the resulting dynamic bandwidth for those offset combinations make up the response or the output variable

For the training data, the offsets for the different intersections forms the independent or predictor variables, while the total bandwidth for that offset combination forms the response variable in the training set. The LWPR module requires the training dataset to be normalized. The offset values are normalized between -1 and 1, by converting them to their z-scores. The output variable or bandwidth is log transformed, since the range of the output can vary widely between different solutions. The selection of training dataset is traditionally done by randomly sampling a percentage of the entire dataset, to reduce the introduction of bias into the model and the percentage of training samples required to correctly estimate the solution space may increase with increase in number of intersections. However, the solution space for dynamic bandwidth optimization usually has a pattern, which may be more accurately captured in the training process, if every n^{th} offset combination input-output pair is sampled to form the training data. Hence, in this paper, we evaluate the performance of such training data selection strategy, along with the strategy of random sampling.

Finally, the model is tuned using the guidance provided for LWPR (Klanke et al., 2008). The implementation of the LWPR model includes the training phase and the prediction phase. After training the model, the predicted solution set is tested and validated against the truth set or the test dataset, which is the results from the exhaustive search. The implementation of LWPR was done using MATLAB 2015b on a 64-bit Windows Operating System, with 128.0 GB RAM.

2.4. Case Study - Site Description

2.4.1. Site A- 3 Intersection Site – US 70, Clayton, NC

The real-world site is a system of three coordinated signalized intersections on US 70 located in Clayton, NC, between Shotwell Road and S Robertson Street, and is shown in Figure 2-2. This route has a total length of 3,566 feet and has 22 driveways. The signal log data for the PM peak plan from 4:44 pm to 6:23 pm, with a cycle length of 220 seconds, was used as input to the DBAT exhaustive search method. This constituted a total of 29 input cycles. A progression speed of 45 mph or 66 ft. /sec was used to run the exhaustive search algorithm. Even though the runtime for the Exhaustive Search algorithm is less than 3 minutes for this site on a standard 64-bit Windows Operating System, this site was selected to test the validity of the LWPR technique in the context of this problem. The total number of offset combinations or solutions for bandwidth optimization on this site is 48,400.

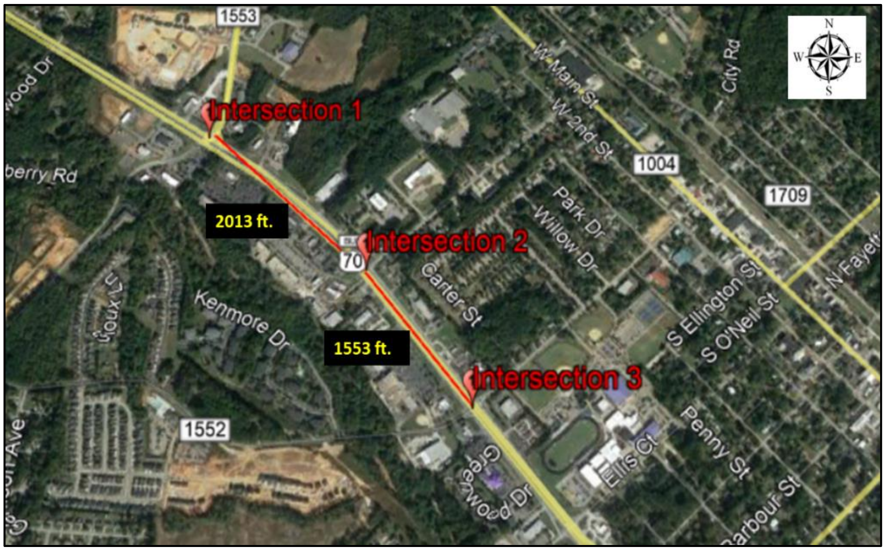


Figure 2-2 - Location of Site A

2.4.2. Site B - 4 Intersection site – National Avenue DDI, Springfield, MO

The second site used was a calibrated, validated model of the National Avenue Diverging diamond interchange (DDI) in Springfield, Missouri, shown in Figure 2-3. (Warchol et al., 2017). The DDI, also known as a double crossover diamond, has two signal-controlled crossover points that allow for free-flowing left turns and is an alternative to the conventional diamond interchange. For the test site, the model boundaries included two DDI crossovers, freeway exit and entrance ramps, and roadway segments leading into and out of the DDI crossovers along the arterial. The common cycle length across all intersection was 90 seconds. A progression speed of 25 mph or 36.66 ft. /sec was used to run the exhaustive search algorithm. The runtime for the Exhaustive Search algorithm for this site takes more than 7 hours to run on a standard Windows Operating System. The total number of offset combinations for four intersections at 1 second search interval using Equation 1 is 729,000. However, due to space complexity, the test set for this site consisted of total number of results for offset combinations, at a search interval of 3 seconds, which equaled 27,000 offset combinations. The training data was sampled from these 27,000 offset combinations or solutions and 10 iterations of training were performed on the training data, using LWPR.

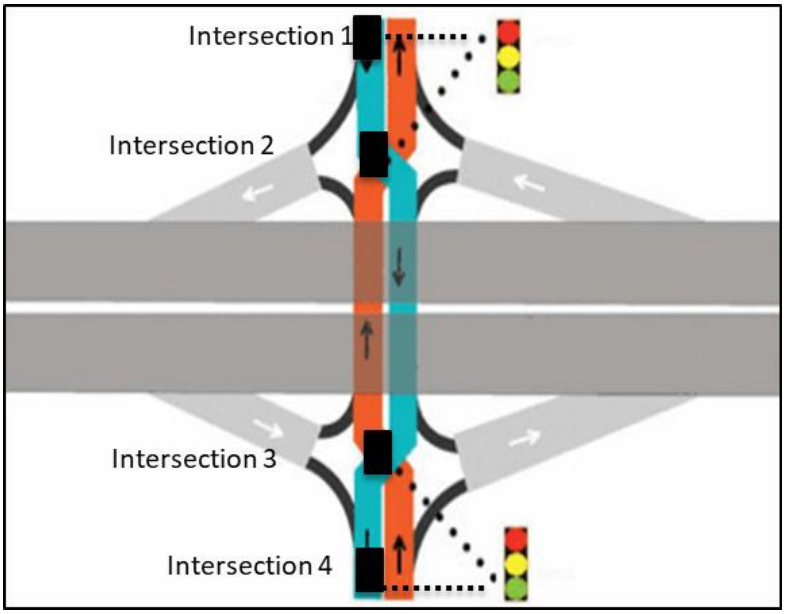


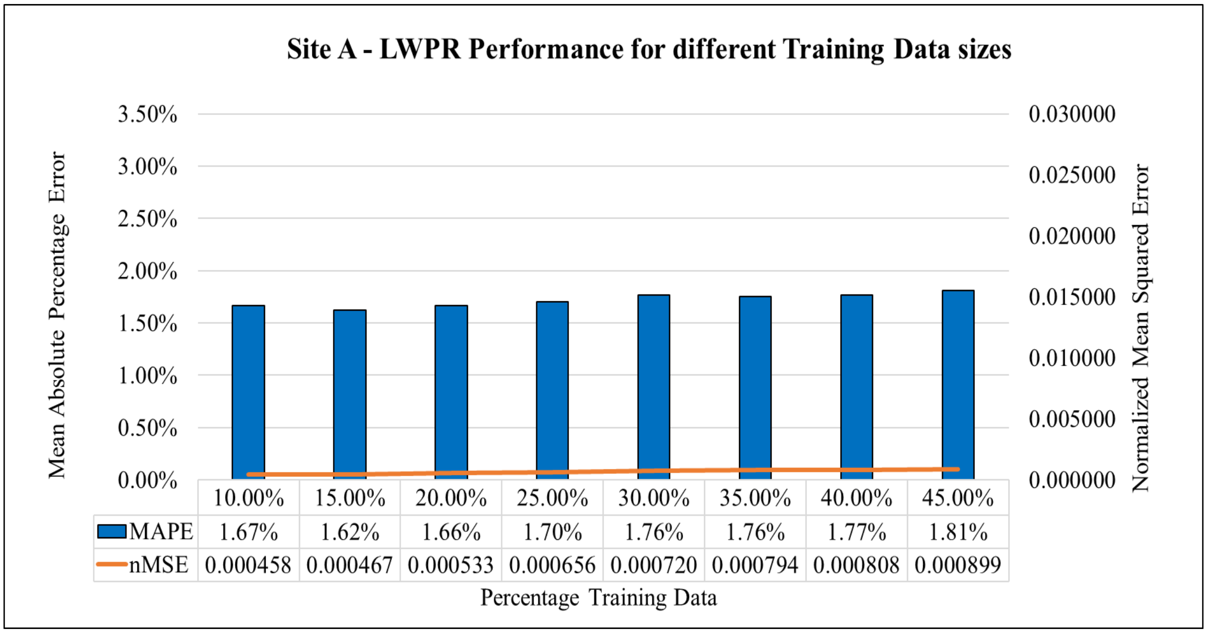
Figure 2-3 - Geometry of Site B

2.5. Results of Case Study

In this section, the results of the performance of the LWPR algorithm with respect to the two case studies are explained and discussed in detail. All implementations of the LWPR algorithm was done using MATLAB 2015b on a Windows 10 Operating System with an Intel Core i9-7900X processor and 128 GB of RAM.

2.5.1. Random Sampling Training Data Size Evaluation

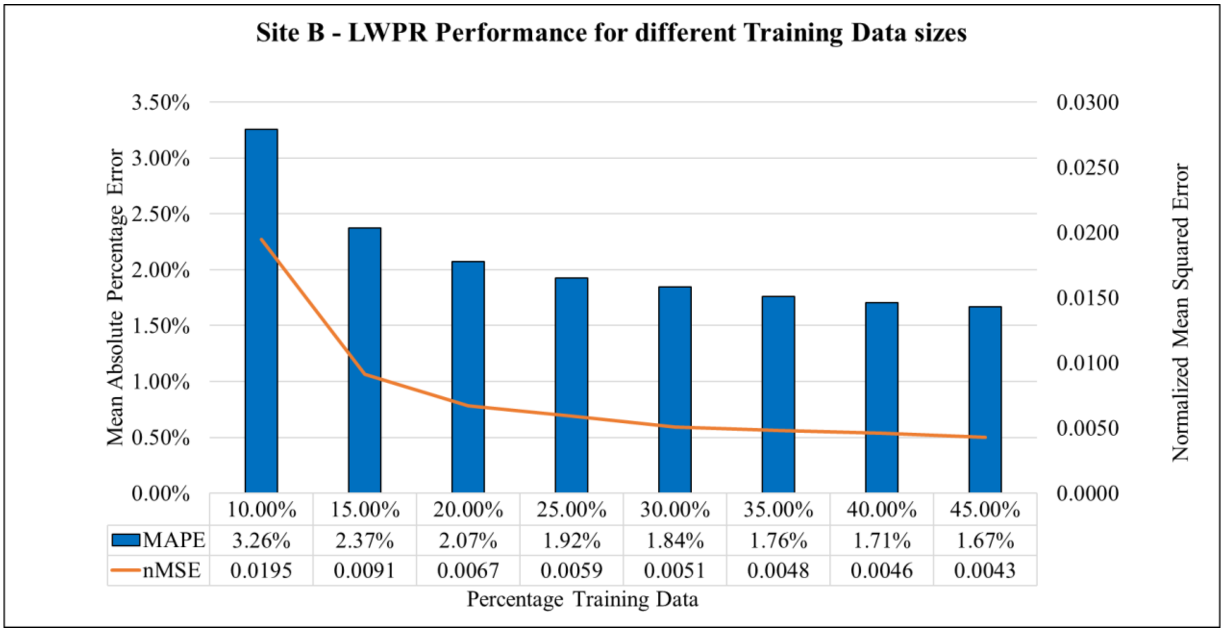
Randomly sampled training datasets of different sizes were evaluated to find the optimal training data sample size for each of the study sites. The model was run 10 times for each training sample size and the Normalized Mean Squared Error (nMSE) and Mean Absolute Percentage Error (MAPE) was averaged across the 10 iterations to produce an average nMSE and average MAPE, as shown in the next two figures. Figure 2-4 shows that for site A, the MAPE does not change significantly between the training sample sizes of 10% and 45%, ranging between 1.67% to 1.81%. The nMSE also does not show significant changes with increased training sample size.



	10.00%	15.00%	20.00%	25.00%	30.00%	35.00%	40.00%	45.00%
MAPE	1.67%	1.62%	1.66%	1.70%	1.76%	1.76%	1.77%	1.81%
nMSE	0.000458	0.000467	0.000533	0.000656	0.000720	0.000794	0.000808	0.000899

Figure 2-4 - Performance of LWPR for Site A averaged across 10 runs of different training data sizes

Figure 2-5 shows the MAPE and nMSE of the LWPR algorithm for Site B, averaged across 10 runs of different training data sizes. It shows that the two metrics decrease significantly with increase in the training data sample size. The MAPE and nMSE starts to stabilize from 3.26% and 0.0195 with 10 % training data to around 1.84% and 0.0051 with 30% training data. The changes in both MAPE and nMSE is quite negligible beyond 30% training data.



	10.00%	15.00%	20.00%	25.00%	30.00%	35.00%	40.00%	45.00%
■MAPE	3.26%	2.37%	2.07%	1.92%	1.84%	1.76%	1.71%	1.67%
—nMSE	0.0195	0.0091	0.0067	0.0059	0.0051	0.0048	0.0046	0.0043

Figure 2-5 - Performance of LWPR for Site B averaged across 10 runs of different training data sizes

2.5.2. Input Strategy - Random Sampling

The performance of each of the 10 runs for 10% randomly sampled training data for Site A is evaluated in Table 2-1. The average percentage error was 1.64% with a deviation of 0.06% between the runs. The 95th percentile absolute error percentage is around 6.43%, while the average standard deviation between the 10 runs is 2.91%. The average run time for model training and prediction is about 4.22 minutes. To assess the solution quality of the predicted search space, the number of actual top 25 solutions present in the predicted top 25 solutions are calculated. For site A, the algorithm is only predicting 4 out of the top 25 solutions on average. This solution quality also changes dramatically between the different runs, as shown by the standard deviation of 4.42.

Table 2-1 - Performance of LWPR on Site A with 10% Randomly Sampled Training Data

Run	**Mean Absolute Percentage Error**	**95th Percentile Absolute Percentage Error**	**Standard Deviation of Absolute Percentage Error**	**Normalized Mean Squared Error**	**Total Run-time (min)**	**Actual Top 25 solutions in Predicted Top 25**
1	1.64%	6.49%	2.68%	0.00046	4.36	2
2	1.72%	6.48%	3.21%	0.00047	4.21	3
3	1.71%	6.69%	3.05%	0.00047	4.07	2
4	1.63%	6.18%	2.97%	0.00049	4.16	2
5	1.62%	6.37%	3.02%	0.00041	4.17	6
6	1.68%	6.53%	2.84%	0.00049	4.26	3
7	1.69%	6.53%	3.33%	0.00045	4.25	4
8	1.62%	6.42%	2.74%	0.00042	4.27	15
9	1.53%	6.09%	2.58%	0.00039	4.25	0
10	1.61%	6.48%	2.65%	0.00048	4.23	5
Average	1.64%	6.43%	2.91%	0.00045	4.22	4.2
Standard Deviation	0.06%	0.18%	0.25%	0.00003	0.08	4.42

A possible reason behind the poor performance of the algorithm for this site may be rooted in the fact that for small systems of signals, the bandwidth does not change significantly between each offset combination. This happens in a system with lower number of intersections due to the presence of slack in the offsets on each intersection. This is shown in Figure 2-6 (a), where the true solution space for bandwidth optimization on site A has flat surfaces, suggesting the presence of several degenerate solutions, all of which are not captured in the estimated solution space. Also, the solution space for offset optimization of a system of 3 intersections can be found in less than 5 minutes using the exhaustive search algorithm, and the use of LWPR did not prove to significantly reduce runtime. LWPR did not provide an accurate enough depiction of the solution space, as shown by the areas of unevenness around the flat surfaces in Figure 2-6 (b).

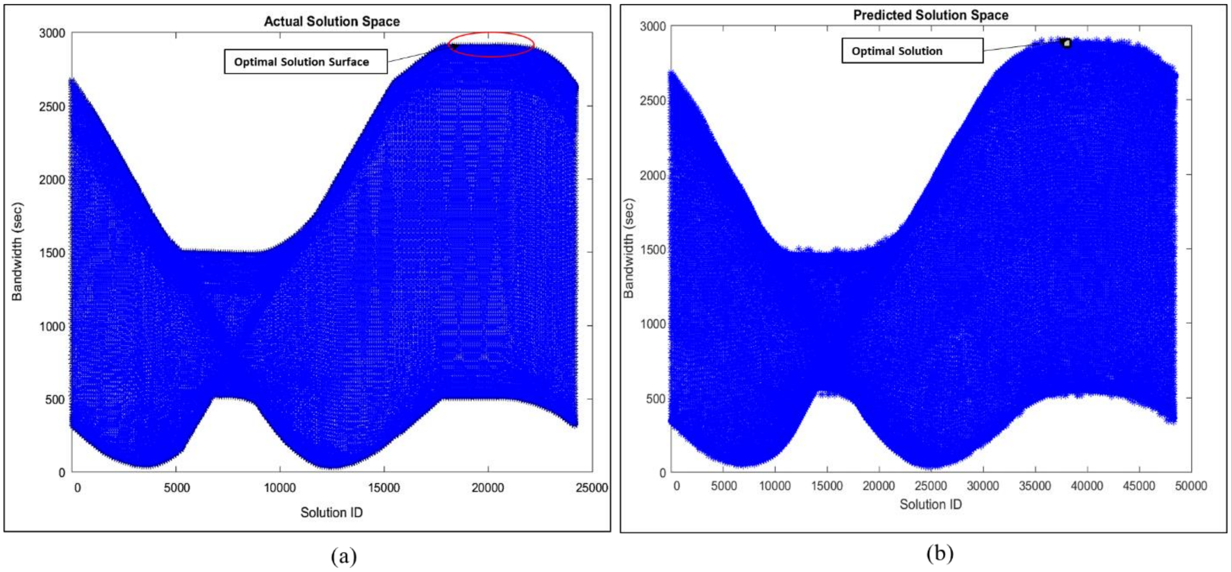


Figure 2-6 - (a) The actual solution space (b) The predicted solution space, with nth offset sampling, over different offset combinations for Site A

Figure 2-7 shows the variation in average and 95th percentile absolute percentage error and the normalized mean squared error between the 10 runs for 10% randomly sampled training data for the site with three intersections. It shows that the average and 95th percentile absolute percentage error did not change significantly between the different runs.

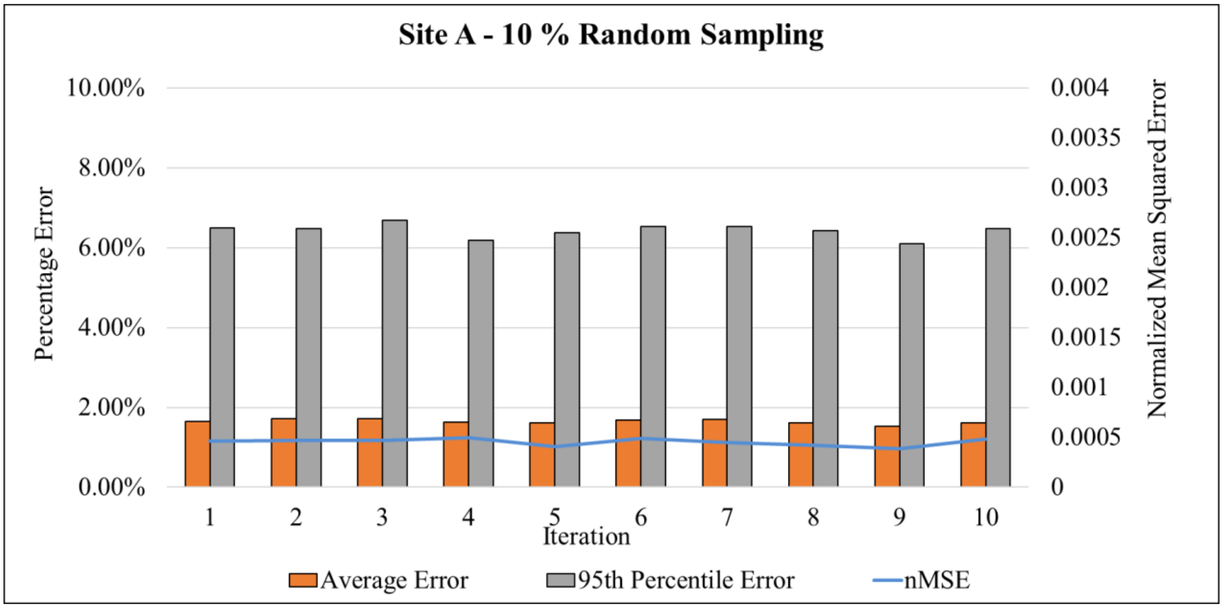


Figure 2-7 - Performance of LWPR algorithm for Site A across 10 runs of 10% randomly sampled training data

The performance of each of the 10 runs for 30% randomly sampled training data for Site B is displayed in Table 2-2.

Table 2-2 - Performance of LWPR on Site B with 30% Randomly Sampled Training Data

Run	**Mean Absolute Percentage Error**	**95th Percentile Absolute Percentage Error**	**Standard Deviation of Absolute Percentage Error**	**Normalized Mean Squared Error**	**Total Run-time (min)**	**Actual Top 25 solutions in Predicted Top 25**
1	1.83%	6.34%	2.49%	0.0051	40.42	19
2	1.77%	6.10%	2.33%	0.005	40.38	16
3	1.80%	6.43%	2.44%	0.0051	41.00	17
4	1.81%	6.33%	2.34%	0.0051	41.15	19
5	1.80%	6.37%	2.40%	0.005	40.69	17
6	1.77%	6.09%	2.37%	0.005	40.30	17
7	1.78%	6.08%	2.41%	0.0048	41.08	22
8	1.84%	6.38%	2.57%	0.0052	40.97	18
9	1.81%	6.37%	2.46%	0.0052	40.66	18
10	1.75%	6.12%	2.35%	0.0049	41.10	19
Average	1.79%	6.26%	2.42%	0.00504	40.78	18.20
Standard Deviation	0.03%	0.14%	0.08%	0.00013	0.33	1.69

The average percentage error was 1.79% with a deviation of only 0.03% between the runs. The 95th percentile absolute error percentage is around 6.26%, while the average standard deviation between the 10 runs is 2.42%. The average run time for model training and prediction is about 40.78 minutes. Looking at the number of actual top 25 solutions present in the predicted top 25 solutions, the algorithm can predict 18 out of the top 25 solutions, on average, which is about 72%. This does not change too much between the different runs, as shown by the standard deviation of 1.69. This indicates that the LWPR algorithm may be a good alternative to explicit enumeration of the offset optimization solution space for a system of signals with 4 intersections.

The information provided in Table 2-2 is also displayed graphically in Figure 2-8 below. There is very little to no variation between the MAPE and nMSE across the 10 different runs using 30% randomly sampled training data.

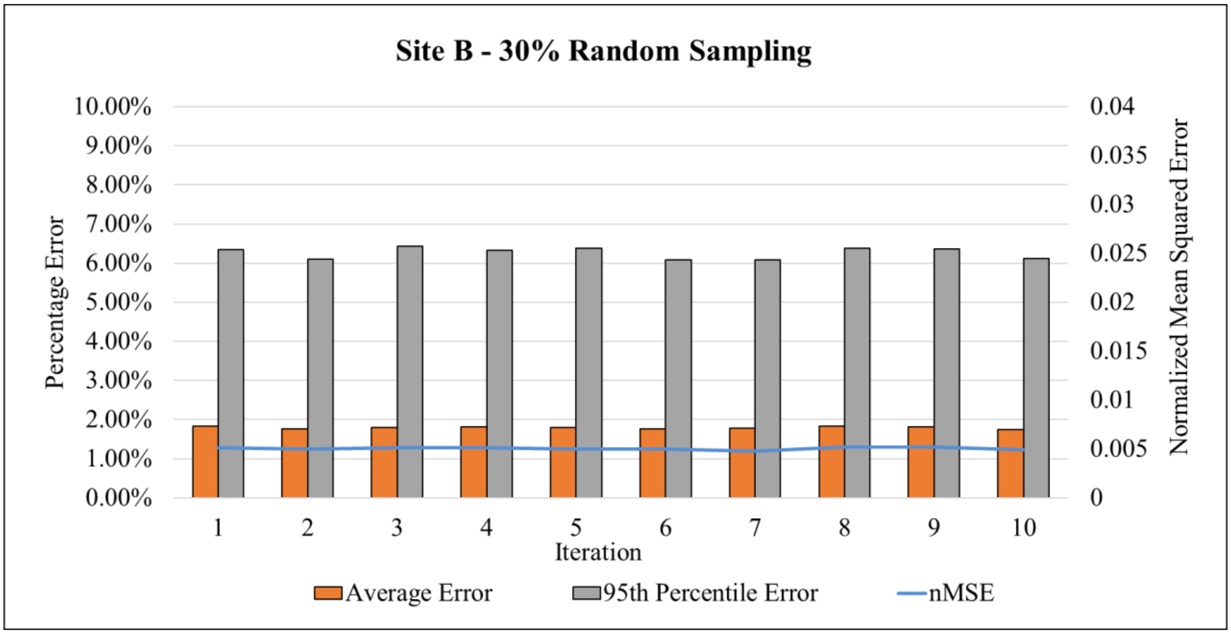


Figure 2-8 - Performance of LWPR algorithm for Site B across 10 runs of 10% randomly sampled training data

2.5.3. Input Strategy - nth Solution Sampling

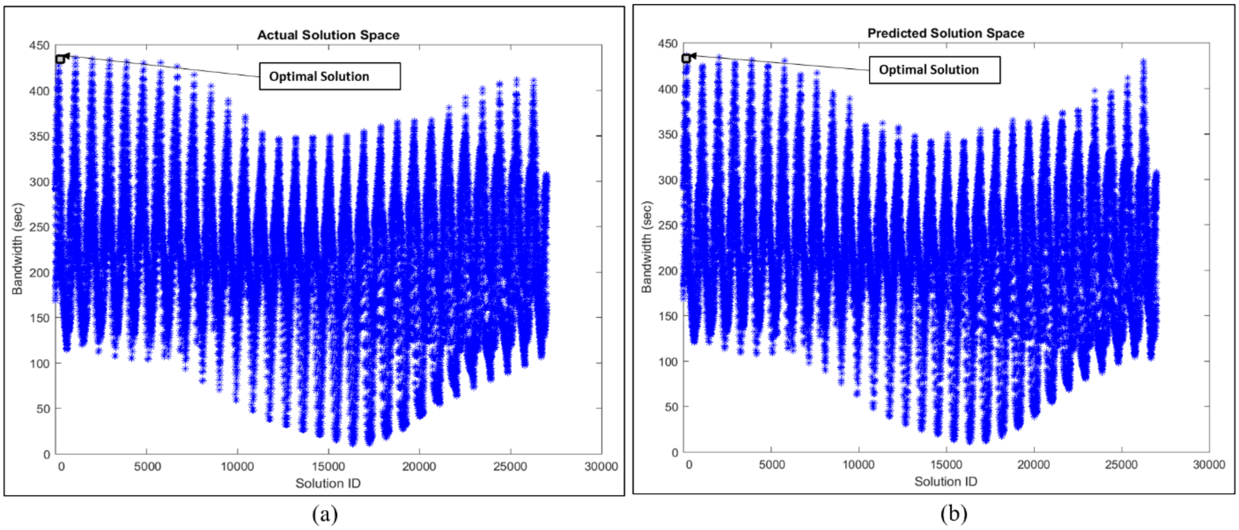
Another form of input training data selection was evaluated for the site with four intersections, i.e., site B. Since the characteristics of the offset optimization solution space can heavily depend on the geometry of the signalized corridor, we sampled every nth offset combination input-output pairs to form the training data. The training data for site B was sampled for every 2 seconds and two iterations of LWPR were run, each corresponding to a different starting point of the nth offset sampling method. This was done to train the model with some more knowledge about the pattern of the entire solution space. This training data selection strategy yielded a sample size of 3375 data points or 12.5%.

The results of these runs are summarized in Table 2-3 below. It shows that the average MAPE across the two runs is 1.85% and the 95th percentile absolute percentage error is about 7%. The standard deviation in error in prediction is about 2.7%, whereas the nMSE is around 0.0055. One thing to be noted is that the total run-time is around 30 minutes, which is 10 minutes less than the run-time using 30% randomly sampled data. This is because this method only uses about 12.5% training data. The nth offset sampling method yielded accuracy similar to the random sampling technique, in terms of predicted the top 25 actual solutions, about 18 solutions or 72%.

Table 2-3 - Performance of LWPR on Site B with nth Offset Sampling Strategy

Run	Mean Absolute Percentage Error	95th Percentile Absolute Percentage Error	Standard Deviation of Absolute Percentage Error	Normalized Mean Squared Error	Total Run-time (min)	Actual Top 25 solutions in Predicted Top 25
1	1.84%	6.80%	2.67%	0.0055	30.76	18
2	1.86%	7.20%	2.73%	0.0051	30.33	18
Average	1.85%	7.00%	2.70%	0.0053	30.54	18
Std Deviation	0.01%	0.28%	0.04%	0.00028	0.30	0

This shows that with only 12.5% training data, the nth offset sampling was able to estimate the search solution space with the same accuracy as 30% randomly sampled training data and with a run-time 10 minutes less. The actual solution space and the estimated solution space, along with the optimal solution are displayed in Figure 2-9, for the first run of the nth offset sampling technique. LWPR is able to predict the solution space, closely resembling the actual solution space.



(a)

(b)

Figure 2-9 - (a) The actual solution space (b) The predicted solution space, with nth offset sampling, over different offset combinations for Site B

2.6. Conclusions and Future Work

This paper showed investigative efforts in finding a method alternative to Exhaustive Search for the dynamic bandwidth optimization problem. The main idea was to develop a method that takes a sample output from the exhaustive search as input to train a Machine Learning model and estimate the output for the remaining candidate solutions. The Machine Learning algorithm selected for this purpose was Locally Weighted Projection Regression (LWPR) and its ability to estimate the entire search space of the combinatorial problem was tested on two datasets.

From the numerical evaluations, it was seen that the LWPR worked well in terms of average error in prediction, with 10% random sampled training data for the 3-intersection site. However, it did not seem to find the global optimal solution region effectively for the 3-intersection site, which may be to the presence of many degenerate solutions and slack in the offsets on a system with lesser number of intersections. For larger sites with more than 3 or 4 intersections, we expect the presence of flat surfaces in the solution space to diminish. Since the method did not show a major improvement in runtime for this site either, this leads to the conclusion that for smaller systems of signals, e.g., with 3 intersections, exhaustive search is probably a better technique.

For the 4-intersection site, the LWPR method performed well and could predict the global top 25 solutions with an accuracy of 72%. However, to achieve that accuracy in estimating the solution search space, 30% of randomly sampled input-output pairs was used as the training data set. With an increase in dimensionality from the other study site and an increase in training data sample size, the run-time of the LWPR algorithm for this site was about 40 minutes.

Another training data selection strategy was also evaluated, which involved sampling every nth offset combination input-output pairs to constitute the training dataset. The results from this type of selection technique was comparable to the random sampling technique. One may argue that this may introduce bias into the process of selecting the training dataset. However, for the problem of dynamic bandwidth optimization, this technique can provide the model with more information about the solution space, enabling it to provide a higher accuracy with lesser percentage of training data. This technique may prove to be more computationally effective when systems with more than 4 signals are considered. The more the number of intersections, the more complex the solution space may get. Hence, providing more information to train the model can help improve accuracy as well as run-time.

The computational complexity of LWPR is O(n), i.e., it is linear with the number of inputs. For the 4-intersection site, the total number of solutions was 729,000 and MATLAB runs out of memory while performing the prediction on the entire dataset. Due to lack of resources like parallel computing, the LWPR technique was unable to be tested on larger datasets. However, it can be argued that with unlimited memory and computing capability, the LWPR method could be well-suited for the process of estimating the entire solution space, given only a small percentage of data for training. Use of parallel computing for implementing the testing phase of the algorithm could speed up this process even further. With better hardware and software configuration, future research is required to determine the accuracy of the nth offset sampling technique, by testing more sites with larger number of intersections. Moreover, the findings from this research reinforces the need for a better method of finding the optimal solution for maximizing the dynamic bandwidth on coordinated arterials.

2.7. References

Agarwal, M., Goyal, M., & Deo, M. C. (2010). Locally weighted projection regression for predicting hydraulic parameters. Civil Engineering and Environmental Systems, 27(1), 71-80.

Atkeson, C. G., & Schaal, S. (1995). Memory-based neural networks for robot learning. Neurocomputing, 9(3), 243-269.

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). *Locally weighted learning for control*. Artificial Intelligence Review, 11(1), 75-113. doi:10.1023/A:1006511328852

Basher, S., Purushothaman, S., & Rajeswari, P. (2015). *Vehicle tracking using locally weighted projection regression method*. International Journal of Computer Science and Information Security, 13(9), 83-89.

Chang, C., & Lin, C. (2011). Libsvm. TIST ACM Transactions on Intelligent Systems and Technology ACM Trans. Intell. Syst. Technol., 2(3), 1-27.

Christy, C. D., & Dyer, S. A. (2006, April). Estimation of soil properties using a combination of spectral and scalar sensor data. In Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE (pp. 729-734). IEEE.

Day, C. M., & Bullock, D. M. (2011). Computational Efficiency of Alternative Algorithms for Arterial Offset Optimization. Transportation Research Record, 2259(1), 37–47. https://doi.org/10.3141/2259-04

Day, C. M., & Bullock, D. M. (2011). Computational Efficiency of Alternative Algorithms for

Arterial Offset Optimization. Transportation Research Record, 2259(1), 37–47. https://doi.org/10.3141/2259-04

de Julian-Ortiz, J., Pogliani, L., & Besalu, E. (2010). Two-variable linear regression: Modeling with orthogonal least-squares analysis. Journal of Chemical Education, 87(9), 994-995. doi:10.1021/ed100307z

Gettman, D., Shelby, S. G., Head, L., Bullock, D. M., & Soyke, N. (2007). Data-Driven Algorithms for Real-Time Adaptive Tuning of Offsets in Coordinated Traffic Signal Systems. Transportation Research Record, 2035(1), 1–9. https://doi.org/10.3141/2035-01

Gettman, D., Shelby, S. G., Head, L., Bullock, D. M., & Soyke, N. (2007). Data-Driven Algorithms for Real-Time Adaptive Tuning of Offsets in Coordinated Traffic Signal Systems. Transportation Research Record, 2035(1), 1–9. https://doi.org/10.3141/2035-01

Kim, S., Hajbabaie, A., Williams, B. M., & Rouphail, N. M. (2016). Dynamic Bandwidth Analysis for Coordinated Arterial Streets. *Journal of Intelligent Transportation Systems*, *20*(3), 294-310.

Nguyen-Tuong, D., Seeger, M., & Peters, J. (2009). Model Learning with Local Gaussian Process Regression. Advanced Robotics, 23(15), 2015-2034. doi:10.1163/016918609X12529286896877

Park, H., Kim, N., & Lee, J. (2014). Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over KOSPI 200 index options. Expert Systems with Applications, 41(11), 5227-5237. doi:10.1016/j.eswa.2014.01.032

Pontes, J., & Santos, C. P. (2012). *Velocity control of a two DOF walking system*. AIP Conference Proceedings, 1479(1) doi:10.1063/1.4756177

Vijayakumar, S., D'souza, A., & Schaal, S. (2005). *Incremental Online Learning in High Dimensions.* Neural Computation, 17(12), 2602-2634.

Vink, J. P., & Haan, G. D. (2012). Comparison of machine learning techniques for target detection. Artificial Intelligence Review Artif Intell Rev, 43(1), 125-139.

Williams, C. K. I., & Rasmussen, C. (1996). Gaussian processes for regression. In M. Touretzky & Hasselmo (Eds.), *Advances in neural information processing systems*, 8. Cambridge, MA: MIT Press.

3. A MIXED INTEGER LINEAR PROGRAM TO OPTIMIZE DYNAMIC BANDWIDTH ON SIGNALIZED COORDINATED ARTERIAL STREETS

3.1. Introduction

Traffic signals aim at improving overall flow of traffic and increase safety at busy intersections. In a typical urban area, they might direct the movement of as many as 100,000 vehicles per day [1]. The benefits of well-timed traffic signals are manifold and can also have positive impacts on road safety and fuel emissions. To reduce traffic delay and fuel consumption, signalized intersections that are close to one another are often coordinated to permit platoons of vehicles to exit through a corridor smoothly and assuring continuous flow of the associated movements at a planned speed on the major street. This functionality can profoundly enhance the quality of service of the facility.

The signals in a coordinated system typically operate by using pre-timed or coordinated-actuated control, and the coordinated phases typically serve the through movements on the major street. While developing well-timed signal timing plans for coordinated corridors, various objective functions and decision variables may be considered. One of the more common objectives used for such purpose is maximizing the arterial bandwidth. Arterial bandwidth is defined as the total amount of time per cycle available for vehicles to travel through a system of coordinated intersections at the given progression speed. It is also defined as the time difference between the first and last hypothetical trajectory that can travel through the entire arterial at the progression speed without stopping. A research study conducted by Yang showed that bandwidth-based signal timing plans generally outperform delay-based solutions based on several field studies [2]. Bandwidth is also a very intuitive objective, since daily commuters can relate easily to the frustration stemming from having to stop at traffic lights which are close to each other. A common decision variable associated with maximization of arterial bandwidths is the offset at each intersection. Offset is defined as the time from a reference point, such as the start of green or yellow of the coordinated phase at an intersection, to the same reference point at the other intersections. Offset reference points help structure the relationship among coordinated intersections by defining the point in time when the cycle begins timing. Offset optimization is the science of selecting offsets for a corridor of signalized intersections to allow platoons of vehicles to move through the corridor without stopping at red lights.

Much research, as will be summarized in the literature review, has focused on developing mathematical models to maximize bandwidth by computing optimal parameters of traffic signal controllers, such as cycle length, start of green, and offsets between intersections. However, most traditional methods related to offset optimization have assumed fixed green times at each intersection as a basis for calculating the bandwidth, and hence, a repeating traffic pattern where no movements are skipped. However, most current traffic signals operate on a semi-actuated mode of control, using detection only for the minor movements at an intersection. Under this type of control, vehicle actuation on uncoordinated phases may cause the phase to skip or gap-out, making it end earlier and allocating the remaining green time to the coordinated phases on the major streets. This makes the green indication for the coordinated phases start earlier than planned, and this phenomenon is called ‘early return to green’ [1]. This necessitates the development of optimization techniques which account for the green times and resulting bandwidth which differ between each signal cycle. Study on dynamic bandwidth analysis has confirmed that programmed green bandwidth consistently underestimates the size of the actual dynamic bandwidth, which renders plans built on top of programmed bandwidth analysis to be potentially sub optimal [3].

The cycle-by-cycle duration of green on coordinated phases, found from signal controller split monitor data, gives an indication of how much green time is required for the major coordinated phases. This information can serve as an input to new optimization models to maximize dynamic bandwidth. This paper presents a mixed integer linear programming model formulation to solve the problem of maximizing dynamic bandwidth, which uses archived signal controller split monitor data as input to define the constraints. This accuracy of the proposed method is first evaluated using two hypothetical system of signals and programmed green time data. Then, the model is evaluated for a real-world study site, with actual field observed signal log data and the results are validated against the results from the exhaustive search technique [3]. Finally, conclusions and future work required to make this model more robust and adaptable are discussed.

3.2. Literature Review

Wu et al. broadly classified existing traffic signal optimization approaches into two categories: those that minimize delay and stops such as TRANSYT-7F and SIGOP, and those that maximize arterial progression bandwidth such as PASSER II and MAXBAND [5]. MAXBAND and PASSER II are two well-known software tools that are used widely to optimize programmed bandwidth [6,7]. Other techniques for programmed bandwidth maximization are discussed below.

TRANSYT-7F uses a traffic model which considers the flow interaction between successive road sections, platoon dispersion, etc. Good convergence on the optimum signal settings is achieved by using a hill climbing type of optimization procedure that minimizes a chosen balance between total delay and the number of stops [8]. The SIGOP II signal timing optimization procedure consists of a flow model and an optimization methodology, with an objective to minimize vehicle delay, stops, and excess queue length. The optimization procedure uses the method of successive approximations within the framework of a dynamic programming methodology [9].

The algorithm developed by Brooks and Little, for two-phase signals, established the primary principles of bandwidth optimization, which have since been adopted by most bandwidth-based software packages [10,6]. For example, Messer et al. enhanced the original algorithm to handle multiphase signals, by developing Progression Analysis and Signal System Evaluation Routine (PASSER II), a macroscopic deterministic optimization model [7]. The phase sequence and cycle length that provide maximum two-way progression for a specified arterial signal system is determined using an iterative gradient search method. It uses Brook's Interference Algorithm and Little's Optimized Unequal Bandwidth Equation for bandwidth maximization [10,11].

Morgan and Little were the first to suggest a mathematical formulation [11]. Provided the cycle length, signal red times as a percentage of the cycle length, the signal positions or distance between intersections and progression speeds, they found that there is a half-integer synchronization that gives maximal equal bandwidths. Later, Little set up the formulation of the maximal programmed bandwidth problem as a mixed-integer linear program, thus opening the possibility of introducing new decision variables and constraints in determining a global optimal solution [12].

Little et al. developed a more versatile method called MAXBAND, in which they extended the mixed-integer formulation to multiphase signals, and included cycle time, offsets, progression speeds, and order of left-turn phases, as decision variables [6]. Following that, Gartner et al., still using mixed-integer linear programming, improved MAXBAND by developing MULTIBAND [13]. This new method incorporated a systematic traffic-dependent criterion which provides a capability to adapt the progression scheme to the specific traffic flow pattern on each link of the artery. The MAXBAND model was improved years later by Kim, who proved that the model is over constrained and can provide sub-optimal solutions in some cases [4].

Serafani and Ukovich also formulated a mathematical model that dealt with the fixed-time traffic control problem of finding a periodic schedule for traffic signal control [14]. Girianna and Benekogal used Genetic Algorithms (GA) to dynamically optimize signal-timing parameters in a discrete-time signal coordination problem [15]. Tian et al. proposed a new heuristic approach to application of bandwidth-oriented signal timing based on a system partition technique, which they said could be easily used by software packages like PASSER II and Synchro. Yin et al. presented a new offline offset refiner, which addresses the problem of uncertain starts and ends of green and used a large amount of archived signal status data available from real-time signal operations, to fine tune the signal offsets to provide smoother progression in either one-way or two-way coordination [16].

Wu et al. proposed a bandwidth optimization method, including detail steps to calculate upper/lower influences and relative offset for intersections, based on a group partition method for coordinated arterials [5]. They concluded that the signal timing plan from improved bandwidth optimization and group partition method is much better than the result from Synchro 6.0 optimization function and Messer’s method.

Buckholz indicated that coordinated signal systems do not perform very well in certain conditions, including skipped phases that cause early return to green on the coordinated phases resulting in a disruption on the arterial progression [17]. For minimizing the early return to green effect on coordinated arterials, Skabardonis describes the procedures developed for translating fixed-time timings to actuated controller settings for arterials and grid networks and summarizes the results from their application on 14 real-world data sets that can be applied to the MAXBAND, PASSER, and TRANSYT-7F timing programs [18]. The performance of the resulting timing plans was evaluated using the NETSIM simulation model. Similar study was conducted by Chang to effectively assist users in selecting the proper controller-detector combinations, improving system detector locations and optimizing actuated timing parameters in actuated control operation of arterial signal systems [19].

Beard & Ziliaskopoulos proposed a mixed-integer linear programming formulation to solve the combined system optimal dynamic traffic assignment and signal optimization problem, using a cell transmission model to model the traffic conditions [20]. The formulation is suited to respond to oversaturated traffic conditions and can be adapted to account for turning movements, protected and permissive phases (gap acceptance), and multiple signal controller types: dynamic (traffic adaptive) and pre-timed.

Peifeng et al. developed a simple algorithm to maximize variable bandwidth progression, by adjusting the offset and phase sequence pattern for each signalized intersection [21]. Their arterial coordination model involved evaluating a 0-1 comprehensive search function that determines whether the signal light is available for a particular through movement at the given time period, but their algorithm showed limitations in terms of computational complexity.

Liu & Xu proposed an approach based on differential evolution bacteria foraging algorithm, in which not only the phases which provide right of way to the coordinated directions but the phases which provide right of way to the uncoordinated directions are considered [22]. Shirvani & Maleki developed a new mixed integer variable bandwidth optimization model, which uses Bellman–Zade principles of fuzzy decision making [23]. It calculated a fuzzy membership function for each bandwidth as well as each green split and simultaneously optimizes green splits, offsets, cycle length, and left turn phase sequence patterns.

Arrival on green or percentage of vehicles arriving during green time has also been used to maximize traffic flow on coordinated arterials. Purdue University and the Indiana DOT developed the Purdue Coordination Diagram, which uses high-resolution vehicle event data to optimize cycle length and vehicle arrival percentage during green time [24]. In addition, this system uses first vehicle estimated trajectory to adjust offset to improve percentage of vehicle arriving on green. University of Minnesota developed a real-time arterial performance monitoring system called SMART-SIGNAL, which also uses hi-resolution data to calculate vehicle arrival type and the proportion of vehicle arriving during green phase [25]. This system can adjust offsets using virtual probe trajectories. Both new signal systems require expensive 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.

Kim developed an exhaustive search algorithm that enumerates all possible offset combinations and calculates the two-way bandwidth by counting the number of seconds that would accommodate a through-vehicle trajectory traveling at a specified progression speed without encountering a red indication at any of the intersections [3]. A linear programming method was also proposed, as opposed to the traditional mixed-integer programming formulations to optimize the two-way variable bandwidth on major arterials [4]. The methodology used archived signal log data regarding phase start and end times in each cycle in both directions. But this method cannot be extended to systems with larger number of intersections as the absence of an integer variable does not provide it the capability to identify which input cycle to build a band in.

3.3. Methodology

The main idea behind the development of the MILP to build directional green bands for the major street approached, by constraining the start and end of each directional band inside of the green durations for each cycle. The green duration for the major street approaches for each cycle includes the allotted time and extra time that is donated from skipped or early termination of the minor street phases. After defining the observed start and end time at each intersection, the mathematical model searches for the optimal offset combination, which maximizes the sum of directional bandwidth.

In this section, the split monitor data used as input for the mathematical model is described and the mathematical formulation of the new Mixed Integer Linear Program (MILP) is presented.

3.3.1. Signal Controller Split Monitor Data

The used green time duration for the major approaches (phases 2 and 6), including the allotted time and the extra time donated by the minor phases are gathered from signal controller split monitor data. Raw status entries for each cycle, regarding the number of seconds of green time allocated to each phase or movement at each intersection for each cycle, is logged by the signal controller. These logs, known as the split monitor data, are stored in the non-volatile RAM memory and can be cleared upon upload from a central computer. The split monitor data reports the following: “Time Stamp”, “Cycle Length”, “Offset”, “Plan”, “Extra Time”, “Used Phase”, etc. This input table is shown for the required phases for a single intersection in Table 3-1 below. Using the timestamp as start of each cycle, the extra time is added to the used time for the phases 2 and 6 to create actual time stamps of start and end of the green indication for each intersection.

This data forms the input for the mixed integer linear programming model discussed below.

Table 3-1 - Sample signal controller split monitor log data

Time Stamp	Cycle (sec)	Offset (sec)	Plan	ExtraTimeCP1 (sec)	ExtraTimeCP2 (sec)	UsedPhase2 (sec)	UsedPhase6 (sec)
4/2/18 16:06	110	75	24	1	2	26	43
4/2/18 16:07	110	75	24	16	0	29	28
4/2/18 16:09	110	75	24	0	0	26	26
4/2/18 16:11	110	75	24	0	0	45	26
4/2/18 16:12	110	75	24	0	0	29	29
4/2/18 16:14	110	75	24	2	0	30	30
4/2/18 16:16	110	75	24	0	0	30	38
4/2/18 16:18	110	75	24	1	0	30	52
4/2/18 16:20	110	75	24	3	0	30	30
4/2/18 16:22	110	75	24	19	1	30	39

3.3.2. Mixed Integer Linear Programming Model Formulation

The key structure of the Mixed Integer Linear Programming model presented in this paper is based on many of the same constraints explained in Kim's paper [4].

The objective function of the linear programming model is to maximize the sum of two-way bandwidth and includes a fractional value (f) between 0 and 1, that is used to represent the distribution of traffic in the two directions.

Let I, J and K respectively denote the set of all intersections in the system off signals, all directional bands or cycles included in the study period, and all directions of traffic flow.

$$\max \sum_{\forall\, j \in J} \left(b_j + f\bar{b}_j\right) \tag{3.1}$$

where,

$b_j = \; j^{th}$ bandwidth in the outbound direction $(k = 1)$,

$\bar{b}_j = \; j^{th}$ bandwidth in the inbound direction $(k = 2)$,

f = ratio of traffic demand in the outbound to the inbound direction.

The outbound and inbound band sizes are computed by subtracting the start time $(x_{i,j,k})$

of the corresponding band from its end time ($y_{i,j,k}$) as follows:

$$b_j = y_{i,j,1} - x_{i,j,1} \qquad \forall i \in I, \forall\, j \in J \tag{3.2}$$

$$\bar{b}_j = y_{i,j,2} - x_{i,j,2} \qquad \forall i \in I, \forall\, j \in J \tag{3.3}$$

$$y_{i,j,k} \geq x_{i,j,k} \qquad \forall\, i \in I, \forall j \in J, \forall\, k \in K \tag{3.4}$$

where,

$x_{i,j,k}$ = continuous variable to calculate the start of band at intersection i and cycle j in direction k,

$y_{i,j,k}$ = continuous variable to calculate the end of band at intersection i and cycle j in direction k.

The offset at intersection i, O_i, is constrained to be between $-C$ and C.

$$-\mathrm{C} \leq O_i \leq C \qquad \forall\, i \in I \tag{3.5}$$

where C = common cycle length.

To compare the results of the MILP with that from other techniques, O_i was configured as an integer variable in this chapter. However, it is not required to be an integer variable in general, especially since having this as a continuous variable can reduce computational time with growth in problem size.

The travel time between intersections $i \in I$ and $z \in Z_{i,k}$, where is defined by $Z_{i,k}$, where $Z_{i,k}$ denotes the set of intersections downstream from intersection 1 to i in k direction. 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 below:

$$x_{i,j,k} = x_{1,j,k} + \sum\nolimits_{\forall\, z \in Z_{i,k}} T_{i,z} \qquad \forall\, i \in I,\ \forall j \in J,\ \forall k \in K \tag{3.6}$$

$$y_{i,j,k} = y_{1,j,k} + \sum\nolimits_{\forall\, z \in Z_{i,k}} T_{i,z} \qquad \forall\, i \in I,\ \forall j \in J,\ \forall k \in K \tag{3.7}$$

Figure 3-1 below shows the geometry of the LP for a system of three signals and how the bandwidths between each cycle are dynamic.

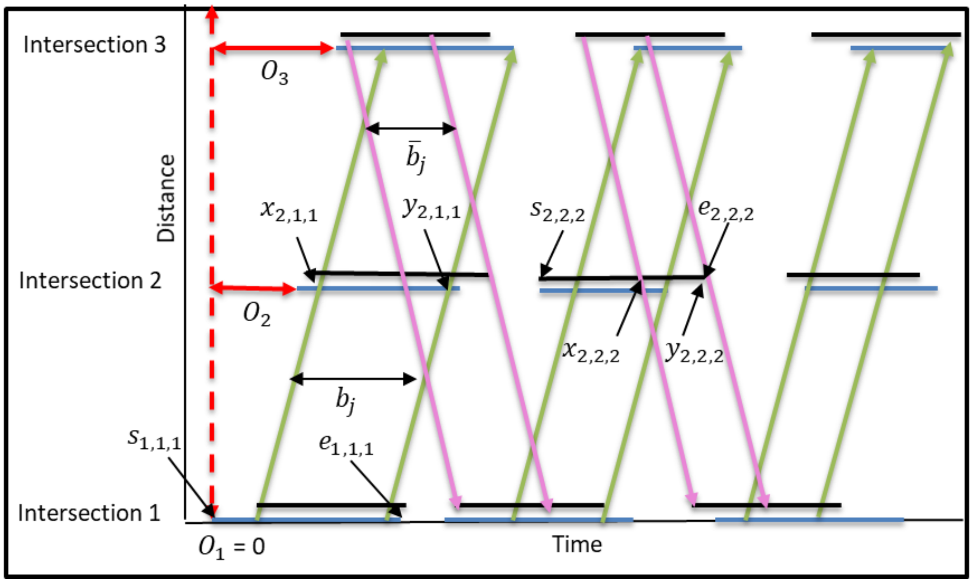


Figure 3-1 - Geometry of Dynamic Bandwidth Optimization

The start ($s_{i,j,k}$) and end ($e_{i,j,k}$) of the green times for intersection i, cycle j and direction k from the signal controller log data serve as the input. The final set of constraints in this model is used to restrict the start and end of the green bands within the start and end of cycle-by-cycle green time from the input data. This is where the integer binary variables are finally introduced. When the number of intersections to be analyzed increases, instead of building a green band in every cycle, the model requires the flexibility to choose which cycle to build a band in. This is achieved by using the concept of one out of 'n' constraints, in which a set of binary variables are used to indicate whether a constraint holds, by multiplying it to a big number M. M can be easily calculated from the input data as the maximum start or end of green times in both direction of travel. A set of three binary variables are introduced for each band 'j' at intersection 'i' and in direction ‘k’. They are configured slightly differently, however, for each direction of travel, as described below.

The binary variables are defined as follows:

$$L_{1ijk} = \begin{cases} 1, if\ band\ j\ at\ intersection\ i\ is\ formed\ in\ cycle\ (j-1)\ in\ direction\ 'k' \\ 0, otherwise \end{cases}$$

$$L_{2ijk} = \begin{cases} 1, if\ band\ j\ at\ intersection\ i\ is\ formed\ in\ cycle\ j\ in\ direction\ 'k' \\ 0, otherwise \end{cases}$$

$$L_{3ijk} = \begin{cases} 1, if\ band\ j\ at\ intersection\ i\ is\ formed\ in\ cycle\ (j+1)\ in\ direction\ 'k' \\ 0, otherwise \end{cases}$$

The constraints that ensure that the start and end of bandwidths are within the start and end of green times for the first intersections in each direction of travel are as follows:

$$s_{1,j,\mathrm{k}} + O_1 \ \leq x_{1,j,\mathrm{k}} \leq e_{1,j,\mathrm{k}} + O_1,\ \forall\, j = 1\ to\, J, k = 1\ to\ 2 \qquad (3.8)$$

$$s_{1,j,1} + O_1 \ \leq y_{1,j,1} \leq e_{1,j,1} + O_1,\ \forall\, j = 1\ to\, J, k = 1\ to\ 2 \qquad (3.9)$$

For the intersection $i = 2\ to\ I$, the new constraints that search the cycle prior to the current cycle to build a band ‘j’ are defined by the following equations:

$$s_{i,j-1,\mathrm{k}} + O_i - M\, L_{1,i,j,k} \ \leq x_{i,j,\mathrm{k}} \leq e_{i,j-1,\mathrm{k}} + O_i + M\, L_{1,i,j,k},\ \forall\, i = 2\ to\ I, j = 1\ to\, J, k = 1\ to\ 2 \qquad (3.10)$$

$$s_{i,j-1,\mathrm{k}} + O_i - M\, L_{1,i,j,k} \ \leq y_{i,j,\mathrm{k}} \leq e_{i,j-1,\mathrm{k}} + O_i + M\, L_{1,i,j,k},\ \forall\, i = 2\ to\ I, j = 1\ to\, J, k = 1\ to\ 2 \qquad (3.11)$$

For the intersection $i = 2\ to\ I$, the new constraints that search the current cycle to build a band ‘j’ are defined by the following equations:

$$s_{i,j,\mathrm{k}} + O_i - M\, L_{2,i,j,k} \ \leq x_{i,j,\mathrm{k}} \leq e_{i,j,\mathrm{k}} + O_i + M\, L_{2,i,j,k},\ \forall\, i = 2\ to\ I, j = 1\ to\, J, k = 1\ to\ 2 \qquad (3.12)$$

$$s_{i,j,\mathrm{k}} + O_i - M\,L_{2,i,j,k} \le y_{i,j,\mathrm{k}} \le e_{i,j,\mathrm{k}} + O_i + M\,L_{2,i,j,k}, \quad \forall\, i = 2\ to\ I, j = 1\ to\ J, k = 1\ to\ 2 \tag{3.13}$$

For the intersection $i = 2\ to\ I$, the new constraints that search the cycle after the current cycle to build a band 'j' are defined by the following equations:

$$s_{i,j+1,\mathrm{k}} + O_i - M\,L_{3,i,j,k} \le x_{i,j,\mathrm{k}} \le e_{i,j+1,\mathrm{k}} + O_i + M\,L_{3,i,j,k}, \quad \forall\, i = 2\ to\ I, j = 1\ to\ J, k = 1\ to\ 2 \tag{3.14}$$

$$s_{i,j+1,\mathrm{k}} + O_i - M\,L_{3,i,j,k} \le y_{i,j,\mathrm{k}} \le e_{i,j+1,\mathrm{k}} + O_i + M\,L_{3,i,j,k}, \quad \forall\, i = 2\ to\ I, j = 1\ to\ J, k = 1\ to\ 2 \tag{3.15}$$

Finally, to ensure only one of the three constraints hold, the sum of the three binary variables for each band 'j' at intersection 'i' and in direction 'k' are set equal to 2.

$$\sum_{t=1}^{3} L_{t,i,j,k} = 2, \qquad \forall\, i \in I,\ \forall j \in J,\ \forall k \in K \tag{3.16}$$

3.4. Numerical Evaluations

The new model is tested for accuracy and robustness, first using a hypothetical simple pre-timed system of three signals and then using another hypothetical 6-intersection system of signals with programmed green times. Finally, the model is tested on a real-world three-intersection site, in Clayton, North Carolina, using archived signal log data as the input to the model. The results of the MILP are compared against the results of the exhaustive search.

3.4.1. Three-Intersection Hypothetical Site – Programmed Bandwidth

To test the correctness of the new MILP, it was evaluated for one of the hypothetical scenarios presented in the paper providing an enhanced MAXBAND formulation by Kim [4]. The scenario considers three coordinated signalized intersection, with a common cycle length of 100 seconds. Adjacent intersections were at 825 feet from each other and the progression speed considered is 45 mph or 66 ft/sec. The effective red time at each intersection is 50 seconds in both directions. To calculate the fraction of distribution of bands between the intersections, a total demand of 1,000 vehicles per hour per lane (vphpl) in both directions was considered and the models were evaluated for inbound to outbound demand proportion varying from 9:1 to 1:9. The results of these evaluations are displayed in Table 3-2. The results of the MILP match exactly the solutions given by the enhanced MAXBAND model, demonstrating the validity correctness of solutions of the model.

Table 3-2 - MILP Optimal Solutions compared against Enhanced MAXBAND for 3-Intersection Hypothetical Test Scenario

DEMAND (vphpl)			**ENHANCED MAXBAND SOLUTIONS**						**MILP SOLUTIONS**					
			Offsets(sec)			**Bandwidths(sec)**			**Offsets(sec)**			**Bandwidths(sec)**		
OUT	**IN**	**f**	**INT1**	**INT2**	**INT3**	**OUT**	**IN**	**TOTAL**	**INT1**	**INT2**	**INT3**	**OUT**	**IN**	**TOTAL**
100	900	9.00	0	-7.5	-20	5	45	50	0	-7.5	-20	5	45	50
200	800	4.00	0	-2.5	-15	10	40	50	0	-2.5	-15	10	40	50
300	700	2.33	0	2.5	-10	15	35	50	0	2.5	-10	15	35	50
400	600	1.50	0	7.5	-5	20	30	50	0	7.5	-5	20	30	50
500	500	1.00	0	12.5	0	25	25	50	0	12.5	0	25	25	50
600	400	0.67	0	12.5	5	30	20	50	0	12.5	5	30	20	50
700	300	0.43	0	12.5	10	35	15	50	0	12.5	10	35	15	50
800	200	0.25	0	12.5	15	40	10	50	0	12.5	15	40	10	50
900	100	0.11	0	12.5	20	45	5	50	0	12.5	20	45	5	50

The time-space diagram for three inbound and outbound bands for the hypothetical test scenario for a demand proportion equal to 1 is shown in Figure 3-2.

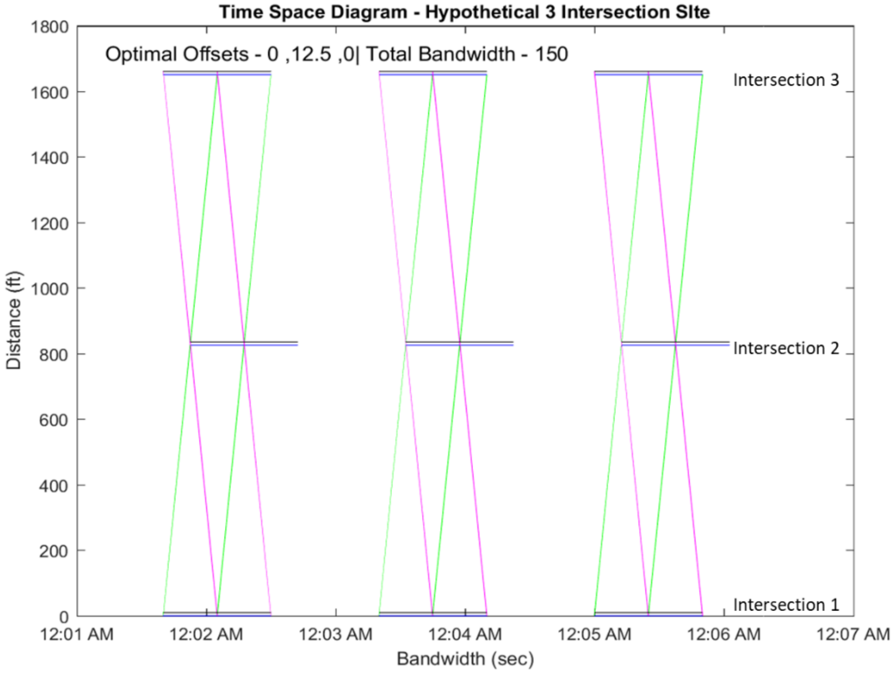


Figure 3-2 - Time Space diagram showing three bi-directional bandwidths for 3-intersection hypothetical site (f=1)

3.4.2. Six-Intersection Hypothetical Site – Programmed Bandwidth

To test the robustness of the new MILP, it was evaluated for a six-intersection hypothetical site. The scenario considers six coordinated signalized intersection, each 600 feet apart and with a common cycle length of 90 seconds. Adjacent intersections were at 300 feet from each other and the progression speed considered is 60 ft/sec. The green to red split at each intersection is 70/30 percent or 60/30 sec in both directions. The new MILP model was evaluated for inbound to outbound demand proportion of 1, i.e., equal bandwidths in both directions. The expected offset combination for this hypothetical scenario is 0, 5, 10, 10, 5, 0. The resulting bandwidth is 35 seconds per cycle in both directions. The optimal solution of this model displayed in the time-space diagram in Figure 3-3 below.

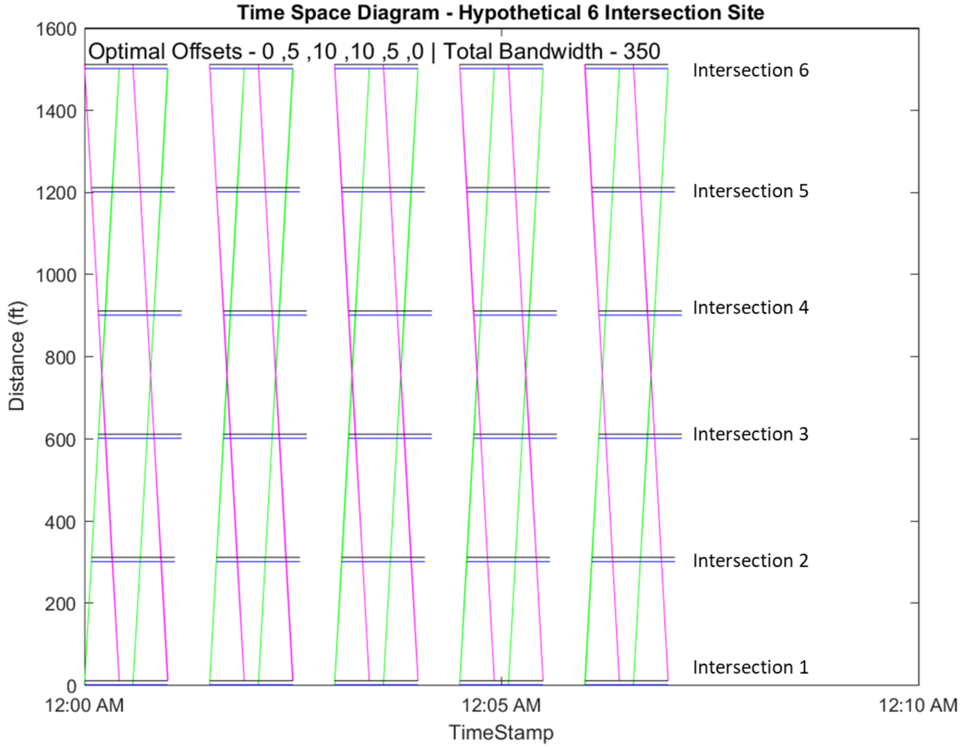


Figure 3-3 - Time Space diagram showing three bi-directional bandwidths for 6-intersection hypothetical site (f=1)

3.4.3. Three-Intersection Real-world Site – Dynamic Bandwidth

Finally, the new MILP model was tested using real-world split monitor data for a system of three coordinated intersections in Clayton, NC. A progression speed of 45 mph or 66 ft/sec was considered. The input data consisted of 20 cycles of signal logs from a weekday AM peak plan, between 6:30 am and 7:30 am, with a common cycle length of 170 seconds. The location and geometry of this site is shown in Figure 3-4.

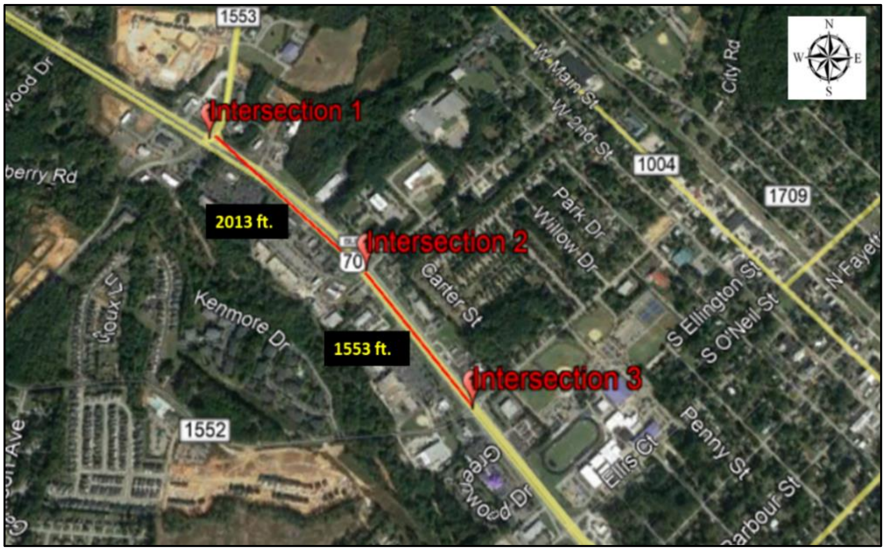


Figure 3-4 - Location of Study Site in Clayton, NC

Figure 3-5 shows the time-space diagram for the optimal offset combination as provided by the MILP program. The optimal offset provided by the MILP is 0, 5, 86 and the total bandwidth in both directions is 2883.01 seconds.

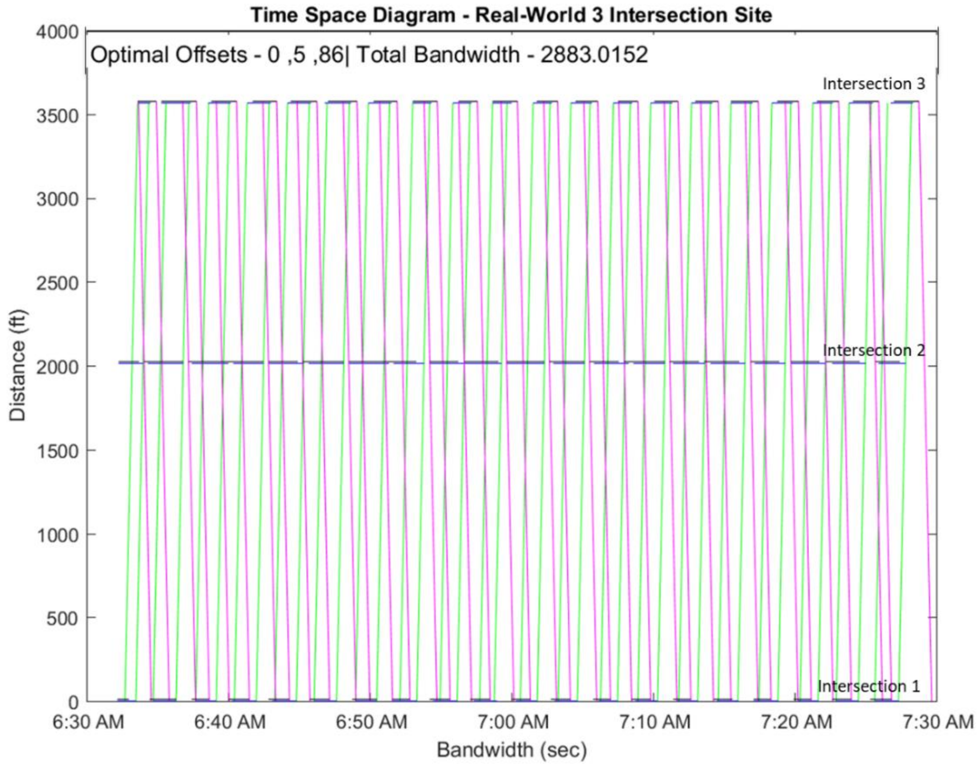


Figure 3-5 - Time Space diagram showing the bi-directional bandwidths for 20 input cycle for 3-intersection real-world site (f=1)

From Table 3-3, we can see that the average bandwidth for the solution in the outbound direction was 81.4 seconds per band and in the inbound direction was 66 seconds per band. Also, the number of bands in the inbound direction is 19. On examining the time-space diagram, it shows that the 1 out of 3 constraints helped the MILP select the second cycle at intersection 2 and 3 to build the first inbound green band.

Table 3-3 - MILP optimal solution quality

Direction of Travel	**Number of green bands**	**Average Bandwidth (sec)**	**Minimum Bandwidth (sec)**	**Maximum Bandwidth (sec)**
Outbound	20	81.39773	46.5303	101
Inbound	19	66.05582	38.5	86

3.4.4. Comparison of MILP Results to Exhaustive Search Solutions for Real-World Site

The solution of the MILP for the real-world site was compared to the results of the exhaustive search [3]. Figure 3-6 shows the solutions for all offset combinations in blue and the optimal solution in yellow. The optimal offset combination calculated by the MILP is shown in read while the observed offset in field is shown in green. The solution of the MILP lies close to the exhaustive search optimal solution.

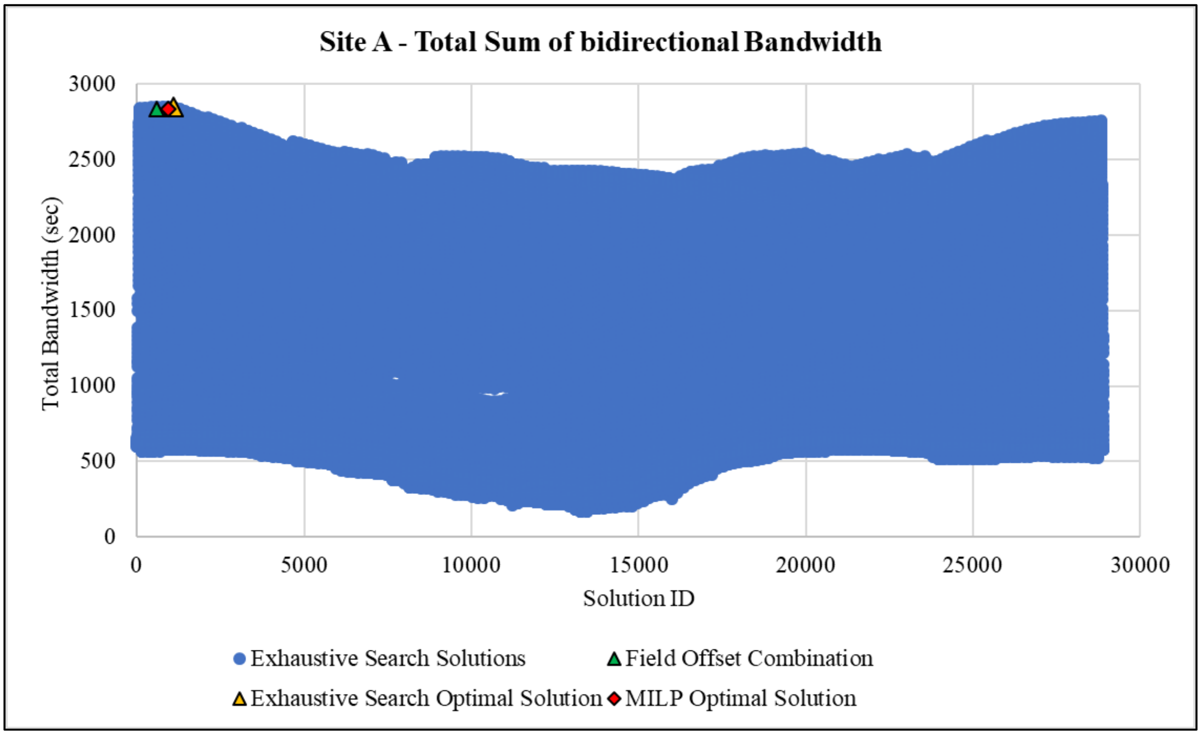


Figure 3-6 - Exhaustive search solutions for all offset combination for real-world study site

In Table 3-4 below, we compare the bandwidths, as calculated by the exhaustive search and MILP for the two optimal solutions. It shows that the optimal offset combination resulting from the exhaustive search method is 0, 6, 85, while that calculated by the MILP is 0, 5, 86. The MILP objective function value is 31 seconds higher than the objective function value for the optimal offset combination calculated by the exhaustive search algorithm. This is because of the discretization for the bandwidth in each cycle as calculated by the exhaustive search algorithm was designed [3]. It draws a hypothetical trajectory at 1 second intervals and then sums up the number of discrete seconds that a vehicle can traverse the system of signals without encountering a red light. However, the MILP can account for those fractional seconds because the start and end of bandwidths are configured as continuous variables.

Table 3-4 - Comparing MILP optimal solution to Exhaustive Search solution

Method	Offset 1	Offset 2	Offset 3	Total Bandwidth (sec)
Exhaustive Search	0	6	85	2852
MILP	0	5	86	2883.01

The bandwidth calculated by the exhaustive search algorithm for the MILP optimal offset combination (0, 5, 86), as shown in Table 3-5 was about 55 seconds lower than the bandwidth calculated by the MILP. That is why this offset combination does not show up as the optimal solution from the exhaustive search and is due to the fractional seconds calculated by the MILP and not by the exhaustive search.

Table 3-5 - Comparing bandwidths calculated by MILP and Exhaustive Search for MILP optimal offset combinations

Method	Offset 1	Offset 2	Offset 3	Total Bandwidth (sec)
Exhaustive Search	0	5	86	2838
MILP	0	5	86	2883.01

3.5. Conclusions and Future Work

This paper presented a new Mixed Integer Linear Programming formulation to solve the problem of maximizing dynamic bandwidth on major arterials, by using cycle-by-cycle actual used green times for each of the major approaches from the signal controller split monitor data as input. The mathematical model was tested on two hypothetical programmed bandwidth scenarios and one real world case study. The results for the hypothetical three-intersection site were compared against the results for MAXBAND model, under different values of demand ratios. The results of the new model for a six-intersection hypothetical site showed that it can be extended to systems with larger number of intersections as well.

For the real-world case study, a three-intersection site in Clayton, NC was selected, and the input data used was for 20 cycles for a weekday AM peak plan. The result from the MILP was compared to the results of the exhaustive search algorithm that it is based on. It was found that the total 2-way bandwidth calculated by the MILP optimal solution was about 31 seconds higher than the optimal solution of the exhaustive search. The reason behind this is that the exhaustive search calculates the bandwidth in every cycle by rounding down to the nearest second.

The model shows promise for use in real-world systems with larger number of intersections. However, due to lack of data, testing could not be conducted for larger systems of signals. However, in practice, the size of a typical coordinated signalized arterial managed by NCDOT is usually 3-5 intersections, and very rarely the number reaches 7 or 8. That is why the model presented in this chapter can be considered to be valid for use in most real-world sites in the near future.

Extension of the new model to optimization of dynamic green bands for a network of intersections and multiple movements (i.e. relaxing the sole focus on through trip travel along the entire arterial) holds promise for further research. Other factors, such as queue clearance and platoon dispersion, etc. can be considered and included into the model.

The model presented in this paper does not take into consideration the effects of platoon dispersion, as discussed in TRANSYT 7F. Incorporation of platoon dispersion in this model may make it non-linear and move it away from the realms of linear optimization techniques. However, the models discussed in TRANSYT 7F could be modified to include dynamic bandwidth optimization as well.

The model also does not account for secondary bandwidths, which may occur within a cycle due to the extended green times on the major phases. However, future work is required to correctly assess the impact of these secondary bands. It can be argued to have a negative impact on the overall objective function by causing drivers to drive rashly and utilize these small bands to avoid stopping, thereby increasing the risk of crashes. In that case, secondary bands can be considered as a conflicting objective in future extensions of this model and be solved using bi-criteria optimization techniques.

3.6. References

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

[2] Yang, X. K. (2001). Comparison among computer packages in providing timing plans for Iowa Arterial in Lawrence, Kansas. Journal of Transportation Engineering, 127(4), 311-318.

[3] Kim, S., Hajbabaie, A., Williams, B. M., & Rouphail, N. M. (2016). Dynamic Bandwidth Analysis for Coordinated Arterial Streets. Journal of Intelligent Transportation Systems, 20(3), 294-310.

[4] Kim, S. (2014). Dynamic Bandwidth Optimization for Coordinated Arterial (Unpublished doctoral dissertation).

[5] Wu, X., Tian, Z., Hu, P., & Yuan, Z. (2012). Bandwidth optimization of coordinated arterials based on group partition method. Procedia-Social and Behavioral Sciences, 43, 232-244.

[6] Little, J. D., Kelson, M. D., & Gartner, N. H. (1981). MAXBAND: A versatile program for setting signals on arteries and triangular networks.

[7] Messer, C. J., Whitson, R. H., Dudek, C. L., & Romano, E. J. (1973). A variable-sequence multiphase progression optimization program (No. 445).

[8] Robertson, D. I. (1969). TRANSYT: a traffic network study tool.

[9] Lieberman, E. B., & Woo, J. L. (1976). SIGOP II: A new computer program for calculating optimal signal timing patterns. Transportation research record, 596, 16-21.

[10] Brooks, W. D. (1965). Vehicular traffic control: Designing traffic progression using a digital computer. IBM-Data Processing Division, Kingston, NY.

[11] Morgan, J. T., & Little, J. D. (1964). Synchronizing traffic signals for maximal

bandwidth. Operations Research, 12(6), 896-912.

[12] Little, J. D. (1966). The synchronization of traffic signals by mixed-integer linear programming. Operations Research, 14(4), 568-594.

[13] Gartner, N. H., Assman, S. F., Lasaga, F., & Hou, D. L. (1991). A multi-band approach to arterial traffic signal optimization. Transportation Research Part B: Methodological, 25(1), 55-74.

[14] Serafini, P., & Ukovich, W. (1989). A mathematical model for the fixed-time traffic control problem. European Journal of Operational Research, 42(2), 152-165. doi:10.1016/0377-2217(89)90318-4

[15] Girianna, M., & Benekohal, R. F. (2004). Using genetic algorithms to design signal coordination for oversaturated networks. Journal of Intelligent Transportation Systems, 8(2), 117-129.

[16] Tian, Z., & Urbanik, T. (2007). System partition technique to improve signal coordination and traffic progression. Journal of Transportation Engineering, 133(2), 119-128.

[17] Buckholz, J. W. (1993). The 10 major pitfalls of coordinated signal timing. ITE Journal, Vol. 63, No. 8, pp. 27-29.

[18] Skabardonis, A. (1996). Determination of timings in signal systems with traffic-actuated controllers. Transportation Research Record: Journal of the Transportation Research Board, (1554), 18-26.

[19] Chang, E. (1996). Guidelines for actuated controllers in coordinated systems. Transportation Research Record: Journal of the Transportation Research Board, (1554), 61-73.

[20] Beard, C., & Ziliaskopoulos, A. (2006). System optimal signal optimization formulation. Transportation Research Record: Journal of the Transportation Research Board, (1978), 102-112.

[21] Peifeng, H. U., Zongzhong, T. I. A. N., Zhenzhou, Y. U. A. N., & Shunping, J. I. A. (2011). Variable-bandwidth progression optimization in traffic operation. Journal of Transportation Systems Engineering and Information Technology, 11(1), 61-72.

[22] Liu, Q., & Xu, J. (2012). Traffic signal timing optimization for isolated intersections based on differential evolution bacteria foraging algorithm. Procedia-Social and Behavioral Sciences, 43, 210-215.

[23] Shirvani, M. J., & Maleki, H. R. (2016). Enhanced variable bandwidth progression optimisation model in arterial traffic signal control. IET Intelligent Transport Systems, 10(6), 396-405. doi:10.1049/iet-its.2015.0061

[24] 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.

[25] 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.

4. OPTIMIZING TRAFFIC SIGNAL RE-TIMING DECISIONS USING ARCHIVED DATA FOR PERFORMANCE METRICS

4.1. Introduction

Increases in traffic volumes can place heavier demands on urban streets in overcrowded areas, necessitating the need for more effective traffic control and evaluation of signal timing plans. A well-timed signal system can reduce fuel consumption, eliminate unnecessary stops and delays, and encourage driver safety. To realize such benefits, a system of signals in close proximity are coordinated so that vehicles arriving at the first intersection can ideally traverse the entire system of signals with minimal stops, thereby enabling better progression through a major corridor. However, optimization of vehicular progression through coordinated signal timing plans alone is not enough to ensure reliable system performance on urban streets. After initial optimization, maintenance needs to be conducted and signals need to be re-timed to ensure that a system continues to operate optimally despite population growth and other factors that impact a transportation network.

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, improved control strategies and, occasionally, minor roadway improvements *(1)*. Re-timing can greatly increase the flow of traffic through signalized corridors and requires constant monitoring of various sources of field data.

Signal systems re-timing involves designing effective signal timing plans to meet changing travel patterns and characteristics. Yet, although current FHWA guidance states that traffic signals should be reviewed and retimed at intervals of 30 months to 3 years, most agencies generally retime their signals within 5 years *(2)*. Sufficient data is key to a successful re-timing plan. However, nearly half of the transportation agencies surveyed (43%) as part of the *Urban Mobility Report* and *National Traffic Signal Report Card*, indicated that they do not regularly collect and analyze traffic data for signal timing, and many existing traffic data collection programs do not assess the quality of data collected *(3)*. Resource limitations are the most cited reason for suboptimal signal timing plans. However, many studies show that re-timing can be a cost-effective expenditure, with an average cost of $3,700 per signal and 26 person-hours of work for most agencies *(2)*. Consequently, state DOTs and traffic agencies not only lack resources, but often lack a means of prioritizing re-timing projects to optimize the resources they have. Hence, signal re-timing project planning needs to be more data-driven, defensible and efficient than is currently the state of practice in the United States.

One potential solution, Advanced Traffic Management Systems (ATMS), enables the logging of various data points that can provide insight into the performance of arterial streets. With the advent of Automated Traffic Signal Performance Measures (ATSPMs), state DOTs have a new way to continuously monitor the performance of traffic signals using high-resolution data. Like Utah and Indiana, more state DOTs are becoming interested in the ATSPM program, with many investigating the benefits of constant monitoring of traffic signal performance. However, this approach requires the use of high-resolution data from vehicle detectors, which are expensive to install and maintain, in addition to certain data acquisition equipment and communication infrastructure. ATSPMs are still a new initiative used only by certain state DOTs and most of the states in the country are still in need of funds and resources to implement ATSPMs. Thus, in many parts of the U.S., decisions regarding selection of signals for re-timing are still based on ad-hoc simulations or the locations that have garnered the most consistent citizen complaints.

To make the decision-making process more systematic and informed, sound data analysis is required. Archived signal controller logs and other data sources, such vehicle probe-based travel time data, are more readily available than the detector data needed for ATSPMs. While reported travel times or speeds on an arterial street can provide insights about the corridor level performance of the road, the cycle-by-cycle used green times logged by the signal controllers, provide the potential for calculating and evaluating intersection-level performance measures. Other information such as geometry of the road and Average Annual Daily Travel (AADT) can also be used to eliminate the need for costly roadway improvements designed to increase capacity and to identify whether signal timing plans need to be revised for a site. This paper provides a methodological approach for mining through large amounts of signal controller log data to calculate intersection-level arterial performance measures to optimize state DOT and agency resources. This approach incorporates existing and widely available data sources to build a decision-making tool to aid with selecting and prioritizing corridors in need for signal re-timing. The methodology is also demonstrated using a case study, which details the steps for analysis of five different intersections, ultimately providing exact recommendation for mobility improvement at these intersections by improving time of day signal timing plans.

4.2. Literature Review

The development of signal re-timing prioritization tools has not been studied extensively, although the benefits of re-timing traffic signals have been widely established. There are several studies that assess the benefits of signal re-timing and discuss the state-of-practice in the field of signal performance monitoring. The Virginia Department of Transportation recently developed a traffic signal needs assessment framework to support signal maintenance and operations funding allocation, which incorporated several performance measures including the mean time between failures, mean time to repair, average signal component service life, average remaining service life, and share of preventive maintenance costs with respect to overall maintenance costs *(4).* NCHRP synthesis 409 provides a detailed review of the current traffic signal retiming practices in the context of traffic signal design, signal timing objectives, requirements, and practices. The synthesis also discusses strengths and weaknesses in the signal retiming process and identifies the need for further research to develop detailed design requirements for archived data user services and management systems, which can help reduce the cost of evaluating existing signal timing plans and would also improve their usefulness in determining the need for retiming *(3).*

Li et al. (2015) used crowd-sourced vehicle probe data to evaluate travel time improvements associated with signal re-timing *(5)*. Their study compared pre-maintenance, post-time-of-day and clock maintenance, and post-progression optimization travel time data. They found that while signal time-of-day plan maintenance accounted for some of the benefits, the savings from that approach were less reliable than progression optimization, which improved both travel times and reliability. In terms of measuring signal performance, Sharma and Bullock used vehicle delay and queue length as quantitative measures in for real-time measurement of vehicle delay and queue length at a signalized intersection and compared these automated delay and queue estimates with field measurement *(6).*

Intersection delay is also applied in a study by Liu et al., and is measured for each phase measured in real-time via an advanced surveillance system *(7)*. They evaluated the performance of the proposed system with a high-performance microscopic traffic simulation program, Paramics, and demonstrated the effectiveness of the proposed system. Another study by Liu et al. introduced a method for time-dependent queue length estimation that uses high-resolution “event-based” traffic signal data to identify traffic state changes that distinguish queue discharge flow from upstream arrival traffic *(8)*.

Another congested corridor study was conducted by Krohn et al., where they performed travel time assessment along five corridors in the greater Philadelphia, Pennsylvania area with AADTs greater than 30,000 vehicles using connected vehicle data *(9)*. Following six weeks of monitoring, both before and after the installation of adaptive signal control, four out of the five corridors showed substantial reductions in travel times that amounted to approximately $32 million in user benefits. They recommended that these techniques be integrated into modern traffic signal system central systems to prioritize timing initiatives and noted that they can be used to complement high resolution traffic signal data performance measures.

In recent years, the Federal Highways Association (FHWA) has focused significant efforts on developing performance measures for transportation systems, including signalized corridors. Day et al. (2014) developed various traffic signal performance measures based on high-resolution controller event data, which consist of a log of discrete events such as changes in detector and signal phase states *(10)*. They also discussed signal operations, from the perspectives of vehicle capacity allocation and vehicle progression and showed how the travel time data can be used for evaluating system operations and assessing the impact of signal retiming activities.

AASHTO Innovation Initiative 2013 focused on disseminating the benefits of Automated Traffic Signal Performance Measures (ATSPMs) and emphasized that signals should be retimed more frequently *(11)*. After learning about automated traffic signal performance (ATSPM) measures at the 2012 TRB Annual Meeting, the Utah DOT partnered with Purdue University and the Indiana DOT to implement ATSPMs. Consequently, a cost–benefit analysis of ATSPM implementation in Utah showed a user-cost savings of more than $3 million due to reduced delays at traffic signals *(12)*. The Utah DOT ATSPM website includes the Purdue Link Pivot Tool *(13)*, which uses arrivals on green as the measure for offset optimization. The measure is analyzed using vehicle counts from detector data.

Similarly, the 2016–2025 State Transportation Improvement Program, which identifies projects that will receive funding during that period, is the first 10-year plan developed under the 2013 Strategic Transportation Investments law. It is a data-driven approach that uses different criteria and measures such as existing congestion, safety score, freight volume, percentage of truck volume, etc. to help prioritize capacity improvement projects all over the state *(14)*. While traffic signal retiming projects are not funded under this program, this data-driven approach may be extended to retiming if the data is available and collected.

Overall, existing literature shows that most previous studies related to traffic signal performance focus on either complex algorithms that can be difficult to execute or require significant amount of high-resolution traffic data that requires purchasing additional equipment. Even traditional signal timing tools like Synchro and Vistro use volume counts as an input into their simulation models, which can be difficult to obtain than controller data. Also, their solutions are based more on average travel conditions rather than at a more granular level, like cycle to cycle. Cycle-by-cycle signal controller logs, however, have not been utilized as the primary data source to assess the effectiveness of signal timing plans. Thus far, little research has focused on developing a simpler technique to utilize these readily available resources and providing exact timing recommendations for each movement at an intersection.

4.3. Data Sources

The signal re-timing prioritization method described in this paper can incorporate data from different sources that can be easily accessed by all state government agencies. These data sources and the information they provide are summarized below:

4.3.1. Geometry

Information related to geometric features of a road segment can be gathered from most state DOT websites, in the form of traffic data event shape files. They provide GIS shape files that provide data such as number of lanes and lane widths. Traffic data event shape files also contain information related to the default capacities on signalized arterials, which is used in the methodology for re-timing prioritization as described in the subsequent sections.

4.3.2. Average annual daily traffic (AADT)

Average annual daily traffic (AADT) is the estimate of typical daily traffic on a road segment for all days of the week and is developed using a full year of traffic count data, usually recorded by a permanent traffic counter. NCHRP Report 765 provides guidelines on AADT adjustment factors, which include daily factors, weekly factors, seasonal adjustment factors, and axle correction factors developed from Automatic Traffic Recorders (ATRs) that collect data continuously throughout the year *(15)*. Analyzing trends in AADT from year-to-year can provide estimates of growth in daily travel and can help detect a corridor in need of capacity improvements such as additional lanes or road widening, as opposed to one that needs re-timing of its traffic signals. In the state of North Carolina, there are a limited number of ATRs, and most AADT values are estimated based on short term counts every one to two years.

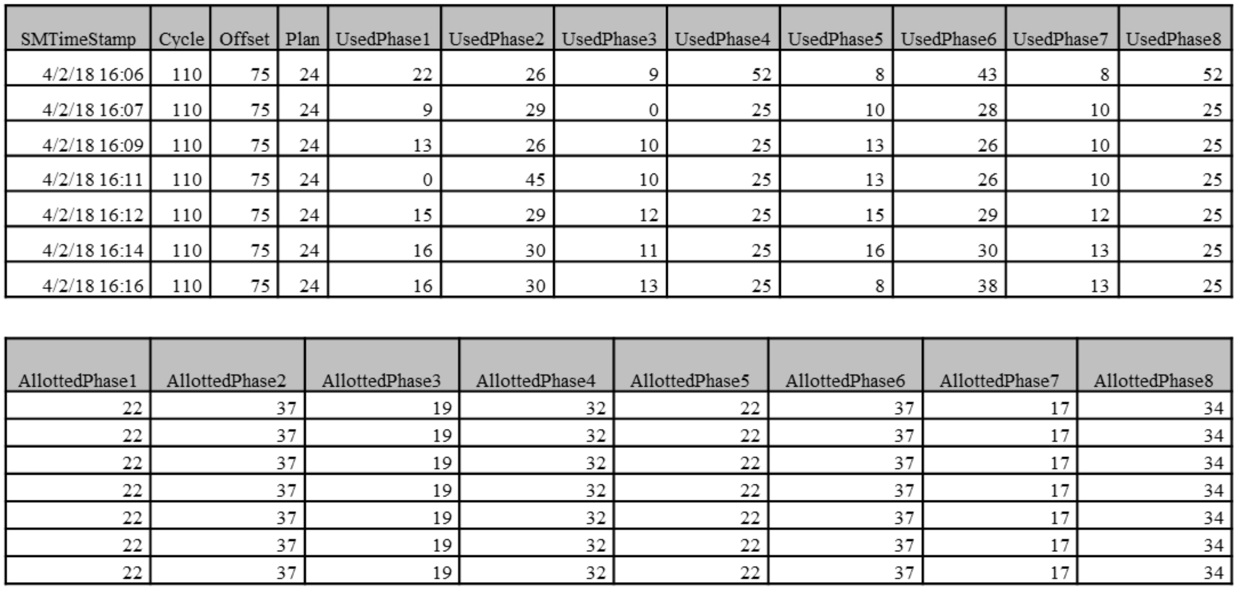
4.3.3. Probe Vehicle based Travel Time

Emerging ITS technologies like probe vehicles make use of cell phones, GPS, and Automated Vehicle Location (AVL) data from a portion of a traffic stream and help in evaluating the traffic conditions of road networks. Probe vehicle data or floating car data serve as a valuable source of data that can be used to determine operating conditions of a signalized corridor and identify congestion patterns. Commercial probe data aggregators like HERE, INRIX and TomTom report this data for Traffic Message Channels (TMC) representing directional roadway links. The data consists of speed and travel time aggregated for a stream of vehicles for each TMC at every 1-minute, 5-minute or 15-minute interval in a day. New reliability performance measures proposed by the Federal Highways Administration (FHWA) as part of the Moving Ahead for Progress in the 21st Century Act (MAP-21) can be calculated to assess reliability-based performance on the corridor *(16)*.

4.3.4. Signal Controller Log Data

The actual phase length and reason for phase termination for a specific phase at an intersection can be gleaned from split monitor data provided by traffic signal controllers. High-resolution phase data is recorded as raw status entries for every cycle. This data logs information regarding the number of seconds of green time allocated to each phase or movement at each intersection for each cycle. This can help evaluate if vehicle detection for the minor movements causes the phases to be called and extended as expected for the prevailing traffic conditions. The performance of a timing plan can be gauged by calculating metrics like the percentage of max-outs, gap-outs, force-offs, etc., which are explained in the next section. Most Advanced Transportation Controllers (ATC) provide system log data for monitoring the system’s operational status, through information like the offset, active vehicle and pedestrian phases, coordination plan, offsets, preemptions, etc. Table 4-1 shows an example of the data collected from the archived signal controller logs, which include timestamps, cycle length, offset, plan number, used green time and the allotted green times for each movement. Each observation corresponds to each cycle within the data collection period.

Table 4-1 - Sample signal controller split monitor log data



SMTimeStamp	Cycle	Offset	Plan	UsedPhase1	UsedPhase2	UsedPhase3	UsedPhase4	UsedPhase5	UsedPhase6	UsedPhase7	UsedPhase8
4/2/18 16:06	110	75	24	22	26	9	52	8	43	8	52
4/2/18 16:07	110	75	24	9	29	0	25	10	28	10	25
4/2/18 16:09	110	75	24	13	26	10	25	13	26	10	25
4/2/18 16:11	110	75	24	0	45	10	25	13	26	10	25
4/2/18 16:12	110	75	24	15	29	12	25	15	29	12	25
4/2/18 16:14	110	75	24	16	30	11	25	16	30	13	25
4/2/18 16:16	110	75	24	16	30	13	25	8	38	13	25

AllottedPhase1	AllottedPhase2	AllottedPhase3	AllottedPhase4	AllottedPhase5	AllottedPhase6	AllottedPhase7	AllottedPhase8
22	37	19	32	22	37	17	34
22	37	19	32	22	37	17	34
22	37	19	32	22	37	17	34
22	37	19	32	22	37	17	34
22	37	19	32	22	37	17	34
22	37	19	32	22	37	17	34
22	37	19	32	22	37	17	34

4.4. Methodology

This section describes the methodology developed for the different steps in data analysis, using the previously described data sources. The overall methodology is showed schematically in Figure 4-1 below. The flowchart describes how the different types of recommendation regarding mobility improvement on a signalized corridor can be concluded, using the readily available data sources described above. After selecting a signalized corridor to be examined, the volume to capacity ratio is calculated using the site geometry and AADT data. If the volume to capacity ratio is greater than 1, the analysis can conclude, and the immediate recommendation is that the site needs capacity improvement, such as road widening or additional lanes. If the volume to capacity ratio is less than 1, the methodology calls for analysis of probe-based travel times on the corridor. The trends in travel time can then be developed, along with computing the various MAP-21 travel time reliability metrics. If the travel time trend shows an increasing pattern and the travel time reliability metrics are higher than the acceptable limits, recommendations in the methodology suggest that the corridor needs re-timing, i.e., offset optimization will need to be performed for the entire corridor.

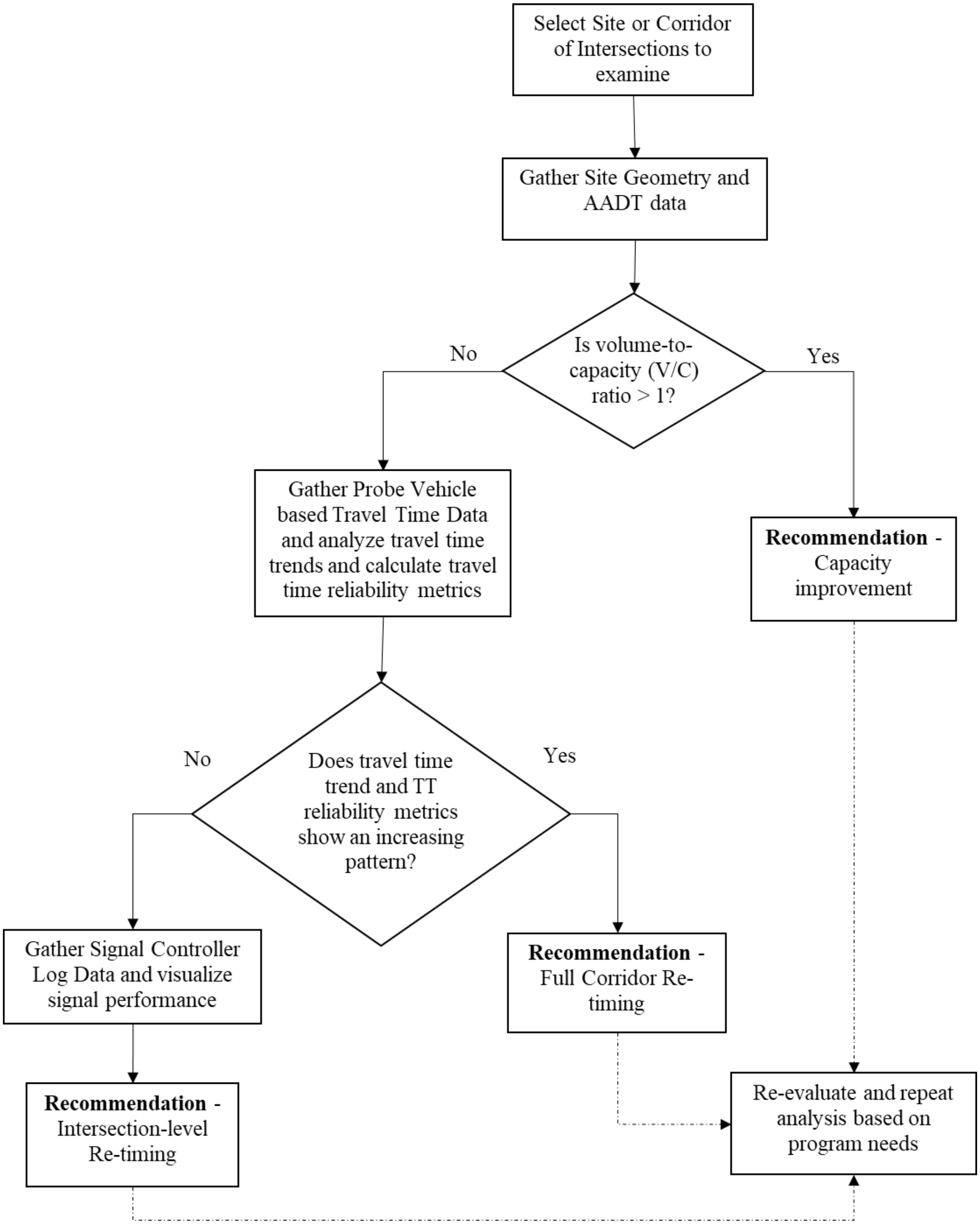

Select Site or Corridor of Intersections to examine
Gather Site Geometry and AADT data
Is volume-to-capacity (V/C) ratio > 1?
No
Yes
Gather Probe Vehicle based Travel Time Data and analyze travel time trends and calculate travel time reliability metrics
Recommendation - Capacity improvement
Does travel time trend and TT reliability metrics show an increasing pattern?
No
Yes
Gather Signal Controller Log Data and visualize signal performance
Recommendation - Full Corridor Re-timing
Recommendation - Intersection-level Re-timing
Re-evaluate and repeat analysis based on program needs

Figure 4-1 - Signal Re-Timing Prioritization Methodology

However, if no increasing travel time trend is detected, an analysis of signal controller log data is conducted, and various signal performance measures are calculated from the green times provided for every cycle of the timing plan for each uncoordinated phase at every intersection. This intersection-level analysis can help detect four main types of failures. The first is a *Power Failure*, in which case the system will not run, and the signal lights may be flashing or completely blank. The second type is a *Communication Failure*, in which case the system will fail to transfer data. In both these cases, no signal controller log data will be available for analysis and installation of new equipment or reconfiguration of existing equipment will be necessary. The third type of failure is a *Time of Day Plan Selection Failure*, in which the major phases for an intersection may fall out of coordination and continue running on free plan. The last type, *Phase Timing Failure*, is a result of inefficient change intervals, splits, allotted time, etc. in timing plans for minor phases at the intersection. This type of failure can cause driver frustration and may be a primary reason for citizen complaints. Inadequate time allotted to each phase or turning movement can increase intersection delay, while inadequate change interval may have safety impacts.

In the next section, this methodology and the various signal performance metrics are described in detail through a case study for an arterial street in Garner, North Carolina.

4.5. Case Study – Timber Drive, Garner, NC

4.5.1. Site Description and Temporal Scope of Study

The site selected for demonstrating the signal re-timing prioritization methodology is a 4-mile road segment on Timber Drive in Garner, NC, shown in Figure 4-2. This is a two-lane state route, with a posted speed limit of 45 mph that includes six semi-actuated and coordinated signalized intersections.

Travel time data for this site was collected for a full year, between May 2017 and April 2018. Only weekdays were considered, as the weekday evening signal timing plan was evaluated for this analysis. For the signal log data analysis, two rounds of data collection were carried out to capture the changes in trend of performance at each intersection and to develop more robust recommendations. The first round of data collection was for two weeks, between September 13, 2017 and October 2, 2017, and the second round of data collection was conducted for four weeks, between April 2, 2018 and April 26, 2018.

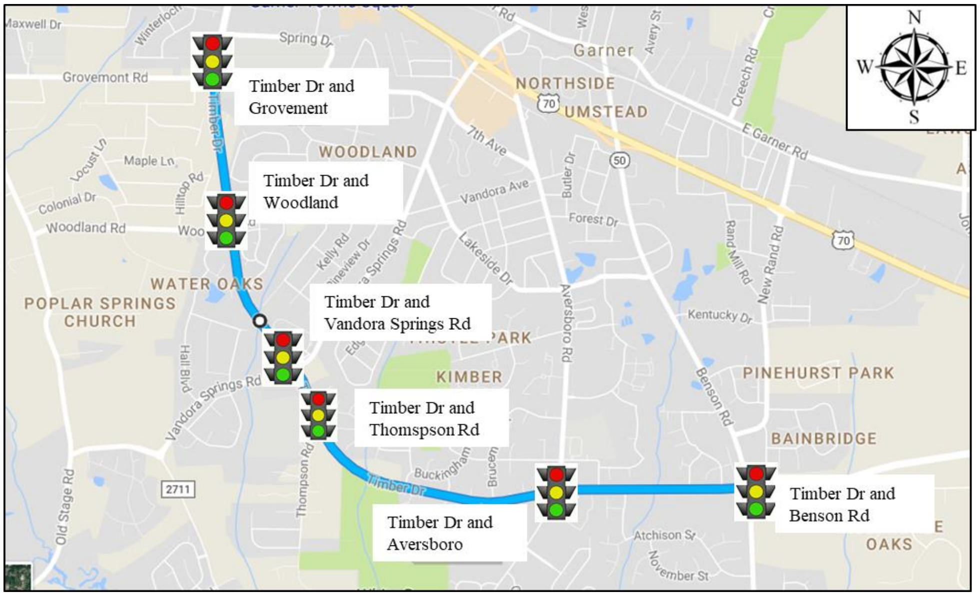


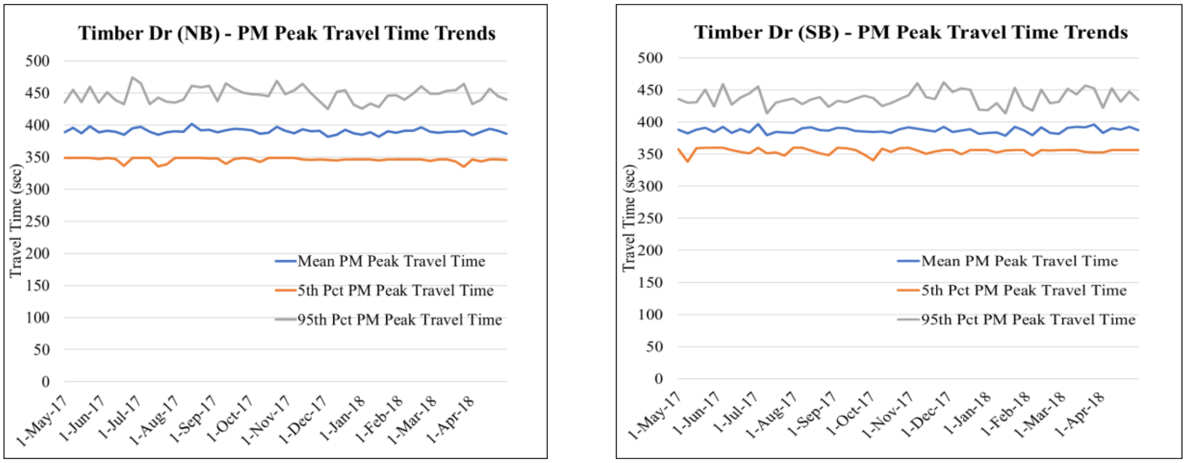
Figure 4-2 - Location of Study Site (Timber Drive, Garner, NC)

4.5.2. Volume-to-capacity Ratio Evaluation

According to the North Carolina Department of Transportation (NCDOT), this site had an Average Daily Traffic of approximately 20,000 vehicles in 2015. The default capacity for this road segment, calculated using the HCM Urban Street worksheet, is 33,100 *(17)*. The volume to capacity ratio of this site is less than 1, which eliminates a need for capacity improvement on this road based on the study method and supports further travel time analysis.

4.5.3. Travel Time Trend Analysis

Figure 4-3 below shows the trend in mean, 5th percentile and 95th percentile travel times over the six months of data collection for the study site in both directions. The source of travel time data was HERE (extracted from RITIS.org), which provides probe-based travel time and speed data on a Traffic Message Channel (TMC) level *(18)*. It does not show any increasing pattern in travel time over the six months.



(a) (b)

Figure 4-3 - Travel Time Index in (a) North Bound (NB) direction and (b) South Bound (SB) on Timber Drive by Hour of Day

The Federal Highways Administration (FHWA) has proposed new travel time reliability performance measures, such as Level of Travel Time Reliability (LOTTR), as part of the MAP-21 Act *(16)*. These performance metrics, described below, were calculated for the study site, using travel time data for a 12-month period. The 5-minute averaged TMC travel times were converted to route travel times by adding those reported for each time interval across TMCs. Level of Travel Time Reliability (LOTTR) is a reliability measure, proposed by the National Performance Reliability Measures (NPRM) defined by Equation 1 and are computed for each period discussed above.

$$Level\ of\ Travel\ Time\ Reliability\ = \frac{80th\ Percentile\ Travel\ Time_i}{50th\ Percentile\ Travel\ Time_i} \tag{4.1}$$

where,

i = one of the four time-periods, which are, weekdays between 6am and 10pm, weekdays between 10am and 4pm weekdays between 4pm and 8pm and weekends between 6am and 8pm.

The threshold value for LOTTR is set to 1.50, which means that segments reporting a value less than this are meeting expectations. The metrics used for this study site are summarized in Table 2. Since the temporal scope of analysis excluded weekends, LOTTR was calculated for the study site for the three weekday time periods. From Table 4-2, the LOTTR for the three weekday time windows for both direction of travel on the study site is close to 1, indicating reliable travel during these periods. Even the 80th percentile and median Travel Time Index on both direction of travel was around 1.1, indicating little congestion.

Table 4-2 - Level of Travel Time Reliability (LOTTR) metrics for Timber Drive North Bound and South Bound

Timber Drive North Bound LOTTR			
	6 am to 10 am	**10 am to 4 pm**	**4 pm to 8 pm**
80th Percentile Travel Time Index	1.12	1.17	1.17
50th Percentile Travel Time Index	1.12	1.14	1.14
LOTTR	1.00	1.01	1.00
Timber Drive South Bound LOTTR			
	6 am to 10 am	**10 am to 4 pm**	**4 pm to 8 pm**
80th Percentile Travel Time Index	1.11	1.12	1.13
50th Percentile Travel Time Index	1.10	1.11	1.12
LOTTR	1.00	1.00	1.01

4.5.4. Signal Controller Log Data Analysis

Signal log data for the study site was collected from NCDOT. Their closed-loop signal systems utilize OASIS, a traffic control firmware developed by Econolite for implementation in an Advanced Transportation Controller (ATC) Type 2070, an open architecture hardware and software standard published by AASHTO, ITE, NEMA, and CALTRANS. NCDOT's effort to transition all state-maintained systems to OASIS is streamlining of access to data for state-maintained signals and systems. The data provided consists of the cycle length, offset values, allocated green time in seconds, and used green time in seconds for each phase or movement at an intersection and for each cycle of a time of day plan, depending on the temporal scope of data extraction. It also provides the amount of extra time provided to the major street through phases at each cycle.

To analyze this data, the following metrics were calculated for each non-coordinated phase at each intersection (phases 1, 3, 4, 5, 7) and their distributions for each phase were analyzed. The phase numbering is in accordance to the National Electrical Manufacturers Association (NEMA) signal phasing, where odd numbers are used to designate the left turn phases while even numbers are used to designate the through phases *(19)*.

1. *Force-off* - when a phase ends regardless of continued demand, i.e., it has reached it maximum green time. This may include extra time from previous phases that terminated early.
2. *Gap Outs* – when the entire allotted green time is not utilized by the phase and the unused green time of a phase is usually transferred to the next phase.
3. *Min-Outs* – when the used green time for the minor phase is equal to the minimum green. The minimum green is the first timed portion of the green interval and is generally based

on the number of vehicles that can be in queue between the upstream phase detector and the stop line.

4. *Skips* – when a minor or non-coordinated phase is not actuated, i.e., the used green time is 0 seconds.

When signal phases are observed as maxing-out or forcing off frequently during time periods when traffic demand is expected to be low, it may indicate a detector malfunction, where the inductive loops trigger a constant on condition. A high proportion of force-offs would typically only occur in a fully functional signal when there is a high number of queued vehicles in each cycle, resulting in consistent phase failures and carryover queues. This means that these phases are using all the programmed green times and could benefit from increased allotted time. If a phase shows a higher percentage of skips, it indicates that the green time allocated to that minor phase is too long or that that phase could operate in permitted phases. Similarly, a higher percentage of min-outs could mean that the green time allotted to the phase might be excessive. The percentage distribution of these metrics, for the weekday PM peak plan between 4pm and 7pm, at each of the intersections on the study corridor for the two rounds of data collection are discussed next, followed by the identification of types of failure and finally, recommendations for timing improvement are provided.

Intersection 1 - Timber Dr and Grovement Rd

This intersection showed no power failure, nor did it experience a time of day plan section failure because the data logs showed that the PM weekday plan had kicked in for each day of both rounds of data collection. However, the signal controller did start having communication issues midway through the second round of data collection, resulting in a communication failure. The signal performance metrics for both the collected data are displayed in Figure 4-4. This figure shows that phase 1, a left turn on the major street, is skipped very frequently and the trend continues in the second round of data collection. This scenario exhibits a *Phase Timing Failure* and the trend indicates that that movement could be well served in a permissive phase, in which no dedicated green time is allotted to the phase.

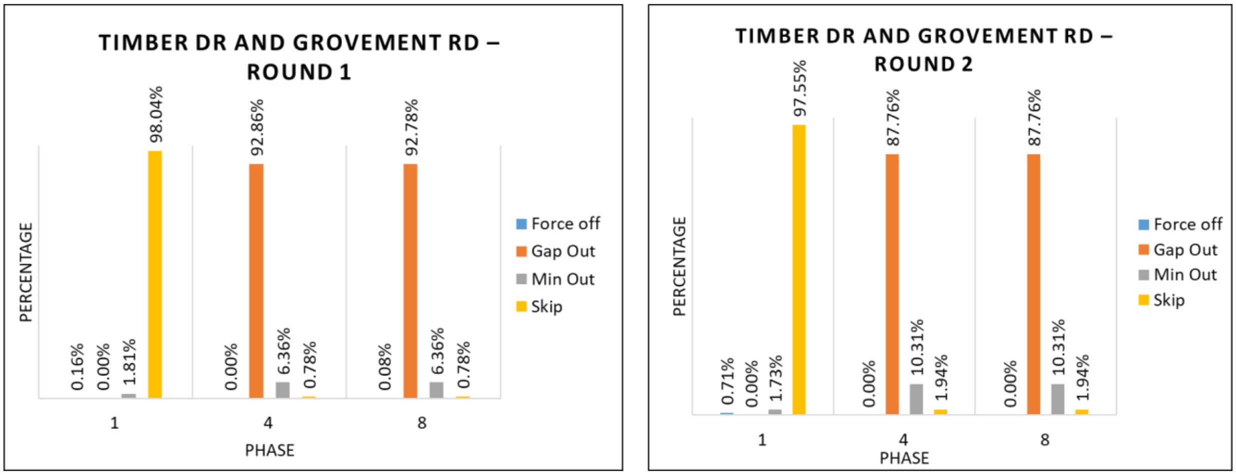


Figure 4-4 - Distribution of Force offs, Gap Outs, Min Outs and Skips for minor phases for both rounds of data collection for Timber Dr and Grovement Rd

Intersection 2 - Timber Dr and Woodland Rd

In the second intersection, no power, communication or time of day plan selection failure was detected. From Figure 4-5, we can see that Phase 8 forces off rather frequently in both rounds of data collection. The two minor street left turns (phases 3 and 7) at this intersection share the lane with the through movements (phases 4 and 8). Phase 7, however, has dedicated green time, when both phase 3 and 8 are not permitted. This results in inadequate service for the demand on the other side (phase 3 and 8). Based on this timing failure, it is apparent that this intersection could benefit from split phasing, in which all movements at one approach are first served, followed by all the movements of the opposing approach *(19)*.

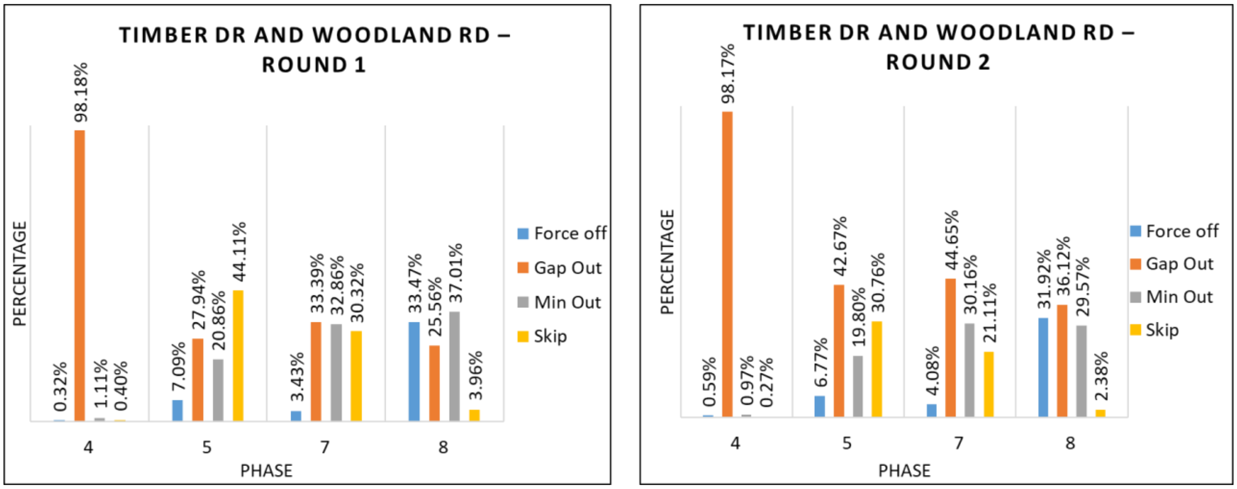


Figure 4-5 - Distribution of Force offs, Gap Outs, Min Outs and Skips for minor phases for both rounds of data collection for Timber Dr and Woodland Rd

Intersection 3 - Timber Dr and Vandora Springs Rd

The data logs for this intersection showed that the weekday PM plan is not called for this intersection but runs on an uncoordinated free plan throughout the two rounds of data collection. This is an example of *Time of Day Plan Selection Failure and* indicates that the hardware or software may need to be reconfigured.

Intersection 4 - Timber Dr and Thompson Dr

The fourth intersection (Thompson Rd) only has dedicated green time for one of the major street left turn movements (phase 5), which is skipped very frequently, as shown in Figure 4-6. For the minor through phases (phases 4 and 8), the percentage of force-offs is found to have increased in the second round of data collection. This shows a substantial increase in demand on these approaches and indicates that increasing the allotted green time for phases 4 and 8 may help resolve the issue.

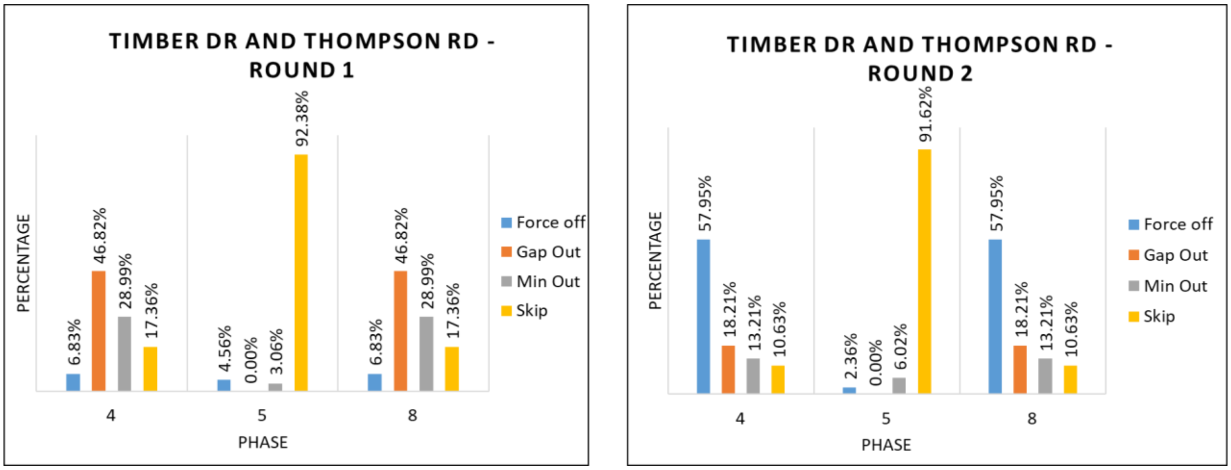


Figure 4-6 - Distribution of Force offs, Gap Outs, Min Outs and Skips for minor phases for both rounds of data collection for Timber Dr and Thompson Rd

Intersection 5 - Timber Dr and Aversboro

At the fifth intersection (Aversboro), a high percentage of force-offs is experienced by phase 8, one of the minor through movements. Although phases 4 and 8 terminate at the same time following the leading left turn movements on the minor street, only phase 8 is frequently using more green time than what is allotted. One possible explanation is that phase 7 skips more frequently and transfers some of the time allotted to it to phase 8, while phase 3 does not transfer as much time to phase 4. This is shown in Figure 4-7.

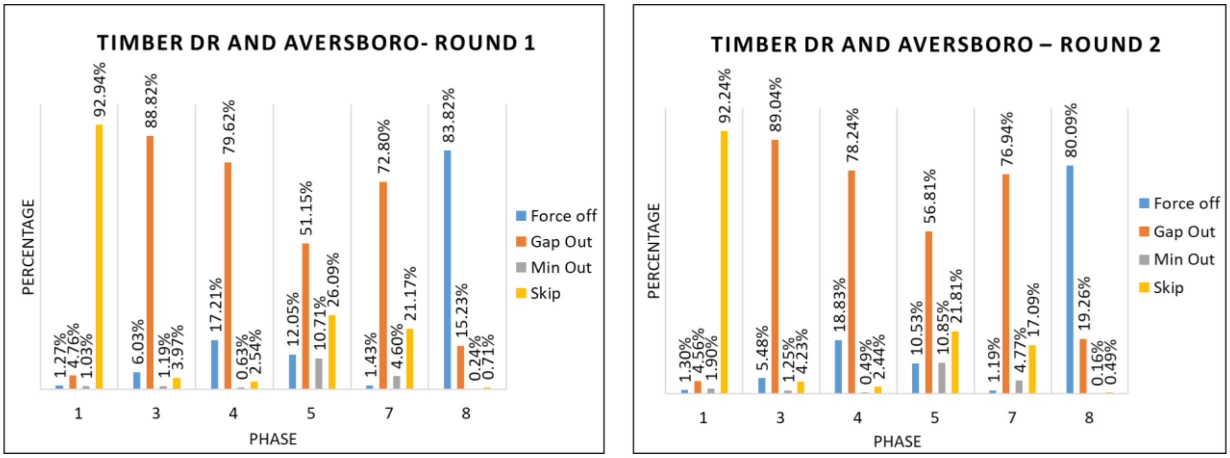


Figure 4-7 - Distribution of Force offs, Gap Outs, Min Outs and Skips for minor phases for both rounds of data collection for Timber Dr and Aversboro

Intersection 6 - Timber Dr and Benson Rd

Finally, Figure 4-8 shows that at the sixth intersection (Benson Rd), the left turn on major street (phase 1) shows a high percentage of force-offs and may benefit from more allotted green time. Similarly, both phase 3 and phase 7, the left turn movements on the minor street, show a higher percentage of force offs, indicating that they may benefit from a higher allotted green time as well.

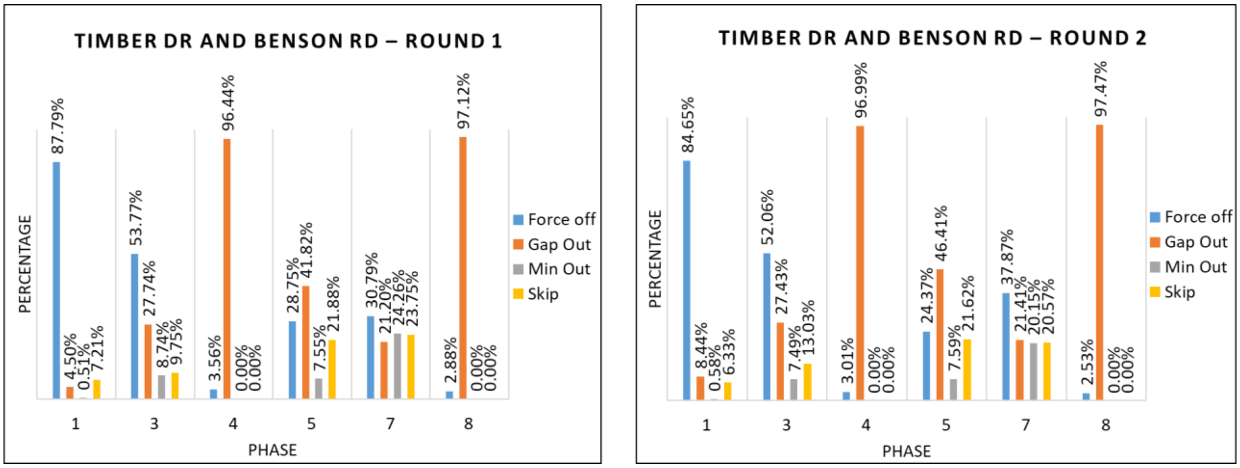


Figure 4-8 - Distribution of Force offs, Gap Outs, Min Outs and Skips for minor phases for both rounds of data collection for Timber Dr and Benson Rd

This case study moves through all the steps of the methodology to identify the problems caused by the timing plan at an intersection level. While this analysis demonstrates how cycle-by-cycle signal controller logs can be used to determine the effectiveness of signal timing plan on the corridor, other case studies may reach a recommendation earlier in the analysis procedure.

4.6. Conclusions and Future Work

This paper presented a novel approach for prioritizing signalized intersections for re-timing. The developed methodology uses easily available data sources and does not require installation of newer and more expensive vehicle detectors. A case study for a six-intersection corridor in Garner, North Carolina was used to explain and demonstrate the procedure, which can detect four different types of traffic signal failures. By calculating and visualizing signal performance metrics using signal controller log data, the case study established how the method can intuitively assist traffic engineers in assessing which movements or phases at an intersection are underserved. The case study presented in this paper demonstrates how cycle-by-cycle signal controller logs can help identify problematic intersections. However, the other data sources described in this paper, like geometry, AADT and travel time, may be more useful in other case studies.

This research is an attempt to investigate the use of archived signal controller log data in assessing the performance of a signalized arterial corridor. It illustrates the benefits agencies can derive from analyzing archived signal timing logs, with one of the primary benefits being that it provides an alternative to installing expensive devices to collect high resolution vehicle detection data. The main benefit of this type of analysis is that it can provide data-supported recommendations for improving intersection timing plans. Finally, it also provides helps explain why repeated citizen complaints may be occurring, even when the travel times on the corridor seem reliable.

However, the main drawback of the methodology developed in this paper is that it relies heavily on the judgement of the analyst to decide if a certain movement at an intersection is being allocated too much or too little time. Future work involving the collection and analysis of data for various sites encompassing different road types and signal timing schemes is needed to establish threshold values or cut off points for the various signal performance metrics. This can help to truly integrate the different data sources presented in this paper and develop a more automated and intelligent decision-making systems.

The methodology developed in this paper also provides a more generalized approach for signal re-timing decision making. For agencies with access to ATSPM on a corridor, this methodology may be extended to provide additional recommendations. This effort can compound the benefits of ATSPM installation, as currently the software provides very limited decision support automatically and relies heavily on user knowledge and effort.

4.7. References

1. Sunkari, S. The benefits of retiming traffic signals. *Institute of Transportation Engineers. ITE Journal*, 2004. 74(4): 26-29.
2. Gordon, RL. *Traffic signal retiming practices in the United States*. Transportation Research Board, Washington D.C., 2010.
3. Schrank, D. L., and Lomax, T. J. *2009 urban mobility report*. Texas Transportation Institute, Texas A & M University,2009.
4. Chen, W., Henley, L., and Price, J. Assessment of traffic signal maintenance and operations needs at Virginia Department of Transportation. *Transportation Research Record: Journal of the Transportation Research Board,* 2009. Vol: 2128.
5. Li, H., Lavrenz, S. M., Day, C. M., Stevens, A., & Bullock, D. M. Quantifying Benefits of Signal Timing Maintenance and Optimization Using both Travel Time and Travel Time Reliability Measures. *Transportation Research Record: Journal of the Transportation Research Board*, 2015.
6. Sharma A., Bullock D.M. and Bonneson, J.A. Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. *Transportation Research Record: Journal of the Transportation Research Board,* 2007. Vol: 2035 pp. 69–80.
7. Liu HX, Oh J-S and Recker W. Adaptive signal control system with online performance measure for a single intersection. *Transportation Research Record: Journal of the Transportation Research Board,* 2002. Vol:1811, pp. 131–138.
8. Liu H, Wu X, Ma W, Hu H. Real-time queue length estimation for congested signalized intersections. *Transportation Research Part C: Emerging Technologies,* 2009. 17(4): 412–

427.

9. Kohn, D., Rymarcsuk, L., Mathew, J., Day, C., Li, H., Patel, A., Farley, D. and Bullock, D.M. Outcome Assessment using Connected Vehicle Data to Justify Signal Investments to Decision Makers, Presented at 96th Annual Meeting of the Transportation Research Board, Washington, D.C., 2017.
10. Day, C.M., R. Haseman, H. Premachandra, T.M. Brennan, J.S. Wasson, J.R. Studevant, and D.M. Bullock. Visualization and Assessment of Arterial Progression Quality Using High Resolution Signal Event Data and Measured Travel Time, Presented at 89th Annual Meeting of the Transportation Research Board, Washington, D.C., 2010.
11. Clayton, R. *Automated Traffic Signal Performance Measures*. AASHTO Innovation Initiative 2013 Focus Technology, 2016.
12. Stevens, D., & Kergaye, C. Pioneering Ideas from the TRB Annual Meeting: Utah DOT Realizes Savings from Innovations. *TR News*, 2017. 310.
13. Day, C. and Bullock, D. M. Link Pivot Algorithm for Offset Optimization. *Purdue University Research Repository*, 2014. doi:10.4231/R7GQ6VPT
14. *State Transportation Improvement Program (STIP).* North Carolina Department of Transportation. https://www.ncdot.gov/initiatives-policies/Transportation/stip/Pages/default.aspx Accessed June 20, 2018.
15. NCHRP Report 765. Analytical Travel Forecasting Approaches for Project-Level Planning and Design. Transportation Research Board (Project 08-83).
16. Federal Register 2016-08014. National Performance Management Measures; Assessing Performance of the National Highway System, Freight Movement on the Interstate System, and Congestion Mitigation and Air Quality Improvement Program. FHWA, U.S.

Department of Transportation.

17. Highway Capacity Manual. Transportation Research Board, National Research Council, Washington, D.C., 2015.
18. RITIS: Regional Integrated Transportation Information System. 2016. www.ritis.org.
19. Koonce, P. *Traffic signal timing manual.* Publication FHWA-HOP-08-024. FHWA, U.S. Department of Transportation, 2008.

5. CONCLUSIONS AND FUTURE RESEARCH

In this research, three different methodologies are presented for developing and maintaining effective signal timing plans on major arterial streets to ensure smooth flow of traffic between intersections and reduce congestion. Arterial bandwidth is considered as the measure of effectiveness, based on which some real-world case studies are also presented.

5.1. Key Findings

The key contributions of this research can be summarized as follows:

1. Chapter 2 showed investigative efforts in finding a method alternative to Exhaustive Search for estimating the entire solution space of dynamic bandwidth optimization problem. Using Locally Weighted Projection Regression (LWPR), a methodology to estimate the entire solution space for the combinatorial problem was developed and evaluated using two study sites. Apart from random sampling technique, another training data selection strategy was also evaluated, which involved sampling every n^{th} offset combination input-output pairs to constitute the training dataset. The code for implementing this can be found in APPENDIX A: MATLAB Code for Locally Weighted Projection Regression. The results from this type of selection technique was found to be comparable to the random sampling technique as it can provide the model with more information about the solution space, enabling it to provide a higher accuracy with lesser percentage of training data.
2. Chapter 3 presented a new Mixed Integer Linear Programming (MILP) formulation to solve the problem of maximizing dynamic bandwidth on major arterials, using cycle-by-cycle actual used green times for each of the major approaches from the signal controller split monitor data as input. This formulation was tested for accuracy using two

hypothetical sites and with real-world split monitor data. The model is a robust technique since it can be augmented to include other factors as well. The MATLAB code implementing this model can be found in APPENDIX B: MATLAB Code for Mixed Integer Linear Program

3. Chapter 4 presented a novel approach for prioritizing signalized intersections for re-timing. The developed methodology uses easily available data sources and does not require installation of newer and more expensive vehicle detectors. By calculating and visualizing signal performance metrics using signal controller log data, a real-world case study was used to demonstrate how the method can intuitively assist traffic engineers in assessing which movements or phases at an intersection are underserved. An example spreadsheet for calculation of the signal performance measures can be found in APPENDIX C: Example EXCEL Spreadsheet Calculations for Signal Performance Metrics from Split Monitor Data
4. Finally, the Mixed Integer Linear Programming (MILP) model is able to find optimal solution for bigger problem quicker than exhaustive search. Since exhaustive search is not fast enough to be used as a validation for the Locally Weighted Projection Regression (LWPR) technique described in Chapter 2, the MILP can be used to validate the results of the LWPR technique in terms of estimating the optimal solution surface.

5.2. Future Research

The research presented in this dissertation leads a foundation for the following future work:

1. The methodology using Locally Weighted Projection Regression (LWPR) suffers from poor performance in terms of computational speed. Use of parallel computing for

implementing the testing phase of the algorithm could speed up this process even further. With better hardware and software configuration, future research is required to determine the accuracy of the nth offset sampling technique, by evaluating sites with larger number of intersections.

2. Due to lack of available data, testing of the MILP model presented in Chapter 3 could not be conducted for larger systems of signals. The new optimization technique should be tested for sites with larger number of intersections.
3. Finding the slack in the offset value at an intersection is important from the perspective of field implementation. This helps the analyst know if the solution is robust in the field and that the system will not fall completely out of sync if the offsets at an intersection change.
4. The MILP model can also be extended by adding other factors as constraints, like queue clearance, or by considering other objective functions, like progression opportunities.
5. Also, the model does not look for secondary bandwidths, which may occur within a cycle as a combination of extra green times on the major phases. However, the impact of these secondary bands needs further research. It could be argued that secondary bands may have a negative impact to the overall objective function from the point of view of road safety. In that case, it can be considered as a conflicting objective in future extensions of this model and be solved using bi-criteria optimization techniques.
6. The new dynamic bandwidth optimization technique should be augmented to include not only major through phases, but also minor phases on side streets, to predict a better offset combination for all movements at intersections. Extension of this to optimization of dynamic green bands for a network of intersections can also be considered.

7. For the re-timing prioritization methodology proposed in Chapter 4, future work involving the collection and analysis of data for various sites encompassing different road types and signal timing schemes is needed to establish threshold values or cut off points for the various signal performance metrics. This can help develop a more automated and intelligent decision-making systems. Since data available to different agencies differ greatly, the tool can be made more adaptable to using any format of input data.

APPENDICES

APPENDIX A: MATLAB Code for Locally Weighted Projection Regression

The code presented in this appendix is for the four-intersection study site, with n^{th} offset sampling technique.

```
clear;
disp('Testing DDI set, Cycle length - 90 sec')

% Number of input variables
d=3;

% Reading All Data
X_all= xlsread('LWPR Site B 09_14_2018.xlsx','Train2');

% Creating filename variable to print and save files
filenameHeader = ['SiteB_nthOffsetSampling'];


% Cycle Length
C = max(X_all(:,4))+1;
offset = X_all(:,2:4);
id = X_all(:,1);
Y_all = X_all(:,5);

% Normalizing offsets
X_all(:,6) = zscore(X_all(:,2));
X_all(:,7) = zscore(X_all(:,3));
X_all(:,8) = zscore(X_all(:,4));
X_all(:,1:5) = [];
n = size(X_all,1);

% Sampling Step Size:
j = 2;

% Sampling every 10 second offset combination
J =0:j*3:87;

%Sampling indices
ind1 = ismember(offset(:,1),J);
ind2 = ismember(offset(:,2),J);
ind3 = ismember(offset(:,3),J);
subset = offset(ind3 == 1 & ind2 == 1 & ind1 == 1, :);
```

```
% Training Set
X = X_all(ind3 == 1 & ind2 == 1 & ind1 == 1, :);
Y = Y_all(ind3 == 1 & ind2 == 1 & ind1 == 1, :);
opt_esa_y = max(Y_all);
indx1 = find(Y_all == opt_esa_y);
indx2 = find(Y == opt_esa_y);
opt_esa_x = X_all(indx1,:);
opt_esa_offset = offset(indx1,:);

% Creating Test Data Set
Xt = X_all;
Yt = Y_all;
offset_test = offset;
offset_test(ind3 == 1 & ind2 == 1 & ind1 == 1, :) = [];
id_test = id;
id_test(ind3 == 1 & ind2 == 1 & ind1 == 1, :) = [];
Yt(ind3 == 1 & ind2 == 1 & ind1 == 1, :)= [];
Xt(ind3 == 1 & ind2 == 1 & ind1 == 1, :)= [];
last = size(Yt,1);

% Log transforming the response variables
Y = log(Y+1);
Yt = log(Yt+1);
n = size(X,1)

% Defining LWPR model parameters
model = lwpr_init(d,1,'name','lwpr_test');
model = lwpr_set(model,'init_D',eye(d)*100);
model = lwpr_set(model,'init_alpha',ones(d)*250);
model = lwpr_set(model,'diag_only',0);
model = lwpr_set(model,'w_gen',0.2);
model = lwpr_set(model,'w_prune',0.7);
model = lwpr_set(model,'meta',1);
model = lwpr_set(model,'meta_rate',250);
model = lwpr_set(model,'update_D',1);
model = lwpr_set(model,'kernel','Gaussian');

tic;
%Train the model
for j=1:10
  inds = randperm(n);
  mse = 0;

  for i=1:n,
     [model,yp,w] = lwpr_update(model,X(inds(i),:)',Y(inds(i),:)');
     mse = mse + (Y(inds(i),:)-yp).^2;
  end

  nMSE = mse/n/var(Y,1);
  fprintf(1,'#Data=%d #rfs=%d
nMSE=%5.3f\n',lwpr_num_data(model),lwpr_num_rfs(model),nMSE);
  if exist('fflush') % for Octave output only
    fflush(1);
  end
end
toc

% Create predictions for the test data
Yp = zeros(size(Yt));
tic;
for i=1:length(Xt),
   yp = lwpr_predict(model,Xt(i,:)',0.001);
   Yp(i,1) = yp;
end
toc

% Calculating error
Yt = exp(Yt)+1;
Yp = exp(Yp)+1;
Y = exp(Y)+1;
ep   = Yt-Yp;
mse  = mean(ep.^2)
nmse = mse/var(Yt,1)
ep_perc = abs(ep./Yt)*100;
disp('Max Percentage error')
max(ep_perc)
disp('Avg Percentage Error')
mean(ep_perc)
disp('Std Dev Percentage Error')
std(ep_perc)
disp('85th percentile Percentage Error')
prctile(ep_perc, 85)
disp('95th Percentile Percentage Error')
prctile(ep_perc, 95)
actual_opt = max(Y_all)
idmax = find(Y_all == actual_opt);
actual_opt_id = id_test(idmax)
actual_opt_offfset = offset_test(idmax,:)
pred_opt = max(Yp)
idmax = find(Yp == pred_opt);
pred_opt_id = id_test(idmax)
pred_opt_offset = offset_test(idmax,:)
```

```
% Saving data as struct
result.actual = [Yt;Y];
result.predicted = [Yp;Y];
result.offset = [offset_test; offset_train];
result.error = [ep_perc; NaN(size(id_train,1),1)];
result.id = [id_test; id_train];
save(filenameHeader,'-struct','result')

% Plotting Actual Solution Space
h(1) = figure(1);
plot(result.id, result.actual, 'b*')
hold on
plot(actual_opt_id(1), actual_opt(1),'--gs',...
    'LineWidth',2,...
    'MarkerSize',8,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor',[0.8,0.8,0.8]);
hold on
title('Actual Solution Space');
xlabel('Solution ID');
ylabel('Bandwidth (sec)');
curtick = get(gca, 'XTick');
set(gca, 'XTickLabel', cellstr(num2str(curtick(:))));
Name=[filenameHeader,'_Actual.png'];
saveas(h(1),Name);


% Plotting Predicted Solution Space
h(2) = figure(2);
plot(result.id, result.predicted, 'b*')
hold on
plot(pred_opt_id(1), pred_opt(1),'--gs',...
    'LineWidth',2,...
    'MarkerSize',8,...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor',[0.8,0.8,0.8]);
hold on
title('Predicted Solution Space');
xlabel('Solution ID');
ylabel('Bandwidth (sec)');
curtick = get(gca, 'XTick');
set(gca, 'XTickLabel', cellstr(num2str(curtick(:))));
Name=[filenameHeader,'_Predicted.png'];
saveas(h(2),Name);
```

APPENDIX B: MATLAB Code for Mixed Integer Linear Program

The MILP formulation uses the MATLAB Toolbox ‘MATLOG’, developed for use at NCSU and is compatible with MATLAB versions 2014b to 2015b. The implementations of the MILP models were done using MATLAB version 2015b.

```
clear
%Enter Excel input filename here
filename = 'Site A’
%Enter proportion of traffic in inbound direction to outbound direction
f = 1

% Reading in data:
S1 = xlsread(filename,'S1');
S2 = xlsread(filename,'S2');
E1 = xlsread(filename,'E1');
E2 = xlsread(filename,'E2');

% Enter common cycle length:
C = 170;

% Input distance(ft) between intersections:
d12 = 2013;
d23 = 1553;

% Input lower and upper limit of offset
lower = -C
upper = C

% Progression speed (ft/sec):
speed = 66

% Finding number of intersections and number of cycles
I = size(S1,1);
J = size(S1,2);

% Creating coefficient matrix for x and y in outbound direction 1
Cx1 =-ones(I,J);
Cy1 = ones(I,J);

% Creating Coefficient matrix for x and y in inbound direction 1
Cx2 =-f*ones(I,J);
Cy2 = f*ones(I,J);
```

```
% Creating coefficient matrix for offset
Coff = zeros(I,J);

% Travel time (between intersections) matrix
TT = [0; d12/speed; d23/speed];

% Creating Coefficient matrices for K's
Ck1 = zeros(I, J);
Ck2 = zeros(I, J);
Ck3 = zeros(I, J);
Ck4 = zeros(I, J);
Ck5 = zeros(I, J);
Ck6 = zeros(I, J);

% Creating LP objective function using Milp function from MATLOG
mp = Milp('SignalTiming');
mp.addobj('max',Cx1,Cy1,Cx2,Cy2,Coff,Ck1,Ck2,Ck3,Ck4,Ck5,Ck6);

%First intersection offset to 0 as reference point
mp.addcstr(0,0,0,0,{[1],{1,1}},0,0,0,0,0,0, '=', 0); % offsets for cycles 2 to J

 % y>x constraints for both directions
for i = 1:I
 for j = 1:J
        mp.addcstr({-1,{i,j}},{1,{i,j}},0,0,0,0,0,0,0,0,0, '>=', 0); %y1-x1>=0
        mp.addcstr(0,0,{-1,{i,j}},{1,{i,j}},0,0,0,0,0,0,0, '>=', 0); %y2-x2>=0
 end
end

% Start and end of bands constraints for first intersection constraint:
for j = 1:J
    mp.addcstr({1, {1,j}},0,0,0,0,0,0,0,0,0,0, '>=', S1(1,j));%x1
    mp.addcstr({1, {1,j}},0,0,0,0,0,0,0,0,0,0, '<=', E1(1,j));%x1
    mp.addcstr(0,{1, {1,j}},0,0,0,0,0,0,0,0,0, '>=', S1(1,j));%y1
    mp.addcstr(0,{1, {1,j}},0,0,0,0,0,0,0,0,0, '<=', E1(1,j));%y1
end

% Finding Big M from data
M = max(E1(I,J), E2(I,J));

% Start and end of bands constraints in outbound direction - cycles 2 to J-1
for i = 2:I
 for j = 2:J-1
        mp.addcstr({1,{i,j}},0,0,0,{-1, {i,1}},{M, {i,j}},0,0,0,0,0, '>=', S1(i,j-1));
        mp.addcstr({1,{i,j}},0,0,0,{-1, {i,1}},0,{M, {i,j}},0,0,0,0, '>=', S1(i,j));
        mp.addcstr({1,{i,j}},0,0,0,{-1, {i,1}},0,0,{M, {i,j}},0,0,0, '>=', S1(i,j+1));
        mp.addcstr({1,{i,j}},0,0,0,{-1, {i,1}},{-M, {i,j}},0,0,0,0,0, '<=', E1(i,j-1));
        mp.addcstr({1,{i,j}},0,0,0,{-1, {i,1}},0,{-M, {i,j}},0,0,0,0, '<=', E1(i,j));
        mp.addcstr({1,{i,j}},0,0,0,{-1, {i,1}},0,0,{-M, {i,j}},0,0,0, '<=', E1(i,j+1));
        mp.addcstr(0,{1,{i,j}},0,0,{-1, {i,1}},{M, {i,j}},0,0,0,0,0, '>=', S1(i,j-1));
        mp.addcstr(0,{1,{i,j}},0,0,{-1, {i,1}},0,{M, {i,j}},0,0,0,0, '>=', S1(i,j));
        mp.addcstr(0,{1,{i,j}},0,0,{-1, {i,1}},0,0,{M, {i,j}},0,0,0, '>=', S1(i,j+1));
        mp.addcstr(0,{1,{i,j}},0,0,{-1, {i,1}},{-M, {i,j}},0,0,0,0,0, '<=', E1(i,j-1));
        mp.addcstr(0,{1,{i,j}},0,0,{-1, {i,1}},0,{-M, {i,j}},0,0,0,0, '<=', E1(i,j));
        mp.addcstr(0,{1,{i,j}},0,0,{-1, {i,1}},0,0,{-M, {i,j}},0,0,0, '<=', E1(i,j+1));
 end
end

% Start and end of green bands constraints in outbound direction for first cycle (J=1)
for i = 2:I
        mp.addcstr({1,{i,1}},0,0,0,{-1, {i,1}},0,{M, {i,1}},0,0,0,0, '>=', S1(i,1)); %xi1-Oi >= Si1
        mp.addcstr({1,{i,1}},0,0,0,{-1, {i,1}},0,0,{M, {i,1}},0,0,0, '>=', S1(i,2)); %xi1-Oi <= Si2
        mp.addcstr({1,{i,1}},0,0,0,{-1, {i,1}},0,{-M, {i,1}},0,0,0,0, '<=', E1(i,1)); %yi1-Oi >= Ei1
        mp.addcstr({1,{i,1}},0,0,0,{-1, {i,1}},0,0,{-M, {i,1}},0,0,0, '<=', E1(i,2)); %yi1-Oi >= Ei2
        mp.addcstr(0,{1,{i,1}},0,0,{-1, {i,1}},0,{M, {i,1}},0,0,0,0, '>=', S1(i,1)); %xi1-Oi >= Si1
        mp.addcstr(0,{1,{i,1}},0,0,{-1, {i,1}},0,0,{M, {i,1}},0,0,0, '>=', S1(i,2)); %xi1-Oi <= Si2
        mp.addcstr(0,{1,{i,1}},0,0,{-1, {i,1}},0,{-M, {i,1}},0,0,0,0, '<=', E1(i,1)); %yi1-Oi >= Ei1
        mp.addcstr(0,{1,{i,1}},0,0,{-1, {i,1}},0,0,{-M, {i,1}},0,0,0, '<=', E1(i,2)); %yi1-Oi >= Ei2
end

% Start and end of green times in outbound direction for last cycle
for i = 2:I
        mp.addcstr({1,{i,J}},0,0,0,{-1, {i,1}},{M, {i,J}},0,0,0,0,0, '>=', S1(i,J-1));
        mp.addcstr({1,{i,J}},0,0,0,{-1, {i,1}},0,{M, {i,J}},0,0,0,0, '>=', S1(i,J));
        mp.addcstr({1,{i,J}},0,0,0,{-1, {i,1}},{-M, {i,J}},0,0,0,0,0, '<=', E1(i,J-1));
        mp.addcstr({1,{i,J}},0,0,0,{-1, {i,1}},0,{-M, {i,J}},0,0,0,0, '<=', E1(i,J));
        mp.addcstr(0,{1,{i,J}},0,0,{-1, {i,1}},{M, {i,J}},0,0,0,0,0, '>=', S1(i,J-1));
        mp.addcstr(0,{1,{i,J}},0,0,{-1, {i,1}},0,{M, {i,J}},0,0,0,0, '>=', S1(i,J));
        mp.addcstr(0,{1,{i,J}},0,0,{-1, {i,1}},{-M, {i,J}},0,0,0,0,0, '<=', E1(i,J-1));
        mp.addcstr(0,{1,{i,J}},0,0,{-1, {i,1}},0,{-M, {i,J}},0,0,0,0, '<=', E1(i,J));
 end

%Travel time constraints in outbound direction
for i = 2:I
 for j = 1:J
        mp.addcstr({[1 -1],{[i i-1],j}},0,0,0,0,0,0,0,0,0,0, '=', TT(i,1));
        mp.addcstr(0,{[1 -1],{[i i-1],j}},0,0,0,0,0,0,0,0,0, '=', TT(i,1));
 end
end
```

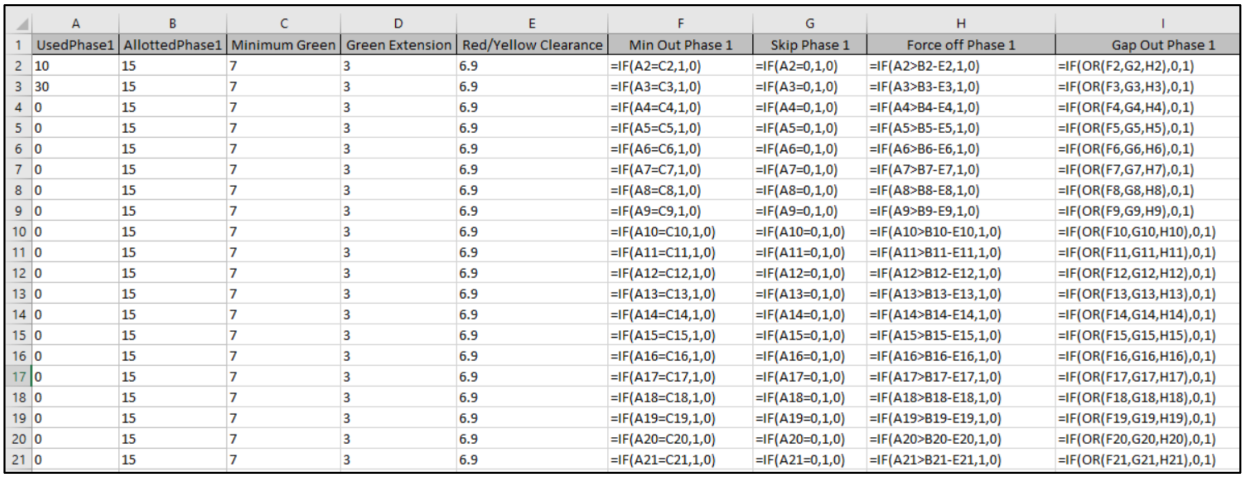
```
% Add constraints for 1 out of 3 constraints – intersection 2 to I and cycles 1 to J-1
for i = 2:I
   for j = 1:J-1
       mp.addcstr(0,0,0,0,0,{1,{i,j}},{1,{i,j}},{1,{i,j}},0,0,0, '=', 2);

   end
end

% One out of two constraints for first cycle – intersection 2 to I
for i = 2:I
       mp.addcstr(0,0,0,0,0,0,{1,{i,1}},{1,{i,1}},0,0,0, '=', 1);

end

% Unwanted offsets to 0
for j=2:J
   for i=1:I
      mp.addcstr(0,0,0,0,{[1],{i,j}},0,0,0,0,0,0, '=', 0);

   end
end

% Start and end of bands constraints for last intersection in inbound direction
for j = 1:J
      mp.addcstr(0,0,{1,{I,j}},0,{-1, {I,1}},0,0,0,0,0,0, '>=', S2(I,j));
      mp.addcstr(0,0,{1,{I,j}},0,{-1, {I,1}},0,0,0,0,0,0, '<=', E2(I,j));
      mp.addcstr(0,0,0,{1,{I,j}},{-1, {I,1}},0,0,0,0,0,0, '>=', S2(I,j));
      mp.addcstr(0,0,0,{1,{I,j}},{-1, {I,1}},0,0,0,0,0,0, '<=', E2(I,j));
end

% Start and end of bands constraints in inbound direction for intersections 1 to I-1 and cycles 2
to J-1
for i = 1:I-1
 for j = 2:J-1
      mp.addcstr(0,0,{1,{i,j}},0,{-1, {i,1}},0,0,0,{M, {i,j}},0,0, '>=', S2(i,j-1));
      mp.addcstr(0,0,{1,{i,j}},0,{-1, {i,1}},0,0,0,0,{M, {i,j}},0, '>=', S2(i,j));
      mp.addcstr(0,0,{1,{i,j}},0,{-1, {i,1}},0,0,0,0,0,{M, {i,j}}, '>=', S2(i,j+1
      mp.addcstr(0,0,{1,{i,j}},0,{-1, {i,1}},0,0,0,{-M, {i,j}},0,0, '<=', E2(i,j-1));
      mp.addcstr(0,0,{1,{i,j}},0,{-1, {i,1}},0,0,0,0,{-M, {i,j}},0, '<=', E2(i,j));
      mp.addcstr(0,0,{1,{i,j}},0,{-1, {i,1}},0,0,0,0,0,{-M, {i,j}}, '<=', E2(i,j+1));
      mp.addcstr(0,0,0,{1,{i,j}},{-1, {i,1}},0,0,0,{M, {i,j}},0,0, '>=', S2(i,j-1));
      mp.addcstr(0,0,0,{1,{i,j}},{-1, {i,1}},0,0,0,0,{M, {i,j}},0, '>=', S2(i,j));
      mp.addcstr(0,0,0,{1,{i,j}},{-1, {i,1}},0,0,0,0,0,{M, {i,j}}, '>=', S2(i,j+1));
      mp.addcstr(0,0,0,{1,{i,j}},{-1, {i,1}},0,0,0,{-M, {i,j}},0,0, '<=', E2(i,j-1));
      mp.addcstr(0,0,0,{1,{i,j}},{-1, {i,1}},0,0,0,0,{-M, {i,j}},0, '<=', E2(i,j));
      mp.addcstr(0,0,0,{1,{i,j}},{-1, {i,1}},0,0,0,0,0,{-M, {i,j}}, '<=', E2(i,j+1));
 end
end
 % Start and end of bands constraints in inbound direction for last cycle
for i = 1:I-1
        mp.addcstr(0,0,{1,{i,J}},0,{-1, {i,1}},0,0,0,{M, {i,J}},0,0, '>=', S2(i,J-1));
        mp.addcstr(0,0,{1,{i,J}},0,{-1, {i,1}},0,0,0,0,{M, {i,J}},0, '>=', S2(i,J));
        mp.addcstr(0,0,{1,{i,J}},0,{-1, {i,1}},0,0,0,{-M, {i,J}},0,0, '<=', E2(i,J-1));
        mp.addcstr(0,0,{1,{i,J}},0,{-1, {i,1}},0,0,0,0,{-M, {i,J}},0, '<=', E2(i,J));
        mp.addcstr(0,0,0,{1,{i,J}},{-1, {i,1}},0,0,0,{M, {i,J}},0,0, '>=', S2(i,J-1));
        mp.addcstr(0,0,0,{1,{i,J}},{-1, {i,1}},0,0,0,0,{M, {i,J}},0, '>=', S2(i,J));
        mp.addcstr(0,0,0,{1,{i,J}},{-1, {i,1}},0,0,0,{-M, {i,J}},0,0, '<=', E2(i,J-1));
        mp.addcstr(0,0,0,{1,{i,J}},{-1, {i,1}},0,0,0,0,{-M, {i,J}},0, '<=', E2(i,J));
end

% Start and end of green times in inbound direction for first cycle
for i = 1:I-1
        mp.addcstr(0,0,{1,{i,1}},0,{-1, {i,1}},0,0,0,0,{M, {i,1}},0, '>=', S2(i,1));
        mp.addcstr(0,0,{1,{i,1}},0,{-1, {i,1}},0,0,0,0,0,{M, {i,1}}, '>=', S2(i,2));
        mp.addcstr(0,0,{1,{i,1}},0,{-1, {i,1}},0,0,0,0,{-M, {i,1}},0, '<=', E2(i,1));
        mp.addcstr(0,0,{1,{i,1}},0,{-1, {i,1}},0,0,0,0,0,{-M, {i,1}}, '<=', E2(i,2));
        mp.addcstr(0,0,0,{1,{i,1}},{-1, {i,1}},0,0,0,0,{M, {i,1}},0, '>=', S2(i,1));
        mp.addcstr(0,0,0,{1,{i,1}},{-1, {i,1}},0,0,0,0,0,{M, {i,1}}, '>=', S2(i,2));
        mp.addcstr(0,0,0,{1,{i,1}},{-1, {i,1}},0,0,0,0,{-M, {i,1}},0, '<=', E2(i,1));
        mp.addcstr(0,0,0,{1,{i,1}},{-1, {i,1}},0,0,0,0,0,{-M, {i,1}}, '<=', E2(i,2));
end

% Add constraints for 1 out of 3 constraints – intersection 1 to I and cycles 2 to J-1
for i = 1:I
    for j = 2:J-1
        mp.addcstr(0,0,0,0,0,0,0,0,{1,{i,j}},{1,{i,j}},{1,{i,j}}, '=', 2);
    end
end

% Add constraints for 1 out of 3 constraints – intersection 1 to I and cycle 1
for i = 1:I
        mp.addcstr(0,0,0,0,0,0,0,0,0,{1,{i,1}},{1,{i,1}}, '=', 1);
    mp.addcstr(0,0,0,0,0,0,0,0,0,{1,{i,J}},{1,{i,J}}, '=', 1);
end

% Travel time constraint in inbound direction
for i = 2:I
 for j = 1:J
        mp.addcstr(0,0,{[1 -1],{[i-1 i],j}},0,0,0,0,0,0,0,0,0, '=', TT(i,1));
        mp.addcstr(0,0,0,{[1 -1],{[i-1 i],j}},0,0,0,0,0,0,0,0, '=', TT(i,1));
 end
end
```

```
 % Defining lower and upper bounds
for i = 1:I
    for j = 1:J
        mp.addlb(0,0,0,0,lower,0,0,0,0,0,0)
        mp.addub(inf,inf,inf,inf,upper,1,1,1,1,1,1)
    end
end

% Defining variable types
mp.addctype('C','C', 'C', 'C', 'I', 'B', 'B', 'B', 'B', 'B', 'B');
lp = mp.milp2ilp;

% Finally solve using intlinprog
[x,fval,exitflag] = intlinprog(lp{:});

% Writing Output Matrices

j=1;
temp = 0;
while j<=J
    for i =1:I
    X1(i,j) = x(temp+i);
    end
    temp = temp+I;
    j = j+1;
end

 j=1;
while j<=J
    for i =1:I
    Y1(i,j) = x(temp+i);
    end
    temp = temp+I;
    j = j+1;
 end

 j=1;
while j<=J
    for i =1:I
    X2(i,j) = x(temp+i);
    end
    temp = temp+I;
    j = j+1;
 end
 j=1;
while j<=J
    for i =1:I
    Y2(i,j) = x(temp+i);
    end
    temp = temp+I;
    j = j+1;
 end

for i=1:I
    O(i,1) = x(temp+i);
end

BW1 = Y1-X1;
BW2 = Y2-X2;
BW1 = BW1(1,:);
BW2 = BW2(1,:);
Total_BW = sum(BW1+BW2,2);

% Drawing Time-Space diagrams

% Updated Green times
O=repmat(O,1,J);
S1 = S1 + O;
S2 = S2 + O;
E1 = E1 + O;
E2 = E2 + O;
t1 = datetime(0000,0,0,0,0,0,'TimeZone','America/New_York');

% Outbound Bandwidths
d = [0 ; d12; d12+d23];
d=repmat(d,1,size(X1,2));
new_X1 = repmat(t1,size(X1));
new_X1 = new_X1+ X1/86400;
new_Y1 = repmat(t1,size(Y1));
new_Y1 = new_Y1+ Y1/86400;

XY1 =  repmat(t1,2,J);
XY1 = XY1+ [S1(1,:); E1(1,:)]/86400;
d1 = zeros(size(XY1));
XY2 =  repmat(t1,2,J);
XY2 = XY2+ [S1(2,:); E1(2,:)]/86400;
d2 = ones(size(XY2))*d12;
XY3 =  repmat(t1,2,J);
XY3 = XY3+ [S1(3,:); E1(3,:)]/86400;
d3 = ones(size(XY3))*(d23+d12);
```

```
 h(1) =figure;
plot(new_X1,d, 'g')
hold on
plot(new_Y1,d, 'g')
hold on
plot(XY1,d1, 'b')
hold on
plot(XY3,d3, 'b')
hold on
plot(XY2,d2, 'b')

% Inbound Direction
X2t=X2;
Y2t=Y2;
X2t = flipud(X2t);
Y2t = flipud(Y2t);
S2 = flipud(S2);
E2 = flipud(E2);
d = [d23+d12+10; d12+10; 0];

new_X2 = repmat(t1,size(X2t));
new_X2 = new_X2+ X2t/86400;
new_Y2 = repmat(t1,size(Y2t));
new_Y2 = new_Y2+ Y2t/86400;

d=repmat(d,1,size(new_X2,2));

XY1 =  repmat(t1,2,J);
XY1 = XY1 + [S2(1,:); E2(1,:)]/86400;
d1 = ones(size(XY1))*d23+d12+10;

XY2 =  repmat(t1,2,J);
XY2 = XY2+ [S2(2,:); E2(2,:)]/86400;
d2 = ones(size(XY2))*d12+10;

XY3 =  repmat(t1,2,J);
XY3 = XY3+ [S2(3,:); E2(3,:)]/86400;
d3 = zeros(size(XY3))+10;

hold on
plot(new_X2,d, 'm')
hold on
plot(new_Y2,d, 'm')
hold on
plot(XY1,d1, 'k')
hold on
```

```
plot(XY3,d3, 'k')
hold on
plot(XY2,d2, 'k')
hold on
xlabel('TimeStamp');
ylabel('Distance (ft)');
hold on
datetick('x', 'HH:MM PM')
hold on
t = datetime('now')
title('Time Space Diagram - Real-World 3 Intersection Site')

% Printing optimal Offsets
dim = [0.125 .88 .8 .04];
desc =['Optimal Offsets - ' num2str(O(1,1)) ' ,' num2str(O(2,1)) ' ,' num2str(O(3,1)) '| Total
Bandwidth - ' num2str(Total_BW)]
annotation('textbox',...
    dim,...
    'String',desc,...
    'FontSize',12,...
    'FontName','Arial',...
    'LineStyle','None',...
        'LineWidth',1,...
    'BackgroundColor',[1  1 1],...
    'Color','k');
t = datetime('now');
Name=[filename,'_Optimal Bandwidth_',datestr(now, 'dd-mmm-yyyy HH_MM'),'_k_',
num2str(k),'.png'];
xlabel('Bandwidth (sec)');
saveas(h(1),Name);

% Saving data as struct
result.X1 = X1;
result.X2 = X2;
result.Y1 = Y1;
result.Y2 = Y2;
result.BW1 = BW1;
result.BW2 = BW2;
save(filename,'-struct','result')
```

APPENDIX C: Example EXCEL Spreadsheet Calculations for Signal Performance Metrics from Split Monitor Data



	A	B	C	D	E	F	G	H	I
1	UsedPhase1	AllottedPhase1	Minimum Green	Green Extension	Red/Yellow Clearance	Min Out Phase 1	Skip Phase 1	Force off Phase 1	Gap Out Phase 1
2	10	15	7	3	6.9	=IF(A2=C2,1,0)	=IF(A2=0,1,0)	=IF(A2>B2-E2,1,0)	=IF(OR(F2,G2,H2),0,1)
3	30	15	7	3	6.9	=IF(A3=C3,1,0)	=IF(A3=0,1,0)	=IF(A3>B3-E3,1,0)	=IF(OR(F3,G3,H3),0,1)
4	0	15	7	3	6.9	=IF(A4=C4,1,0)	=IF(A4=0,1,0)	=IF(A4>B4-E4,1,0)	=IF(OR(F4,G4,H4),0,1)
5	0	15	7	3	6.9	=IF(A5=C5,1,0)	=IF(A5=0,1,0)	=IF(A5>B5-E5,1,0)	=IF(OR(F5,G5,H5),0,1)
6	0	15	7	3	6.9	=IF(A6=C6,1,0)	=IF(A6=0,1,0)	=IF(A6>B6-E6,1,0)	=IF(OR(F6,G6,H6),0,1)
7	0	15	7	3	6.9	=IF(A7=C7,1,0)	=IF(A7=0,1,0)	=IF(A7>B7-E7,1,0)	=IF(OR(F7,G7,H7),0,1)
8	0	15	7	3	6.9	=IF(A8=C8,1,0)	=IF(A8=0,1,0)	=IF(A8>B8-E8,1,0)	=IF(OR(F8,G8,H8),0,1)
9	0	15	7	3	6.9	=IF(A9=C9,1,0)	=IF(A9=0,1,0)	=IF(A9>B9-E9,1,0)	=IF(OR(F9,G9,H9),0,1)
10	0	15	7	3	6.9	=IF(A10=C10,1,0)	=IF(A10=0,1,0)	=IF(A10>B10-E10,1,0)	=IF(OR(F10,G10,H10),0,1)
11	0	15	7	3	6.9	=IF(A11=C11,1,0)	=IF(A11=0,1,0)	=IF(A11>B11-E11,1,0)	=IF(OR(F11,G11,H11),0,1)
12	0	15	7	3	6.9	=IF(A12=C12,1,0)	=IF(A12=0,1,0)	=IF(A12>B12-E12,1,0)	=IF(OR(F12,G12,H12),0,1)
13	0	15	7	3	6.9	=IF(A13=C13,1,0)	=IF(A13=0,1,0)	=IF(A13>B13-E13,1,0)	=IF(OR(F13,G13,H13),0,1)
14	0	15	7	3	6.9	=IF(A14=C14,1,0)	=IF(A14=0,1,0)	=IF(A14>B14-E14,1,0)	=IF(OR(F14,G14,H14),0,1)
15	0	15	7	3	6.9	=IF(A15=C15,1,0)	=IF(A15=0,1,0)	=IF(A15>B15-E15,1,0)	=IF(OR(F15,G15,H15),0,1)
16	0	15	7	3	6.9	=IF(A16=C16,1,0)	=IF(A16=0,1,0)	=IF(A16>B16-E16,1,0)	=IF(OR(F16,G16,H16),0,1)
17	0	15	7	3	6.9	=IF(A17=C17,1,0)	=IF(A17=0,1,0)	=IF(A17>B17-E17,1,0)	=IF(OR(F17,G17,H17),0,1)
18	0	15	7	3	6.9	=IF(A18=C18,1,0)	=IF(A18=0,1,0)	=IF(A18>B18-E18,1,0)	=IF(OR(F18,G18,H18),0,1)
19	0	15	7	3	6.9	=IF(A19=C19,1,0)	=IF(A19=0,1,0)	=IF(A19>B19-E19,1,0)	=IF(OR(F19,G19,H19),0,1)
20	0	15	7	3	6.9	=IF(A20=C20,1,0)	=IF(A20=0,1,0)	=IF(A20>B20-E20,1,0)	=IF(OR(F20,G20,H20),0,1)
21	0	15	7	3	6.9	=IF(A21=C21,1,0)	=IF(A21=0,1,0)	=IF(A21>B21-E21,1,0)	=IF(OR(F21,G21,H21),0,1)