FHWA NJ 2006-001

Evaluation of Adaptive Control Strategies for NJ Highways

FINAL REPORT
January, 2006

Submitted by

Dr. Kaan Ozbay, Ph.D., Associate Professor

Mr. Harsh Doshi, Graduate Research Assistant
Mr. Sandeep Mudigonda, Graduate Research Assistant
Dr. Maria Boile, Ph.D., Assistant Professor

Ms. Jihong Chen, Graduate Research Assistant
Mr. Tom Bongiovanni, Graduate Research Assistant
Mr. Jignesh Shah, Graduate Research Assistant
Mr. Jyot Desai, Graduate Assistant

Dept. of Civil & Environmental Engineering
Center for Advanced Infrastructure & Transportation (CAIT)
Rutgers, The State University of New Jersey
Piscataway, NJ 08854-8014

NJDOT Research Project Manager
Mr. Karl Brodtman

In cooperation with

New Jersey
Department of Transportation
Division of Research and Technology
and
U.S. Department of Transportation
Federal Highway Administration

Disclaimer Statement

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. FHWA-NJ-2006-001	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Evaluation of Adaptive Control Strategies for NJ Highways		5. Report Date January 2006 6. Performing Organization Code CAIT/Rutgers
7. Author(s) Dr. Kaan Ozbay, Mr. Harsh Doshi, Mr. Sandeep Mudigonda, Mr. Tom Bongiovanni, Mrs. Jihong Chen, Dr. Maria Boile, Mr. Jignesh Shah, Mr. Jyot Desai		8. Performing Organization Report No. FHWA-NJ-2006-001
9. Performing Organization Name and Address 1035 Parkway Ave., Trenton, NJ 08625		10. Work Unit No. 11. Contract or Grant No.
12. Sponsoring Agency Name and Address Federal Highway Administration U.S. Department of Transportation Washington, D.C.		13. Type of Report and Period Covered Final Report 01/01/2001 - 06/30/2004 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

In this project, a prototype knowledge-based expert system (KBES) integrated with a Geographical Information System (GIS) is developed. Geomedia Pro is used as the GIS interface for data entry. A software bridge is also implemented to ensure swift data exchange between Synchro, which is a commercial signal optimization software package used by NJDOT, and the developed prototype. This integration will allow the users to transfer intersection data between Synchro and the prototype KBES tool in an efficient way. The rule base of the prototype KBES was developed using the information that exists in the literature, including surveys conducted by other researchers and the simulation studies conducted by the Rutgers research team. The performance of three distinct adaptive control strategies were assessed by using a macroscopic and a microscopic simulation tool namely, PARAMICS. Prototypes for reactive (SCOOT-like), case-based / reactive (SCATS-like) and proactive / predictive (OPAC-like algorithms), each using a different control logic, were developed. These prototypes were tested for various well-calibrated New Jersey intersections. The outcome of these simulation studies were then used to develop general rules in terms of the effectiveness of using adaptive control strategies under various network and traffic conditions. The developed rule base was implemented in Visual Basic and integrated with the developed prototype. The prototype also has the capability of performing interactive macroscopic simulation of OPAC-like, SCOOT-like, and SCATS-like control strategies given the intersection and traffic characteristics. This feature was added to the KBES system to enable the user to further analyze each individual intersection. Finally, a benefit-cost analysis function is implemented and integrated to further support the decision making process. In short, the developed tool provides the traffic engineers and decision makers with a user-friendly suite of tools for guide NJDOT engineers in identifying the most suitable intersections for adaptive control and to accurately assessing their potential benefits over existing control.

17. Key Words Adaptive Control, Highways, Traffic Signal Control Systems, Signal Devices, Control Theory, Traffic Engineering, Computerized Control Systems	18. Distribution Statement		
19. Security Classif (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of Pages 242	22. Price

Form DOT F 1700.7 (8-69)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

One of the key elements of Intelligent Transportation Systems (ITS) is traffic adaptive signal control systems that consist of:

- Hardware, namely signal, controllers, and detectors
- Software that enables communication and implementation of the traffic adaptive signal control logic.

The traffic engineering community has more than 30 years of experience with computer-controlled traffic signals. Many researchers addressed different aspects of these traffic adaptive signal control systems. The main driving force behind these efforts was the research sponsored by FHWA. After the Urban Traffic Control System (UTCS) experience sponsored by FHWA in the 1970s, the most recent incarnation of these efforts is the RT-TRACS project. (4)

Location, traffic conditions, and control strategy can all play a very important role in determining the success or failure of adaptive signal systems. In New Jersey too, agencies have been considering implementing traffic adaptive signal systems. Whereas Newark and Trenton are considering the installation of adaptive signal systems, Atlantic City and New Brunswick have already installed a system. It is clear that the Department of Transportation (DOT), counties and municipalities will continue to install traffic adaptive signal systems to alleviate some of the worsening traffic congestion problems in the State. This is why well-documented procedures are needed to assess the effectiveness of implementing theses systems before they are deployed under traffic and roadway conditions found in New Jersey. These procedures are expected to guide NJDOT engineers in identifying the need for adaptive control and accurately assessing their potential benefits compared with existing control strategies.

In this project, a prototype geographical information system (GIS) based decision support system (DSS) is developed. GeoMedia Pro is used as the GIS interface for data entry. A software bridge is also implemented to ensure data exchange between Synchro and the developed DSS prototype. This integration will allow the users to transfer intersection data between Synchro and the prototype knowledge based experts system (KBES) tool in an efficient way. The KBES was developed using the information that exists in the literature, including surveys conducted by other researchers and the simulation studies conducted by the Rutgers research Team.

A comprehensive literature search of the papers and reports describing the effectiveness and impact of traffic adaptive signal control systems was conducted. This literature research was the first step in identifying the impact of different factors in the effectiveness of traffic adaptive signal control systems. Major attention was devoted completed or on-going ITS field operational tests focused on traffic adaptive signal control systems. This was, in fact, a very important part of this research because there is a wealth of data and information available that can be used to better understand the operational aspects of traffic adaptive signal control systems.

This literature search is not limited to the operational aspects of traffic adaptive signal control systems only but it is also the review of the literature as related to expert systems, decision support systems, and traffic simulation.

Development of the Prototype Decision Support System

Using portions of the three arterials selected by NJDOT, namely, Route 10, Route 23, and Route 18, the attributes and contents of the current NJDOT traffic signal inventory was studied as applied to the development of the prototype DSS proposed for this project. After familiarization with this traffic flow database and signal inventory of NJDOT, a more general database model needed for this project was developed and implemented using GeoMedia GIS tool. This combined database contains:

- Intersection infrastructure data such as intersection geometry and signal timings
- Static traffic data for the roads and intersection.

Naturally, all this data is location-based or geographical. Thus, the database was developed as a GIS i.e., it relates the traffic signal data with geographical information. The integration of a traditional traffic flow and signal database with GIS enables NJDOT to take full advantage of GIS functions, such as advanced visualization tools, location-based search, efficient representation of the transportation network, integration of other infrastructure data such as location of power and communication lines, right-of-way information with traffic signal data, and the ease of large data set manipulation in real time including on-line traffic data. As part of this task, this database was implemented using GeoMedia Pro.

Traffic adaptive signal control systems are found to produce different results at different locations and under different traffic conditions. To assess the capabilities of the developed prototype DSS when applied to different types of signalized intersections, it is very important to understand the factors influencing the outcome of these adaptive control strategies. Thus, the candidate roadways were selected by NJDOT in such a way that they represent the variations attributed to the two major factors described below:

<u>Location-specific Factors:</u> This is one of the most difficult factors to assess due to the virtually infinite number of geometric and geographic combinations that are possible in the real world.

<u>Traffic Flow / Demand-specific Factors:</u> This is an aspect that affects the performance of adaptive traffic signals seriously because these systems are basically traffic responsive and reacting to the changes in traffic flow / demand,

saturation flows, and speeds on the roads. Various studies determined traffic flow/demand and its time-dependent fluctuations as the most important factors that affect the performance of adaptive signals. Gartner et al (1995) concludes that traffic adaptive signal control systems that cannot respond fast enough to quick changes in traffic conditions sometimes perform worse than non-adaptive systems.

In addition to the two factors mentioned above namely location and traffic specific factors, there are two more factors that will affect the performance of adaptive control systems:

<u>Hardware-specific Factors:</u> Although hardware follows well-accepted standards such as NEMA standards, there are differences related to hardware, including:

- type of controller hardware,
- location and number of traffic detectors,
- communication capabilities.

<u>Software Specific Factors:</u> This was a very important aspect of our problem that is difficult to generalize. Currently, there exist several adaptive control strategies that are used in the United States and the world. Some of the most widely used ones studied in this project are:

- OPAC
- SCOOT
- SCATS
- RHODES
- Others

In brief, four factors need to be considered in evaluating the likelihood of success of adaptive control strategies implemented at a traffic intersection. Knowing that each factor can have multiple values, the search space for the best solution appears to be quite large. One way to reduce the problem domain is to select one type of control strategy (e.g., OPAC) and focus on its evaluation under different traffic flow / demand conditions prevailing at different locations. However, after several meetings with NJDOT and as a result of an extensive literature review, the research Team decided to focus on three of the most widely used and accepted control strategies, namely, OPAC, SCOOT, and SCATS.

SCOOT has a reactive nature of control. It adapts to varying traffic by changing cycle length and phase splits in small increments. SCATS has a selective nature of control. It adapts to varying traffic by selecting a timing plan from an offline-stored library of plans that best suits to current traffic demand. OPAC, on the other hand has a proactive nature of control. It detects current traffic demand and predicts future arrivals at an intersection to select a switching strategy that will reduce delay over short time intervals in the future. Thus the three selected strategies have different concepts of controlling signal timings at an intersection. The integrated DSS model consists of four major modules shown in Figure 3.

1. Input –Output Module: A GIS, which allows the decision maker to focus in a certain area and select the study area from a detailed map. Each intersection on the GIS-based map can be connected to a graphical representation of that intersection along with other relevant information. The output, which is the selected intersections, can be shown on the same map using different colors. The decision maker can click on these intersections and look for more detailed information as to why the specific intersection has been selected by the next two modules namely, Knowledge-Based Expert System (KBES) and simulation.

2. Knowledge Based Expert System (KBES): The decision support system required in this project should be able to use both analytical and heuristic knowledge. The quite complex nature of deciding on the best individual or series of traffic intersections for the implementation of adaptive traffic control requires the use of a knowledge-based system that is different from a *traditional algorithmic approach*, which consists of “simple rules” in the form of a flowchart. In fact an expert system is quite different from algorithmic approaches because it uses the expert knowledge the way human experts make decisions.

Thus a rule-based KBES was developed using the well-known steps of expert system development process namely:

Knowledge acquisition: This step involves meeting with experts, reviewing documents in the area, and conducting simulation and site studies to acquire the required knowledge to be used in the development of the rule base. In this project, knowledge was acquired from two major sources:

- Results of previous field and simulation studies reported in the open literature.
- Paramics-based microscopic simulation model capable of simulating three adaptive signal control strategies. This was a major effort requiring the development and programming of algorithms for OPAC, SCOOT and SCATS from scratch and then integration of these into Paramics. After completion of this effort, simulations were run for calibrated intersections for different volume capacity ratios to determine the performance of each strategy and thus acquire the knowledge needed to develop the rule base.

Knowledge elucidation: This step involves processing of expert knowledge to clarify different aspects of the input acquired from the experts. The knowledge acquired in the previous step was carefully studied and categorized to develop simple general rules that are applicable to generic intersections, performing under various traffic and network conditions.

Knowledge representation: This step is the development of rules or a rule base using the expert knowledge obtained and processed previously. In this step, the

C and Visual Basic programming language to code the simple rules developed in the previous step.

Implementation: This step involves the implementation of expert rules. In this case, the rule base was developed to represent various factors related to the intersections and recommendations in the form of if –then rules. These rules were incorporated into the prototype DSS.

3. Simulation Module: KBES is used to select the best candidates for implementing one of the “adaptive control” strategies. However, the rules in the rule base are most of the time generic and more detailed analysis is always needed. To conduct this detailed analysis, a macroscopic simulation program that allows the incorporation of adaptive signal control strategies such as OPAC, SCOOT, and SCATS is developed. The simulation module is basically used to determine the effectiveness of the “adaptive control” at a very detailed level.

The results of this simulation-based evaluation are used to better determine the feasible alternatives.

4. Cost –Benefit Analysis: A benefit-cost analysis module that determines costs and benefits of implementing the selected adaptive control strategy for the selected intersection was developed.

Use and Evaluation of the Prototype DSS through Case Studies

The performance of three adaptive control strategies was assessed using a macroscopic and a microscopic simulation tool, namely, Paramics. Prototypes for reactive (SCOOT-like), Case-based / reactive (SCATS-like) and proactive / predictive (OPAC-like algorithms), each using a different control logic, were developed. These prototypes were tested for various well-calibrated intersections in New Jersey. The outcome of these simulation studies were then used to develop general rules regarding the effectiveness of using adaptive control strategies under various network and traffic conditions. The developed rule base was then implemented in Visual Basic and integrated with the prototype. The developed prototype also has the capability of performing interactive macroscopic simulation of OPAC-like, SCOOT-like, and SCATS-like control strategies, given the intersection and traffic characteristics. This feature was added to the KBES system to enable the user to further analyze each individual intersection. Finally, a benefit-cost analysis function was implemented and integrated to further support the decision making process.

Finally, a series of case studies for selected intersections from the database were conducted to evaluate the developed tool and to better understand various factors that affect the effectiveness of adaptive signal strategies for various traffic and network conditions.

INTRODUCTION

Research Problem and Background

One of the key elements of Intelligent Transportation Systems (ITS) is traffic adaptive signal systems that consist of:

1. Hardware, namely signals, controllers, and detectors
2. Software that enables communication and implementation of the traffic adaptive signal control logic

The traffic engineering community has more than 35 years of experience with computer-controlled traffic signals. Many researchers have addressed different aspects of traffic-adaptive signal control systems. The main driving force behind these efforts was the research sponsored by FHWA. Following the Urban Traffic Control System (UTCS) experience sponsored by FHWA in the 1970's, the most recent incarnation of these efforts is RT-TRACS project (31). Although potential benefits of adaptive traffic signal control strategies have long been recognized by the traffic community, the lack of dependable implementation strategies and a wide-range of performance results shown in Figure 1 created skepticism towards these systems among traffic engineers. Figure 1 shows an enlarging performance envelope for different control generations (GC) that become more traffic adaptive with the increasing number of generation. Gartner et al (64) attributes this large envelope of performance, which covers both losses and gains in performance, to the problems associated with the internal modeling accuracy and logic of the control strategies, as well as to the implementation location and traffic-specific factors. (3)

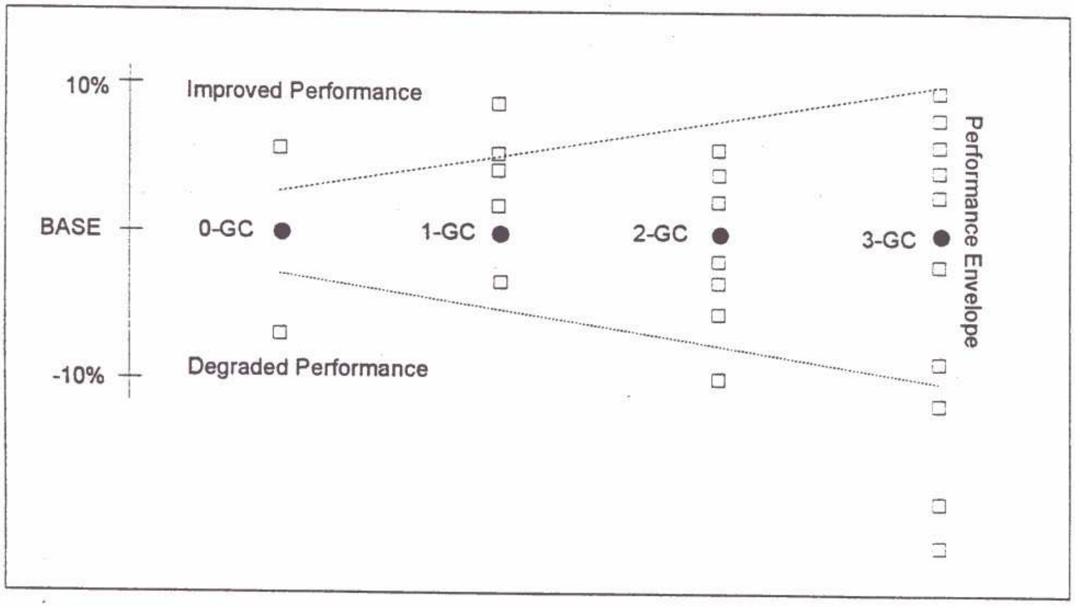


Figure 1 Relative performance of control generations (64)

Thus, location, traffic conditions, and control strategy can all play a very important role in determining the success or failure of adaptive signal systems. In New Jersey, more agencies consider implementing traffic adaptive signal systems. Whereas Newark and Trenton are considering the installation of adaptive signal systems, Atlantic City and New Brunswick have installed a system. It is clear that the Department of Transportation, counties and municipalities will continue to install traffic-adaptive signal systems to alleviate some of the worsening traffic congestion problems in the State. This is why well-documented procedures are needed to assess the effectiveness of implementing these systems before they are deployed under traffic and roadway conditions found in New Jersey. These procedures are expected to guide NJDOT engineers in identifying the need for adaptive control and accurately assessing their potential benefits compared with existing control.

Research Objectives

NJDOT is in the process of upgrading its traffic signal systems with a long-term goal of replacing the existing system with the state-of-the-art computerized adaptive signal systems. There is a growing need for devising an "*optimal site selection process*" for maximizing the return from upgrading signals selected from a quite large pool of candidate sites. Based on our discussions with NJDOT, the best candidate sites are selected based on the criteria below and input from NJDOT engineers. There are four main factors that should be considered in assessing the effects of implementing adaptive signals namely:

1. Geometric Configuration of Signalized Intersections(s): It is now a well recognized fact that the benefits of upgrading an individual or series of signals can be different depending on the location of the individual signal in the network, as well as the location of adjacent signals with respect to that individual signal. Moreover, geometric characteristics such as lane configuration, lane widths, and approach angles play an important role in affecting the benefits of adaptive signal systems.
2. Current and Projected Traffic Conditions: Current and projected traffic demand, the time-dependent fluctuations in traffic demand, saturation flows, current and projected levels of service, and other traffic related factors have a major effect on the performance of "adaptive signals"
3. Transportation Corridor in which Traffic Signals are Located: Arterial roads work in tandem with freeways and other roads as part of a larger transportation corridor. In fact, the effect of an upgraded signal within a transportation corridor, where other traffic management infrastructure, such as, Variable Message Signs (VMS), Closed Circuit Television (CCTV), and Highway Advisory Radio (HAR) exist, has to be assessed to make an optimal selection regarding which intersections must be upgraded.
4. Control Strategy: Another very important factor that directly affects the performance of the adaptive signals is the type of signal control algorithm used to change signal timing plans on-line in real-time. There are several

adaptive signal control strategies that are in use today. OPAC, RT-TRACS, RHODES, SCOOT, and SCATS are some of the most widely used ones. Operational field studies showed that each of these control strategies work varyingly under various conditions and it is important to understand the root cause of these performance differences to be able to select the most suitable strategy. For example, Moore et al (67) gives a good summary of the performance of SCOOT implemented as part of the Anaheim Traffic Control System in California. (67) They concluded that implementation of SCOOT in Anaheim was successful overall (although at certain locations the benefits were almost negligible or even non-existent).

To achieve the above goal of selecting the best candidate intersection(s) for implementing adaptive signal systems, the following research objectives are proposed:

1. Develop a computer-based decision support system (DSS) that takes into account the heuristic nature and inherent uncertainties associated with the above factors directly affecting the performance of these systems. This DSS is an integrated knowledge-based expert system, which combines expert system rules with simulation. The general functioning of the proposed approach for developing the DSS for evaluating adaptive traffic signal control systems is shown in Figure 2.

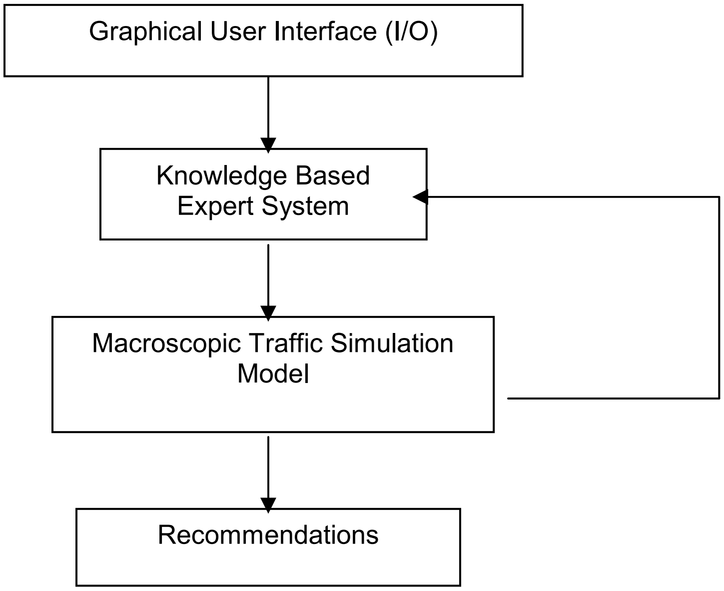


Figure 2 Implementation architecture of the proposed decision support system

To produce this DSS, develop a series of decision rules, threshold values, and interactions among these decision rules, namely an inference engine that can be implemented in the context of the proposed decision support development methodology.

2. Implement a computerized DSS to be employed by NJDOT to prioritize this selection of best upgrade candidates in an efficient and methodical way. The efficient development and usage of the developed DSS shall be ensured by:

- Identifying the input needs of the developed DSS tool.
- Assessing the availability of the input needs of this DSS tool in the NJDOT traffic signal inventory.
- Developing a plan to update the current NJDOT signal inventory according to needs of the DSS tool.
- Identifying most user-friendly graphical user interface, especially regarding input and output functions of the developed system.
- Identifying the best outputs for the developed system i.e., providing the most informative yet well-organized output to the users. A plain list of selected intersections can be too simplistic. On the other hand, we propose a flexible output system that can create assessment reports of different complexity based on the requests of the user.

Research Plan

A comprehensive literature search of the papers and reports describing the effectiveness and impact of adaptive traffic signal control systems were conducted. This literature research was the first step in identifying the impact of different factors in the effectiveness of “adaptive traffic signal control systems”. Major attention was devoted to completed or on-going ITS field operational tests focused on adaptive signal systems. This is, in fact, a very important part of this research because there is a wealth of data and information available to better understand the operational aspects of adaptive traffic control systems.

This literature search is not limited to the operational aspects of adaptive control strategies only, but it is also a review of the literature as related to expert systems, decision support systems, and traffic simulation.

Then, using portions of the three arterials selected by NJDOT, namely, Route 10, Route 23, and Route 18 the attributes and contents of the current NJDOT traffic signal inventory is studied as applied to the development of the prototype DSS proposed for this project. After familiarization with this traffic flow database and signal inventory of NJDOT, a more general database model needed for this project is developed and implemented using GeoMedia Geographical Information System tool. This combined database contains:

- Intersection infrastructure data such as intersection geometry and signal timings
- Static traffic data for the roads and intersection.

Naturally, all this data is location based, i.e. geographical. Thus, the development of the database is done in the form of a GIS, which relates the traffic signal data with geographical information. The integration of a traditional traffic flow and signal database with GIS enables NJDOT to take full advantage of GIS functions such as advanced visualization tools, location-based search, efficient representation of the transportation network, integration of other infrastructure data such location of power and communication lines, right-of-way information with traffic signal data, and the ease of large data set manipulation in real-time including on-line traffic data. As part of this task, this database was implemented using GeoMedia Pro.

Adaptive traffic signals are found to produce different results at different locations and under changing traffic conditions. In order to assess capabilities of the developed prototype decision support system when applied to different types of signalized intersections, it is very important to understand the factors influencing the outcome of these adaptive control strategies. Thus, the candidate roadways are selected by NJDOT in such a way that they represent the variations attributed to the two major factors described below:

- **Location-specific Factors:** This is one of the most difficult factors to assess due to the virtually infinite number of geometric and geographic combinations that are possible in the real world. These include:
    - Intersection geometry and configuration including number and type of lanes, and number of approaches.
    - Relative distance of neighboring intersections with respect to each other.
    - Prevailing sight and stopping distances.
    - Speed limits.

- **Factors Specific to Traffic flow / Demand:** This is an aspect which affects the performance of adaptive traffic signals seriously since these systems are basically traffic responsive reacting to the changes in traffic flow / demand, saturation flows, and speeds on the roads. Various studies determined traffic flow/demand and its time-dependent fluctuations as the most important factor that affects the performance of adaptive signals. Gartner et al (1995) concludes that adaptive control strategies that cannot respond fast enough to quick changes in traffic conditions sometimes perform worse than non-adaptive systems. (3)

Data related to the factors mentioned above were mainly obtained from NJDOT. This study was limited to 3 road sections due to the large number of intersections that needed to be modeled, studied, and analyzed to understand the differences between different signalized intersections regarding of the above factors.

In addition to the two factors mentioned above namely location and traffic specific factors, there are two more factors that will affect the performance of adaptive control systems:

a. **Hardware-Specific Factors:** Although hardware follows well-accepted standards such NEMA standards, there are differences related to hardware including:

- type of controller hardware
- location and number of traffic detectors
- communication capabilities

b. **Software-Specific Factors:** This was a very important aspect of our problem that is difficult to generalize. Currently, there exist several adaptive control strategies that are used in the United States and the world. Some of the most widely used ones studied in this project are:
   - OPAC,
   - SCOOT,
   - SCATS.

In brief there are four factors that need to be considered in evaluating the likelihood of success of traffic adaptive signal control strategies implemented at a traffic intersection. Knowing that each factor can have multiple values, the search space for the best solution appears to be quite large. One way to reduce the problem domain is to select one type of control strategy (e.g., OPAC) and focus on its evaluation under different traffic flow / demand conditions prevailing at different locations. However, after several meetings with NJDOT and as a result of an extensive literature review, the research team decided to focus on three of the most widely used and accepted control strategies, namely, OPAC, SCOOT, and SCATS.

SCOOT has a reactive nature of control. It adapts to varying traffic by changing cycle length and phase splits in small increments. SCATS has a selective nature of control. It adapts to varying traffic by selecting a timing plan from an offline-stored library of plans that suits best to current traffic demand. OPAC on the other hand has a proactive nature of control. It detects current traffic demand, and predicts future arrivals at an intersection to select a switching strategy that will reduce delay over short time intervals in the future. Thus the three selected strategies have different concepts of controlling signal timings at an intersection. The integrated DSS model consists of 4 major modules shown in Figure 3.

1. Input–Output Module: A GIS module allows the decision maker to focus on a certain area and select the study area from a detailed map. Each intersection on the GIS based map can be connected to a graphical representation of that intersection along with other relevant information. The output, which are the selected intersections can be shown on the same map using different colors and the decision maker can click on these intersections and look for more detailed information as to why the specific intersection has been selected by the next two modules namely, Knowledge-Based Expert System (KBES) and simulation modules.

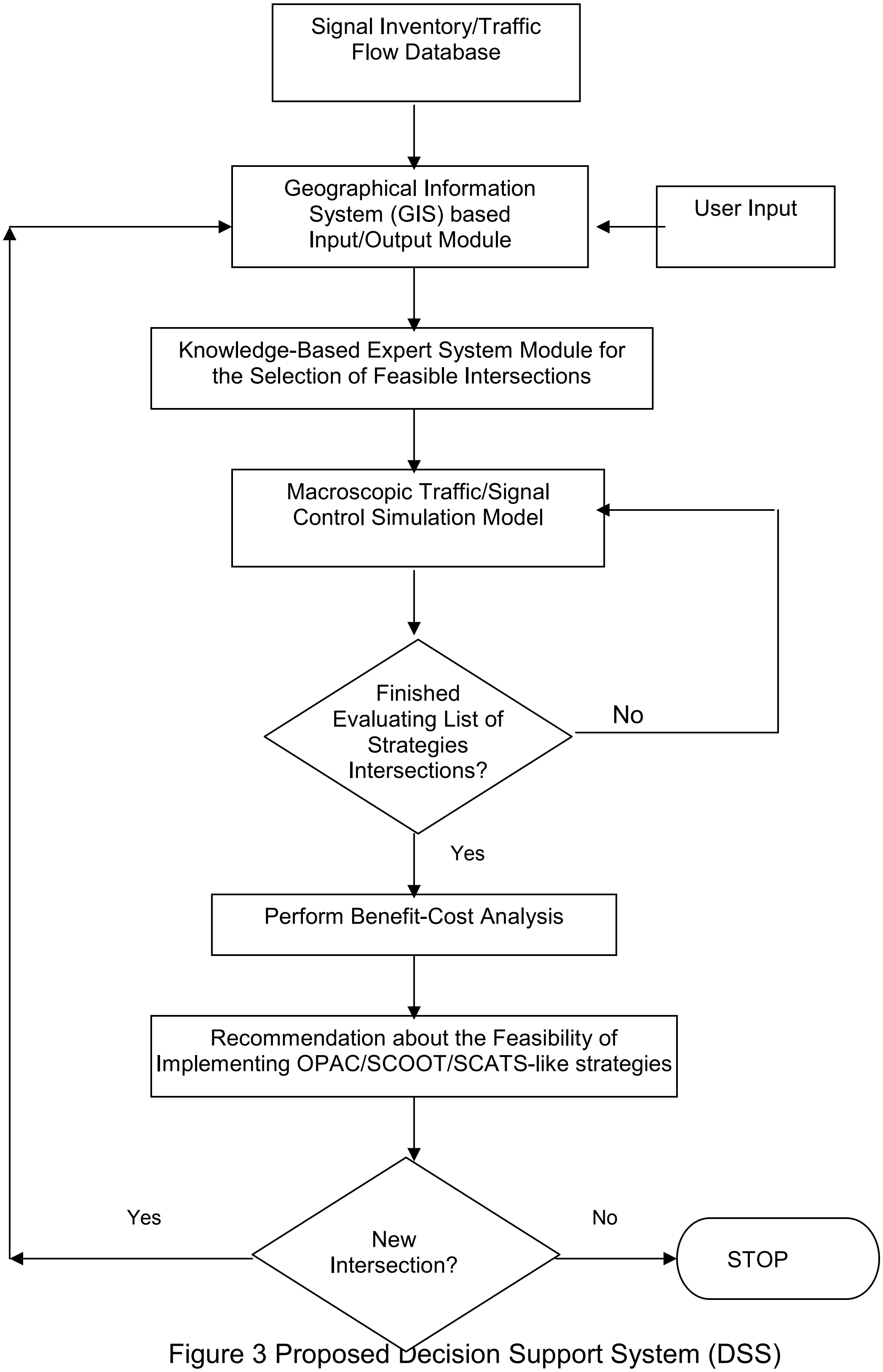


Figure 3 Proposed Decision Support System (DSS)

2. Knowledge Based Expert System (KBES): The DSS required in this project should be able to use both analytical and heuristic knowledge. The quite complex nature of deciding on the best individual or series of traffic intersections for the implementation of adaptive traffic control requires the use of a knowledge-based system that is different from a *traditional algorithmic approach*, which consists of "simple rules" in the form of a flowchart. In fact an expert system is quite different from algorithmic approaches due to the fact that it uses the expert knowledge the way human experts make decisions.

KBESs are widely used in solving engineering problems. Including diagnosis of car engine problems, design of retaining walls, safety evaluation of construction projects (73), prediction of incident clearance durations (69), etc.

In our problem domain where there can be literally thousands of intersections and large number of variables associated with each intersection, it is impossible to consider each intersection in detail. Thus, a tool that eliminates unpromising candidates is needed. This is a typical search problem. However, due to the heuristic nature of our problem, there is no closed-form performance function that can be calculated for each intersection. The selection process requires the incorporation of heuristic knowledge the way human experts employ in the case of many engineering problems. Thus a rule-based KBES is developed using the well-known steps of an expert system development process namely:

1. **Knowledge acquisition:** This step involves meeting with experts, review of documents in the area, and conducting simulation and site studies to acquire the required knowledge to be used in the development of the rule base. In this project, knowledge was acquired from two major sources:

    - Results of previous field and simulation studies reported in the open literature.
    - Paramics based microscopic simulation model capable of simulating three adaptive signal control strategies. This is a major effort requiring the development and programming of algorithms for OPAC, SCOOT and SCATS from scratch and then integration of these into Paramics. After completion of this effort, simulations were run for calibrated intersections for different volume capacity ratios to determine the performance of each strategy and thus acquire the knowledge needed to develop the rule base.

2. **Knowledge elucidation:** This step involves processing of expert knowledge to clarify different aspects of the input acquired from the experts. The knowledge acquired in the previous step was carefully studied and categorized to develop simple general rules that are applicable to generic intersections, performing under various traffic and network conditions.

3. **Knowledge representation:** This step is the development of rules or rule base using the expert knowledge obtained and processed previously. In this

step, C and Visual Basic programming language was used to code the simple rules developed in the previous step.

4. **Implementation:** This step involves the implementation of expert rules. In this case, the rule base was developed in the form of a database of rules that represent various factors related to the intersections and recommendations in the form of if –then rules. These rules were incorporated into the prototype DSS.

Another important aspect of expert systems is their ability to represent *uncertainty* explicitly. In this study domain, uncertainty is a fact due to *the large number of special cases* that exist in a problem domain as large as the arterial transportation network of New Jersey or simply due to the *incomplete information*. Thus, it is important to incorporate the effect of uncertainty when making decisions. Even early KBESs such as MYCIN / EMYCIN developed to diagnose diseases used certainty factors discussed in Ozbay et al. (69). The contribution of various factors to the outcome of a specific rule is represented using weights that depend on the certainty, which is attached to each of the factors. This is accomplished via a valuation process.

This valuation process is developed using a value function that estimates the "utility" of deploying adaptive control strategies at a certain location. This process of estimation of value function for decision making for public agencies was first proposed by Saaty. (70,71) Saaty described this process as an Analytical Hierarchy Process and proposed the estimation of the parameters or weights of a value function that ranks the alternatives using input form real-world decision makers. The proposed value function is in the following form:

$$w_1x_1 + w_2x_2 + \ldots\ldots\ldots\ldots\ldots\ldots w_nx_n = \sum_{i=1}^{n} w_ix_i \qquad (1)$$

where,

$w_i =$ weights for each variable

$x_i =$ considered performance measures from Simulations and KBES

3. Macroscopic Simulation Module: KBES selects the best candidates for implementing one of the adaptive control strategies. However, the rules are often generic and more detailed analysis is always needed. To conduct this detailed analysis, a macroscopic simulation program that allows for the incorporation of adaptive signal control strategies such as OPAC, SCOOT, and SCATS. The simulation module is basically used to determine the effectiveness of the adaptive control at a very detailed level. This type of evaluation has been recorded in the literature and is expected to produce results similar to those shown in Table 1. To ensure an interactive evaluation process, a macroscopic simulation approach to the modeling of three major adaptive control strategies studied in this project was developed.

Table 1 Evaluation results of SCOOT versus time-of-day signal plan using CORSIM [(31)]

	AM Peak	Mid-Day	PM Peak	Night	All
TRANSYT 7F	447.3	940.6	524.4	520.6	2,432.8
SCOOT	313.3	689.7	402.7	412.4	1,818.1
Percent Difference	-30%	-27%	-23%	-21%	-25%

The results of this simulation-based evaluation are used to determine the feasible alternatives better. Although the case shown in Table 1 depicts large improvements resulting from implementation of adaptive signal control, there are many cases in the literature that are not so obvious. Thus, a way of ranking the candidate feasible intersections using the “valuation module” shown in Figure 4 is needed.

4. Benefit-Cost Analysis: A benefit-cost analysis module that determines costs and benefits of implementing the selected adaptive control strategy for the selected intersection is developed.

Finally, a series of case studies for selected intersections from the database are conducted to evaluate the developed tool and to better understand various factors that affect the effectiveness of adaptive signal strategies for various traffic and network conditions. The report is concluded with a series of lessons learned and recommendations regarding the implementation of adaptive signal strategies and possible use of the developed prototype DSS tool.

LITERATURE REVIEW

Introduction

As congestion on highways increases daily, there is a strong need for the development of advanced technologies that will bring greater efficiency to highway systems in a cost-effective manner. Intelligent Transportation Systems aims at reducing congestion and travel time as well as improving safety while reducing the need to build more highway lanes. The traffic signal control systems synchronize the timing of traffic signals in an area so that the arterial's capacity is fully used. These systems vary from simple pre-timed plans to complex adaptive signal control, which optimizes timing plans for a network based on the real-time traffic conditions.

Adaptive traffic control systems can dynamically vary signal timings in response to changing traffic conditions. They can also coordinate with adjacent signals to increase the system throughput and decrease overall delay. They can also give priority to transit and emergency vehicles as part of an integrated ITS system. To support interoperability and data transfer between traffic signal control and other ITS devices, the National Transportation Communications for ITS protocol (1) is being developed. When completed, efficient data transfer with systems of adjoining jurisdictions will provide smooth flow on major corridors across jurisdictional boundaries.

The following survey of literature includes a description of the evolution of adaptive traffic control strategies, algorithms used in adaptive control strategies, and the traffic control hardware required for field implementation of adaptive control systems. The second section includes information regarding the evolution of adaptive signal algorithms. Some of the promising and widely accepted control strategies such as Split Cycle and Offset Optimization Technique (SCOOT), Sydney Coordinated Adaptive Traffic System (SCATS), Optimized Policies for Adaptive Control (OPAC), Real-time Hierarchical Optimized Distributed Effective System (RHODES) and other Real-Time Traffic Adaptive Signal Control (RT-TRACS) strategies, LADoT are discussed. The final section includes information on traffic control hardware.

Evolution of adaptive traffic control strategies

The first steps in the evolution of adaptive traffic control strategies were replacement of manual settings and optimization of signal timing plans by computer models. These models were used to optimize a performance function (such as bandwidth, total delay, and emission) using historic data and computer simulation. These models are called off-line signal timing optimization models.

The Federal Highway Administration (FHWA) developed Urban Traffic Control Systems (UTCS) in the 1970's as a part of a research project to develop and test a number of advanced traffic control strategies. It divided the control strategies into three generations. The first-generation UTCS control includes SOAP, Traffic Network Study Tool (TRANSYT), MAXBAND, PASSER II-80, PASSER III, SIGOP and Method for the Optimization of Traffic Signals in Online Controlled Networks (MOTION). Second and third generations are included in adaptive control strategies.

First Generation UTCS Control (1-GC)

The first-generation UTCS control uses signal timing plans that are calculated off-line using historic traffic data. The plans are selected on the basis of time of day, according to the operator, or by matching an existing library plan that best suits recently measured traffic conditions (if it is in traffic responsive mode). (2) In traffic responsive modes, it becomes necessary to update the plans in a specific time interval so that transition from one plan to another is smooth. The strategies included in the first-generation UTCS control are discussed below:

SOAP

It provides macroscopic analysis for individual intersections, develops timing plans with appropriate cycle lengths, and splits to minimize a performance index. (2)

TRANSYT

TRANSYT is primarily used as an off-line optimization model. However, it can also update field signal settings every minute if used in an online fashion. An iterative optimization procedure decides on signal splits and offsets for a given set of cycle lengths. The TRANSYT-7F version is commonly used in the United States (2).

MAXBAND

The MAXBAND can generate cycle lengths, offsets, speeds, and phased sequences, to optimize throughput on arterials. However, it does not incorporate bus flows. (2)

PASSER II-80

The PASSER II-80 model can optimize bandwidth by calculating the cycle length, phase sequence, and splits for linear arterials. Variations in cycle length and bandwidth as well as multiphase operation for different timing plans, are its salient features, but it does not have emissions or a fuel consumption model.

PASSER III

The PASSER III algorithm uses a macroscopic and deterministic optimization model to minimize average delay for a pre-timed arterial. It can calculate the cycle length, phase sequence, and splits. It can also calculate splits and offsets where higher throughput is desired on congested arterials.

SIGOP

The SIGOP model is primarily used to calculate the cycle length, splits, and offsets in grid networks. It uses a macroscopic simulation model to generate timing plans. It can model up to a maximum of 150 intersections with up to four signal operation phases.

MOTION

The MOTION model can optimize flow conditions and delay in a network. The cycle length is decided based on critical intersections in the network. Then alternate timing plans are generated based on turning counts in the network. Finally determining traffic streams and origin-destination (O-D) patterns generates an optimized network plan. A salient feature of this algorithm is that it gives priority to transit vehicles.

Adaptive Control Strategies

The strategies include UTCS control strategies (second and third generation); Distributed intelligence traffic control system (DITCS), which includes SCATS and TracoNet; split cycle and offset optimization technique (SCOOT); and control strategies under the RT-TRACS program of FHWA (OPAC, RHODES, RTACL, AFT-ISAC, and ARTS). Signal control is based on real-time traffic counts obtained using detectors instead of historic data. The timing plans are continuously updated to adapt to non-recurring congestions, incidents, etc.

Second-Generation UTCS Control Strategy (2-GC)

Signal timing plans are based on flow values predicted using real-time surveillance systems. They are generally updated every five minutes. However, to avoid transition disturbances, new timing plans are implemented once in ten minutes.

Third Generation UTCS Control Strategy (3-GC)

This strategy is a fully responsive online traffic control system that computes control plans to optimize a network-wide control objective using predicted traffic flow as in 2-GC. However, the update interval is smaller compared with 2-GC, and variations in cycle lengths for different signals are allowed within the update period. Table 2 shows comparison of UTCS Control Strategies.

Table 2 Comparison of UTCS control strategies (3)

FEATURE	First Generation Control (1-GC)	Second Generation Control (2-GC)	Third Generation control (3-GC)
Update interval	15 min	5-10 min	3-5 min
Control plan generation	Off-line optimization selection from a library by time of day, traffic responsive, or manual mode	Online optimization	Online optimization
Traffic prediction	None	Historically based	Smoothed current values
Cycle length determination	Fixed within each section	Fixed within variable groups of intersections	Variable in time and space Predetermined for control period

Distributed Intelligence Traffic Control System (DITCS)

DITCS sends synchronization pulses to the intersection controllers, which use control plans generated at one central control location. However, the controllers can adjust split according to the traffic conditions at the intersection. To minimize computation needs at the central control location, most of the functions are performed at the intersection level. SCATS and TracoNet are types of DITCS. SCATS is discussed in detail in Section 4.

TracoNet

This traffic responsive strategy uses a pattern-matching algorithm to effectively coordinate, control, and facilitate the flow of traffic. (2)

Real-time Traffic Adaptive Signal Control System (RT-TRACS)

FHWA commissioned the development of the RT-TRACS (4). The aim of this program is to provide guidelines, which will help in choosing the appropriate traffic control strategy based on given traffic conditions (4). Under this program five control strategies are currently being evaluated. They are: Optimized Policies of Adaptive Control (OPAC – prototype by PB Farradyne Inc.), RHODES (prototype by University of Arizona), RTACL (prototype by University of Pittsburgh/Maryland), ARTS (University of Minnesota) and ISAC-AFT. These strategies can effectively respond to rapidly varying traffic conditions by assessing the status of the network. The control is based on traffic predictions. When needed the systems can switch from one control strategy to another. The strategies are being continuously improved from a thorough understanding of past experiences with adaptive control strategies.

Split Cycle and Offset Optimization Technique (SCOOT)

This strategy was developed at the Transportation Road Research Laboratory in the United Kingdom. This strategy is most effective for networks with demands that are near saturation, near closely spaced intersections, and where traffic conditions are unpredictable. The technique adjusts the cycle length, cycle splits, and offset at a preset degree of saturation. This system is described in detail in Section 3.

Automated Traffic Surveillance and Control (ATSAC)

This strategy was developed by the City of Los Angeles and is based on one of the earliest and most expensive UTCS system. The LADoT control strategy uses loop detectors and CCTV for traffic surveillance and signal optimization software for real-time signal control (2).

Split Cycle and Offset Optimization Technique (SCOOT)

The Transportation Research Laboratory in collaboration with the U.K. Traffic System Suppliers developed SCOOT. It collects data from vehicle detectors to optimize the traffic signal setting and reduce vehicle delay and stops. SCOOT is designed to respond rapidly to varying traffic demand. It makes changes in the timing plan on a cycle-by-cycle basis, but large fluctuations in control behavior resulting from temporary changes in the traffic pattern are avoided to ensure stable operation. By avoiding generation and maintenance of expensive offline signal plans and frequent timing plan updates disruptions in the traffic network are minimized.

The SCOOT Model

SCOOT has three optimization routines: split optimization, the cycle optimization, and the offset optimization. These optimizers estimate the effect of a small change in the signal timing on a performance index, which is based on prediction of measures of effectiveness such as: delay, congestion, queue lengths, number of stops, and occupancy in the region considered. SCOOT divides the network into different regions containing signalized intersections and junctions. The regions are separated where coordination between adjacent intersections is not an issue, for example, widely spaced intersections fall into different regions.

SCOOT uses traffic detectors to respond to changes in traffic demand. Inductive loop detectors are commonly used. Detectors are usually placed at the upstream end of all links, however, the final position can vary depending on network geometry. The SCOOT algorithm uses Link Profile Unit (LPU) (5), which is created every time a vehicle passes the detector. An LPU is a combined representation of link flow and occupancy. Further, cyclic flow profiles of each LPU are generated for a specific time period, which is used for offset optimization.

Approximately 1 to 4 seconds before each update interval, the split optimizer calculates the effect of increasing, decreasing or holding the green time value. It uses the degree of saturation of all approach links on the node. In terms of the SCOOT terminology, degree of saturation is the ratio of the demand of the cyclic flow profile to the demand of the discharge rate multiplied by the duration of the effective green time. SCOOT controls the green time assigned to each node to minimize the maximum degree of saturation amongst all approach links on a node. During the control procedure, the portion of the previous cycle length that was congested is included with the calculation of degree of saturation. The link with the highest congestion is given more green time.

Cyclic flow profiles can predict the arrival time of traffic at the stop line near an intersection. This is used by the offset optimizer in SCOOT to predict the queue length at all links connected to a node. This prediction is used to minimize stops and delays in the region. Congestion on a link is also used in the offset optimizer giving the congested link priority over non-congested links.

The cycle optimizer updates the cycle lengths based on degree of saturation on each approach of every node in the network. A preset value of degree of saturation (usually 90%) is used to increment or decrement the cycle length. If the maximum degree of saturation of the node is less than 90%, the cycle length is decreased in small steps, and if the value is greater than 90%, the cycle length is increased in small steps. The cycle length is varied between a minimum practical cycle length and a maximum value determined by a critical node in the network. Highly under-saturated junctions can be made “double cycled” if such action can reduce the total network delay. A combination of relatively small changes to traffic signal timings allows SCOOT to respond to short-term local peaks in traffic demand and maintain coordination within the signal network.

SCOOT System Architecture and Hardware Requirements

SCOOT can work on both isolated-intersection and network-wide levels. Basically, SCOOT employs a centralized hierarchical, single level of control. The hardware requirements for SCOOT depend on its control type and include central computer requirements, communication hardware, and local traffic controller requirements. The hardware and software requirement, as well as the rough estimate of cost per intersection, is shown in Table 3. Figure 4 shows conceptual diagram of the SCOOT system architecture.

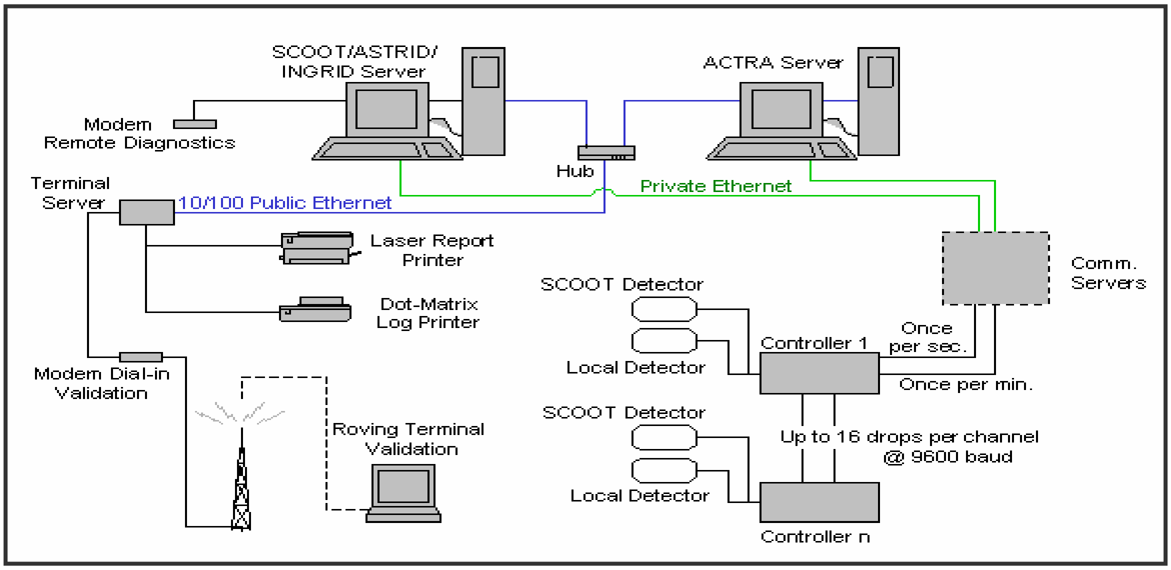


Figure 4 System architecture and hardware requirements for SCOOT (6)

Table 3 Hardware and software requirements for SCOOT (6)

Hardware	Requirements	Comments
Central computers/workstations	√	SCOOT Server - DEC Alpha workstation(s), with OpenVMS
Regional computers/workstations	√	PC running Windows 95/98/NT/2000, LAN connected via X-window Emulation, Remote dial-ins via terminal server, Interface with existing network(s)
Controllers	EPAC, 2070	If using existing controller, controller firmware needs to be upgraded (EPAC) with addition of dedicated communication Unit.
Detector requirements	√	Mainly uses Inductive loops, but other methods can also be used.
Detector location from intersection	Varies with Geometry	Usually placed at upstream end of the approach link when network geometry does not put restrictions
Data transmission (communication requirement)	Leased Line, copper cable, fiber optic or combinations	Specifications of transmission line include 1200 baud 6 drops per channel or 9600 baud 16 drops per channel. Communication Methodology includes FSK, Fixed, or Spread Spectrum Radio.

Software	Requirements	Comments
Central control software	√	
Central database management system	√	
Regional control software	√	

System Cost ($)	Cost ($)	Comments
Central hardware	30000	Cost given here is per intersection and may vary with the version of SCOOT, as well as network geometry.
Central software	N A	
Local controllers	N A	
Detectors	5000-7000	

Implementation and Experiences with SCOOT

Currently, information is available for the sites list in Table 4 of implementation:

Table 4 Implementation sites of SCOOT

Sr. No.	Agency	No. of Intersections
1	City of Anaheim, CA	22
2	Orange County, FL	13
3	City of Minneapolis, MN	56
4	City of Toronto, Canada	300
5	Santiago, Chile	300

In the above cases, SCOOT was used as a secondary system. The installation time of SCOOT at these locations varied from 3 to 18 months. One agency described the installation time as significant. The detector location in SCOOT was generally kept 7 to 14 seconds downstream of the intersection, if the geometry would allow otherwise.

The implementing agencies found the following features of SCOOT useful:

1. Automatic double cycling feature for under-saturated intersections.
2. Ability to manage traffic effectively during special events.
3. Ability to include or not include an intersection into coordination zones based on traffic demand.
4. Better offset plans that allow vehicle progression along an arterial. (This is highly desired for emergency vehicles on the arterial)
5. Ability to handle networks with a low or medium percentage of transit vehicles.

The implementing agencies were dissatisfied with SCOOT for the following reasons:

1. Inability to perform well under certain lane configurations and intersection spacing.
2. Fix phasing sequence that does not allow optimum use of bandwidth.
3. Inability to calculate offset properly if two links are feeding a downstream equally.
4. Inability to handle Protected/Permitted left-turn phasing at intersections where the vehicular gap at an opposing approach is unpredictable.

Some agencies felt that SCOOT interface needed some improvements.

Sydney Coordinated Adaptive Control System (SCATS)

SCATS is a centralized hierarchical signal control system that employs multiple-control level. The Department of Main Roads, NSW, developed SCATS. The local signal controllers in SCATS employ microprocessors, which allow communication between system entities as well as the ability to make signal status decisions at the controller level, a feature also known as distributed intelligence. The SCATS algorithm controls cycle length and green splits at an intersection and offset among adjacent intersections to minimize delay and stops.

SCATS System Description

To allow for efficient operation, SCATS divides the traffic signal network into systems and subsystems. Signal systems are separated and uncoordinated with each other, mainly because of geographical constraints. The subsystems are a group of up to 10 intersections working on cycle length. Typically, SCATS system has a central control station, which is connected with up to 32 regional computers. These regional computers are in turn connected to about 250 local controllers. Green time and splits are calculated based on detector data at each sub-system, whereas, adjacent sub-systems are linked to each other to allow for data transfer between them and facilitate vehicle progression. The typical SCATS system is shown in the Figure 5.

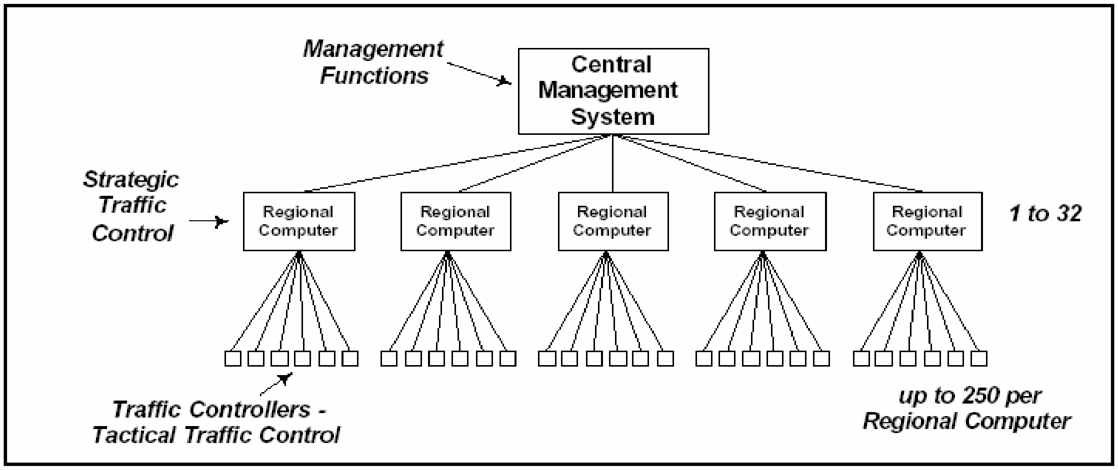


Figure 5 General SCATS architecture (11)

SCATS Algorithm

SCATS adjusts the cycle length at an intersection based on a parameter similar to the degree of saturation. It uses the ratio of effectively used green time to the total available green time for each phase at an intersection. Stop line detectors are used to measure this ratio. Length of the SCATS detector is properly calibrated because vehicles cross the stop line detectors at different speeds. Based on the measured degree of saturation, the subsystem cycle length for the next cycle is changed at the most by ± 6 seconds. Moreover, four preset cycle length values are available: minimum, medium, maximum and a cycle length where additional time is given a highly saturated phase called the *stretch* phase.

The SCATS system has a total of four available green split plans. These plans specify the normal phase sequence, which can be varied between plans and an option to transfer the unused time from one phase to another. One phase in each plan is selected as a stretch phase, which allocates more green time to the highly saturated phase. Along with the cycle length, green split is changed using a “split plan vote’ algorithm. This plan considers the phase with the highest degree of saturation. Two votes for the same plan in three consecutive cycles select that plan. The combination of changes in cycle length and split plan aims to equalize the degree of saturation on all strategic approaches.

To calculate the offset plans, SCATS uses time on the cycle length counter from zero till a selected phase is terminated. In adjacent subsystems, a critical intersection is selected and the time from zero till the termination of the selected phase is used as a reference to calculate external offsets. Five offset plans working on a specific cycle length are available to select the appropriate offset plan based on a voting scheme. This scheme selects an offset plan if it gets four out of five consecutive votes in its favor.

Additionally, sub-systems are coordinated using a link vote system. A link counter is maintained, which coordinates adjacent subsystems if it attains a count of four and breaks the coordination if its value is zero. A provision is also made to force coordination among subsystems if flow measured at “strategic” detectors exceeds preset value.

SCATS Hardware Requirements

Because SCATS employs centralized hierarchical control with multiple-control levels, the hardware, software, and communication requirements vary compared with SCOOT. Communication between the central management computer and the regional computers is based on the LAN-TCP/IP protocol or FD Data line. Communication between regional computers and controllers should support 300Bps point to point (Bell 103) or 1200/2400 Bps Multi drop (V22/V22bis (Table 5).

Table 5 Hardware requirements for SCATS (11)

Hardware	Requirements	Comments
Central computers/workstations	√	SCATS 1 DEC VAX/ALPHA, OpenVMS SCATS 2 Networked PCs
Regional computers/workstations	√	Personal computer with Windows NT Digi Serial Interface System
Controllers	Tactical	170, NEMA (Planned Controllers: SCATS 2070/2070N)
Detector requirements	Inductive Loop	
Detector location from intersection		

Software	Requirements	Comments
Central control software	√	
Central database management system	√	
Regional control software	√	

System Cost ($)	Cost	Comments
Central hardware	30000	Cost given here is per intersection and may vary with the version of SCATS, as well as network geometry.
Central software	40k – 70K	
Local controllers	4000-6000	
Detectors	5000-7000	

Implementation and Experiences with SCATS

Currently, data regarding experience with SCATS is available from the following sites of implementation listed in Table 6.

Table 6 SCATS implementation sites

Sr. No.	Agency	No. of Intersections
1	Oakland County, MI	575
2	Minnesota DOT	71

At both implementation locations, SCATS worked as a primary system. The agencies described the installation time and time to make the system operational as considerable. For example, Minnesota DOT took three years to make the system operational. Autoscope or inductive loop detectors located at stop bar were used for vehicle detection.

The implementing agencies found the following features of SCATS as useful:

1. User-friendly SCATS interface.
2. Ability to handle left turns well.
3. Ability to handle variable volumes.
4. Ability to respond to unpredicted traffic congestion.

However, because of poor set up of timing plans by the engineers in the agency, vehicle progression along the arterial was not good. Also the implementing agencies were dissatisfied with the type of controllers used.

Optimization Polices for Adaptive Control (OPAC)

OPAC is a real-time distributed adaptive signal control system. It can work as a separate controller or as a part of a coordinated system. OPAC is being developed as of the last 12 years and is undergoing verification and validation tests. The different stages in OPAC are:

- OPAC I (1979) – dynamic programming optimization.
- OPAC II (1980) – OSCO search procedure.
- OPAC III (1981) – rolling horizon approach.
- OPAC RT (1986) – real time implementation.
- OPAC IV (1995) – VFC-OPAC network model for real-time traffic adaptive control.
- OPAC V (2000 +) – proactive control integration with DTA for combined control assignment.

OPAC aims to reduce a performance function of total intersection delay by continuously adapting the signal timing plan to the current traffic demand. OPAC uses dynamic programming approach to control the green time assigned to each phase. It develops a flow profile for each phase using a user-specified horizon length. The head of the profile is the actual counts from the detectors placed upstream whereas the tail period of the horizon is predicted from historic values of traffic counts. Detectors on the link are placed according to the head period and number of stages within it. The conceptual design of the most recent OPAC algorithm is shown in Figure 6.

OPAC Algorithm

The current version of OPAC uses the rolling horizon approach to adjust the signal-timing plan and optimize a performance index. OPAC divides the network into sub-networks, which can be linked based on the level of congestion. OPAC adjusts the splits, offsets and cycle length. However, the phase sequence is not changed during operation. OPAC has different algorithms for congested and uncongested networks.

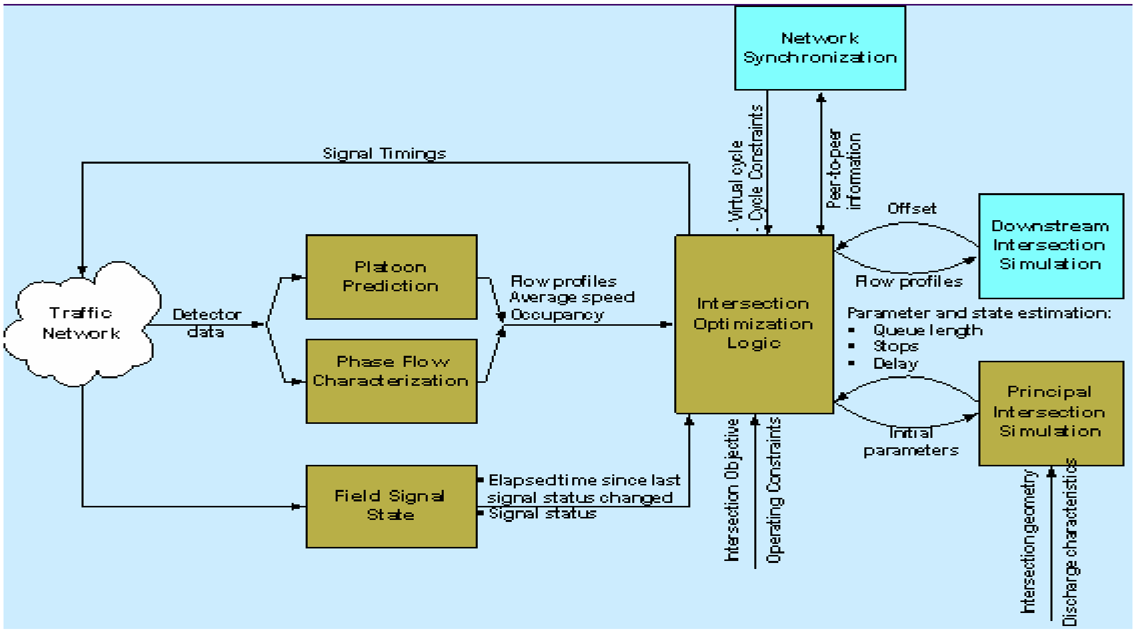


Figure 6 OPAC Conceptual Design (13).

Uncongested Control

Under uncongested control, OPAC uses distributed architecture and control traffic at each local intersection. Network-wide synchronization is achieved by using offline fixed-timed plans or by using a virtual cycle length. The virtual cycle length is selected based on a critical intersection in the network. Cycle length is changed in increments of one or two seconds to allow for a smooth transition from one cycle length to another. Phase lengths are determined using a rolling horizon technique in which the head period is typically 15 seconds, and the tail is typically 60 seconds. A change in phase is made only if the change optimizes a performance index at an intersection. A decision to change a phase is taken during the first four seconds of the head period.

To achieve a network level of control, phase changes are restricted within minimum and maximum green time values assigned to each phase. Traffic demand at each intersection is analyzed to make a decision to use either a longer or shorter cycle length or to use a virtual cycle length. This concept causes OPAC to operate similar to an actuated controller with a small window where signal-switching status is constrained between minimum and maximum green time values.

Congested Control

The main objective of congested control is to maximize the bandwidth. OPAC uses three strategies to achieve this: intersection control, signal interaction, and network control. The intersection control increases the capacity of a congested approach by giving more green time. A decision to change signal timing at adjacent intersections is taken considering the queue length on a given approach and allowing maximum throughput. The signal interaction process clears the approaches to adjacent signals by assigning a phase to a downstream signal that allows maximum number of vehicles to leave the link. Simultaneously, the opposite occurs at the upstream link. The network controller in OPAC optimizes phase lengths and offsets and achieves good vehicle progression.

OPAC Hardware Requirements

Hardware Requirements for OPAC are listed in Table 7.

Table 7 Hardware requirements for OPAC

Hardware	Requirements	Comments
Central computers/workstations	√	Two to three PCs for operator interface, server, and database.
Regional computers/workstations	√	
Controllers	ATC Nema TS2	ATC controllers such as 2070, 2070 lite, and New 170 Controllers
Detector requirements	√	
Detector location from intersection		Depends on network geometry

Software	Requirements	Comments
Central control software	√	
Central database management system	√	
Regional control software		

System Cost ($)	Cost	Comments
Central hardware	20K – 50K	Cost given here is per intersection and may vary with the version of OPAC, as well as network geometry.
Central software	100K-200K	
Local controllers	4000-6000	
Detectors	N.A.	
	[3], [7]	

Implementation and Experiences with OPAC

Currently, OPAC has been implemented and tested at the two locations listed in Table 8.

Table 8 OPAC implementation sites

Sr. No.	Agency	No. of Intersections
1	NJ Route 18, NJ DOT	12
2	Reston Parkway, Virginia	16

NJDOT found the OPAC system very efficient but difficult to manage due to hardware complexity. The system worked as a primary control system with multisoncs “OSAM” system as a backup. NJDOT had difficulties in making the system operational, because it was the organizations first attempt with fiber optic installation. Also OPAC software was being developed during the installation time. The system parameters could be easily changed in OPAC. However, it was difficult to analyze the overall system performance.

Reston Parkway network was tested with OPAC as a part of an RT-TRACS initiative of FHWA to evaluate different adaptive control systems. This was undertaken after initial simulation results of OPAC were found promising. Experiences related to installation time, detector location, etc., with OPAC are available in its evaluation report.

Real-time Hierarchical Optimized Distributed Effective System (RHODES)

RHODES is a fully distributed traffic control strategy with multiple-levels of control. RHODES uses real-time traffic demand from detectors, predicts future arrivals at different time intervals and gives optimal signal control strategy that responds efficiently to future predictions. The systems breaks the signal network into various sub networks that are connected in a hierarchical manner, predicts traffic to allow proactive control, and optimizes the performance index at various levels of hierarchy by using data structures and a computer/communication approach.

RHODES Architecture

RHODES architecture is shown in Figure 7. The highest level of control, Dynamic Network Loading, responds to slow changes in traffic conditions by predicting flow in vehicles per hour at all links in the network. The Network Flow Control assigns green time to each phase based on these flow estimates for different demand patterns. At this level of control, traffic characteristics are measured in platoons. The Intersection Level of Control uses the green time assigned by network flow control to decide on a time where the signal status should be switched. The intersection level of control is based on real-time traffic detectors at intersections. RHODES needs only one intersection located upstream on an approach link. However, stop line detectors are sometimes used to improve queue estimates.

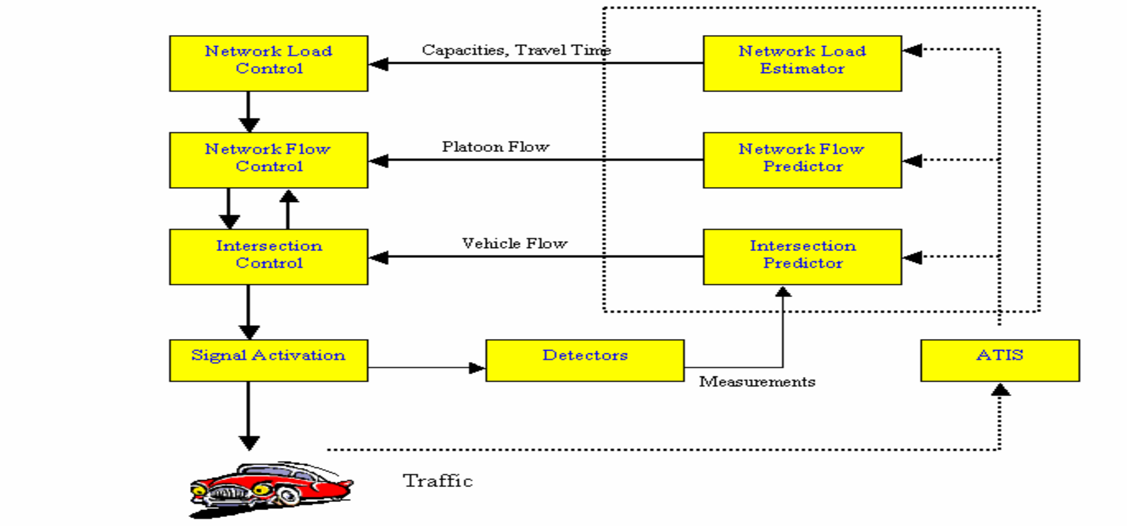


Figure 7 RHODES architecture

RHODES Algorithm

The main components of RHODES are a main controller (also called RHODES), a platoon simulator (APRES-NET), a section optimizer (REALBAND), an individual vehicle simulator (PREDICT), and a local optimizer (COP). These are connected with each other in a feed-forward and feedback structure (Figure 8).

The macroscopic simulator APRES-NET is used by REALBAND to approximate Measures of Effectiveness (MOE's) for different timing plans. It uses traffic demand obtained from detectors to estimate travel time at a downstream intersection. APRES-NET also approximates the total delay, number of stops and number of trips through the network. REALBAND, the section optimizer, is a coordination algorithm. By using this algorithm sub-optimum timing plans allowing each platoon to progress through the intersection are generated, and optimum signal phasing is then selected.

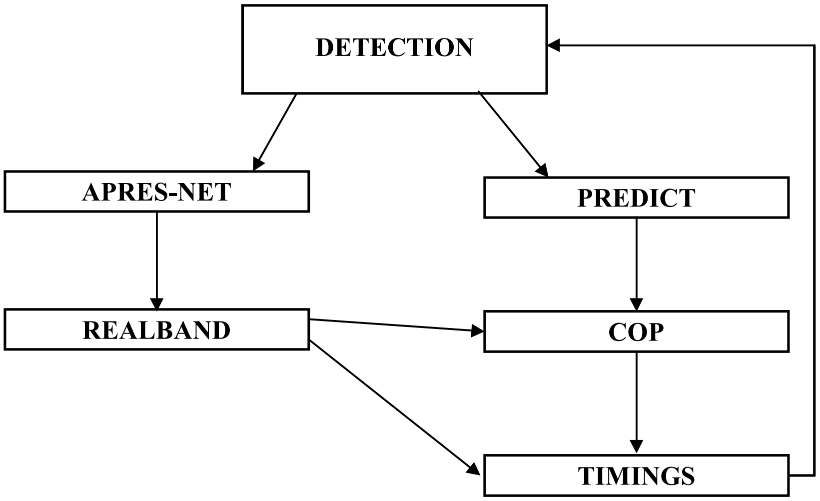


Figure 8 RHODES system

When a vehicle arrives at a detector, PREDICT estimates its arrival time at other detector locations. The output data from PREDICT is used by COP to optimize the signal plan. COP optimizes a signal-timing plan using a dynamic programming approach that uses every possible phase combination to give the best plan. The optimization process is repeated every one to two seconds. Global optimization is performed by APRES-NET and REALBAND. REALBAND creates alternative timing plans, whereas APRES_NET evaluates them. REALBAND gives enough green to pass a platoon, whereas the excess green time is allocated optimally by COP to disperse initial queues using arrival times from PREDICT. However, the optimization process at individual intersections is constrained by the platoon times from REALBAND.

Hardware Requirements for RHODES

Hardware requirements for RHODES are listed in the Table 9

Table 9 Hardware requirements for RHODES (15).

Hardware	Requirements	Comments
Central computers/workstations	√	PC-based AMTS traffic server supporting Serial Communications
Regional computers/workstations	√	Field hardened PC supporting serial communications
Controllers	2070 with VME Co-processor	
Detector requirements	√	
Detector location from intersection		

Software	Requirements	Comments
Central control software	√	
Central database management system		
Regional control software	√	

System Cost ($)	Cost	Comments
Central hardware	50000	Cost given here is per intersection and may vary with the version of RHODES, as well as network geometry
Central software	500	
Local controllers	N.A.	
Detectors	N.A.	

Implementation and Experiences with RHODES

Currently, RHODES has been implemented and tested at the two locations listed in Table 10.

Table 10 RHODES implementation sites

Sr. No.	Agency	No. of Intersections
1	City of Tuscon	-
2	City of Tempe	-

Information regarding the experience with RHODES in the city of Tuscon is not available because university authorities are providing equipment and tests bed purely for testing purposes. Other than some work being finalized via peer-to-peer communication, no running RHODES controllers are on the streets. However, simulation results show significant improvement in throughput using RHODES.

In the city of Tempe, it took two weeks to one month for the system to become operational because 2070 controller work in the TS2 mode with RHODES took the greatest amount of time. The system was able to handle variations in traffic on a cycle-by-cycle basis and traffic at places where capacity was suddenly reduced (work zones). Among the system's faults was its inability to detect upstream vehicle density and speed, as well as calculation of travel time to the stop bar, which was estimated and not measured.

Other Adaptive Control Strategies

LADOT Traffic Control System (LADOT)

Introduction

LADOT is a PC-based real-time adaptive traffic control system developed by LA DOT. A prototype system became operational in 1996, and the pc-based system using windows was completed in 1999. Currently, 375 intersections in Los Angeles are controlled using this strategy.

Methodology

Detectors are located 200' to 300' upstream of the intersections. The detectors collect volume and occupancy data every time interval. LADOT then smoothes the calculated demand and estimates the new cycle length. Calculations are based on heuristic formulas, which calculate the offsets, cycle length, and splits. Splits are calculated considering a critical intersection in the network and offsets are calculated considering a critical link in the network. Input parameters to the heuristic formulas are from volume, occupancy, detector location, queue length, cycle length limits and minimum green time for different phases. These parameters are easily set using a graphical user interface (GUI). Transit priority is incorporated using loop-transponder technology to detect buses and an algorithm that checks the bus schedule and provides green extension/red truncation for late buses. Isolated and over saturated intersections are controlled using a longer cycle length. Major arterials identify critical links and provide progression for a congested approach.

RT-TRACS Control Strategies

University of Minnesota: ARTS

This control strategy has a highly distributed architecture with both the local and network level of control. It uses a rolling horizon approach to optimize a performance index. Detectors are located at the stop bar, mid way on a link and at the upstream end of the intersection. The algorithm uses a mesoscopic simulator called PACKSIM and a section controller called CARS. PACKSIM makes predictions for the head of the horizon in the rolling horizon technique by simulating a platoon of vehicles and using either two of volume, occupancy and/or speed. CARS simulates the effects of green time extension and truncation using estimated turning counts, as well as traffic arrivals from outside the control area with an update interval of five minutes.

The algorithm uses Supervisor Decision Support System (SDSS), which manages a central database and communication among other SDSS and local area controllers. It mainly provides coordination among local area controllers determines phase sequence and manages the interface between ARTS and other functions.

Comments:

1. CARS provides alternatives for timing plan, which differs only in the roll-over period. These timings can become quickly outdated because signals are controlled in real time.
2. The algorithm assumes the new timing plan to be valid for the horizon length while varying them dynamically in each phase. Hence, if the optimal timing plans vary in the short term, they are likely to deviate from optimality in the long term as well.
3. The algorithm does not specify transition to a new recommended phase length.
4. The strategy cannot be used on congested arterials to provide progression because the latest timing plans are used in the horizon calculations as short- term plans.

ISAC: AFT

This control strategy uses a set of fixed time plans for independent networks. The signal timings are changed based on local traffic detection. Change on signal coordination is not done however, for a current signal, a change is made only if delay calculations on conflicting approaches support it. ISAC uses a fuzzy logic algorithm wherein it takes the vectors of volumes at a critical intersection. It selects the plan with minimum deviation from the vector of currently observed volume. Transition is not drastic and is accomplished within few cycles. Also, individual intersections can modify the recommended timing plans based on local detection. Phase skipping and green extension are salient features of this algorithm.

Comments:

1. Extending green time for a phase results in other phases competing for the remaining time and results in queue spill over into the next cycle.
2. This method is inappropriate for congested conditions.
3. Developers feel that the use of fuzzy logic to calculate extension of green phase every second is inappropriate and that this calculation can be accomplished using simple algorithms.

University of Maryland RTACL:

RTACL has a distributed architecture with multiple levels of control. It can be configured for two to four levels of control, with the simplest level providing control at network and local levels. It uses a macroscopic simulator that models the front and back of queues. The back of the queue is estimated using algorithms that model flow, which moves with saturation flow and then average flow when the signal is green and using the shock wave theory when a platoon reaches the end of current queue at a red signal.

Local controllers can optimize their own timing plans based on traffic demand at the intersection. Coordination is achieved by considering the flow in and out of each link. A controller using a horizon length of two cycle lengths predicts the signal timing for adjacent intersection. The controller recommends short-term changes in the current phase length and next phase. These recommendations are evaluated from 10 to 20 alternative signal-timing plans with a different optimization criterion based on the degree of saturation.

The network controller monitors communications between sections and boundaries to the control area. It operates in a supervisory mode recommending the phase sequence and green splits to optimize flows in the networks. This is accomplished using detected, as well as predicted flows.

Comments:

1. Coordination among local controllers is not well defined.
2. The network model assists local controllers without the capability of accurately representing queues.

Traffic Control Hardware

Detection

Loop Detector

The standard device for input to today's signal systems is the loop detector. As vehicles pass over the loop in the pavement, a monitoring unit registers the change in inductance of the loop, and a “call” is placed to the controller. Additional features of the monitoring unit allow for passage or presence modes of operation and the possibility of delaying a detector call. The predominance of loop detectors is attributable to their wide range of measuring capability (count, presence, speed, occupancy, and queue length) and low cost in compared with other devices. Loops also tend to be reliable over time, though periodic adjustment is required to maintain accuracy. One of the greatest shortcomings of loop detectors is that they only serve as “point” detection devices. To provide advanced information, such as queue length and speed, multiple detectors must be used. As more advanced traffic control systems require increasing amounts of information at all approaches to every intersection in a network, the number of loops necessary to meet detection requirements rapidly increases.

Video Detection

Though the technology is still being refined, video can be as accurate and reliable as loops. Because video systems contain analytical software, they can easily compute advanced information, including queue lengths, which can only be nominally monitored by loops. The typical video system contains a digital analyzer that checks camera input for contrasts with the pavement, which is the background of the image. Any contrast (filtered for shadows and weather conditions) is registered as a vehicle at a defined location. Data made available by video may include intersection turning movement counts, traffic density across lanes, automatic vehicle identification (i.e., identifying buses in the traffic stream for possible priority treatment), and incident detection. One camera and analyzer can provide all of the sensing capabilities of many loops on an approach.

Intersection Controllers

There are basically two types of intersection controllers: National Electrical Manufacturers Association (NEMA) standard controllers and those built to the Model 170 specification. There are two standards of NEMA controllers: NEMA TS1 released in 1976 and NEMA TS2 released in 1992.

NEMA TS-1: NEMA developed performance specification TS-1 for traffic-actuated controllers. These specifications describe the function of the controller and the interface between controller and cabinet. NEMA specifies the sequence of green, yellow, and red intervals for an individual traffic movement. NEMA did not specify hardware. Because the internal architecture of the controller is not specified, the manufacturer must develop software for the controllers and that would work only with the manufacturer’s product. Generally, software is not accessible to the user.

NEMA TS2: NEMA TS-2 specifications were published in 1992. They are similar to TS-1, but the functionality is defined without commanding the hardware requirements. There are two types of TS-2 controllers. The first type is a pure controller for new systems and installations, and the second type provides TS-1 cabinet interfaces to ensure backward compatibility with existing cabinets.

Model 170: The California DOT, the city of Los Angeles, the New York State DOT, and the FHWA developed an open-architecture general-purpose microcomputer for traffic control. It was developed around the same time as the NEMA TS-1. This controller uses software made by a third party, and this software is accessible by user. Because the standards also define the microprocessor to be used and its memory map independent, software developers can create the products. Model170 has been programmed so that it can be used as a traffic controller, variable message-sign controller, ramp meter, field master and other traffic-system devices. Since it does not support today's standards, model 170 has become outdated.

Experience with using NEMA TS1 standard and Model 170 controllers for advanced transportation control systems have identified several needs for controllers in the future. Primarily, the needs are uniform electrical interfaces and flexible software platforms. As the profession witnesses the move to advanced traffic control systems, it will become increasingly necessary to have field systems in place that can respond quickly to real-time conditions and controls. There is a need for open, sophisticated hardware platforms that are designed for network installation and system integration. One possibility for future transportation controllers is the VMEbus, an open architecture real-time industrial computing platform. In fact, the VMEbus platform is being used in the Advanced Transportation Controller as represented by the Caltrans/City of Los Angeles 2070 specification.

Expert Systems

Regarding adaptive signal control algorithms, it is often difficult for transportation agencies to determine which algorithm is most suited to its application network. This is partially attributable to the lack of past field implementations and also due to the lack of testing different adaptive signal control algorithms on the same types of networks. As seen in the previous sections, adaptive signal control algorithms have been implemented at various sites in the past and their performance is also reviewed in literature. However, there is no comprehensive study that shows what to expect out of an adaptive control strategy when applied to a particular network. Also, there is very less information regarding the cost effectiveness of such strategies.

Expert systems use human knowledge to solve problems in real life that would normally require human intelligence. Such systems represent the human intelligence as data or rules. These systems collect small fragments of human knowledge into a knowledge base, which is used to reason through a problem, using appropriate knowledge. Expert systems are developed as a decision support tool to make better decisions and take better action faster. They first collect the available information/knowledge from various sources and prepare a meaningful database. This database is based on a set of defined attributes that form input to the problem statement. These inputs resemble a set of conditions for which a user likes to know how a particular system would perform when given such input.

After the information is collected, it can be used for classification and prediction. Classification and prediction can be conducted using various methods such as linear regression, rule induction, decision tree, neural network etc. The database is classified based on the set of defined attributes and the output of the system is predicted if the input is a defined set of attributes.

Rule-Based Systems

It is possible to create a rule-based system using a set of assertions and a set of rules that specify how to act on the assertion set. The expert system encodes the knowledge of an expert (such as field implementation and simulation) into a rule set. When exposed to the same data, the expert system performs in a similar way to the expert. A rule-base system has its strengths and limitations that should be considered before using it. It should be applied when input-output behavior can be represented in the form of if-then rules. The system should not be applied if there are too many attributes (inputs) to the system as it becomes difficult to maintain the database and the decision process becomes more difficult to implement.

The following inputs are required for rule-based systems:

1. A set of facts that is relevant to the beginning state of the system (attributes).
2. A set of rules that comprises any and all relevant actions to be taken within the scope of a problem.
3. A condition, which determines if a solution exists or not.

A normal rule-based system begins with a rule base, which has all relevant knowledge encoded into if-then rules. The system then examines all rules and determines a subset of rules whose conditions are satisfied based on the given input. From this subset of rules, one rule is selected based on a conflict resolution strategy. When this rule is selected actions specified in the “then” part of the rule are carried out. This method of selecting rules and performing actions is repeated until there are no more rules whose conditions are satisfied or until a rule indicating system termination is selected.

The conflict resolution strategy chooses from a set of rules work on different algorithms. These are given below:

1. First Applicable: If rules are specified in a predefined order, knowing the first selected rule, other subsequent rules are automatically selected. The potential problem is that the system may go into infinite loop.
2. Random: It simply chooses a random rule. It is used widely in game theory.
3. Most Specific: This chooses a rule when most conditions specifying a rule are satisfied.
4. Least Recently Used: This chooses a rule that is least used. If all rules are needed for a solution of a given problem this is a perfect strategy.
5. “Best” Rule: This assigns ‘weights” to rules that specifies how much it should be considered compared with the alternatives. The rule with the most preferable outcome is chosen.

MODELING OF NJ ARTERIALS (NJ ROUTE 10, NJ ROUTE 23, NJ ROUTE 18)

Three Highways in New Jersey were studied in this project: NJ 10, NJ 23, and NJ 18. Each of these arterials was simulated using CorSim, Paramics, and Synchro. These networks were calibrated in each of these software packages so that the adaptive control strategy prototypes that were developed later on could be tested using these networks. Each of these networks is described below:

Route 10 Network

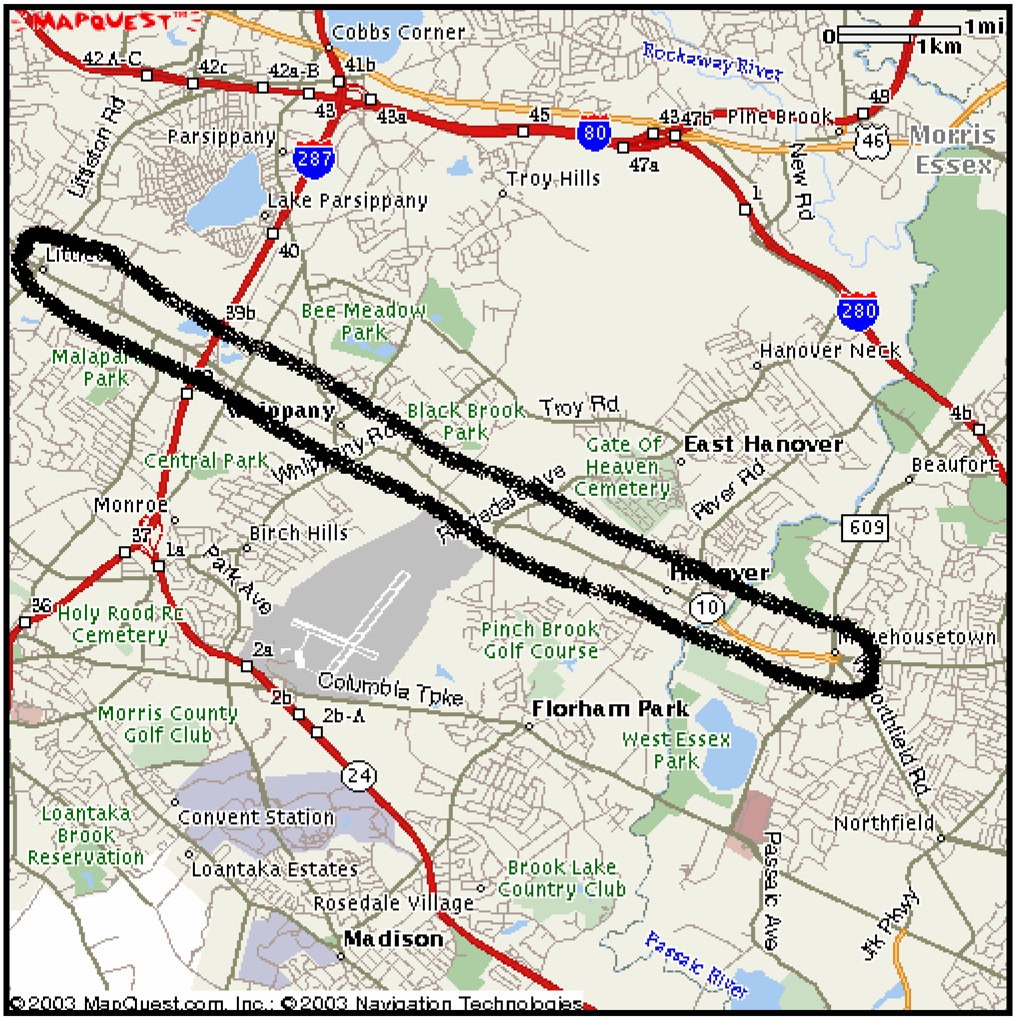
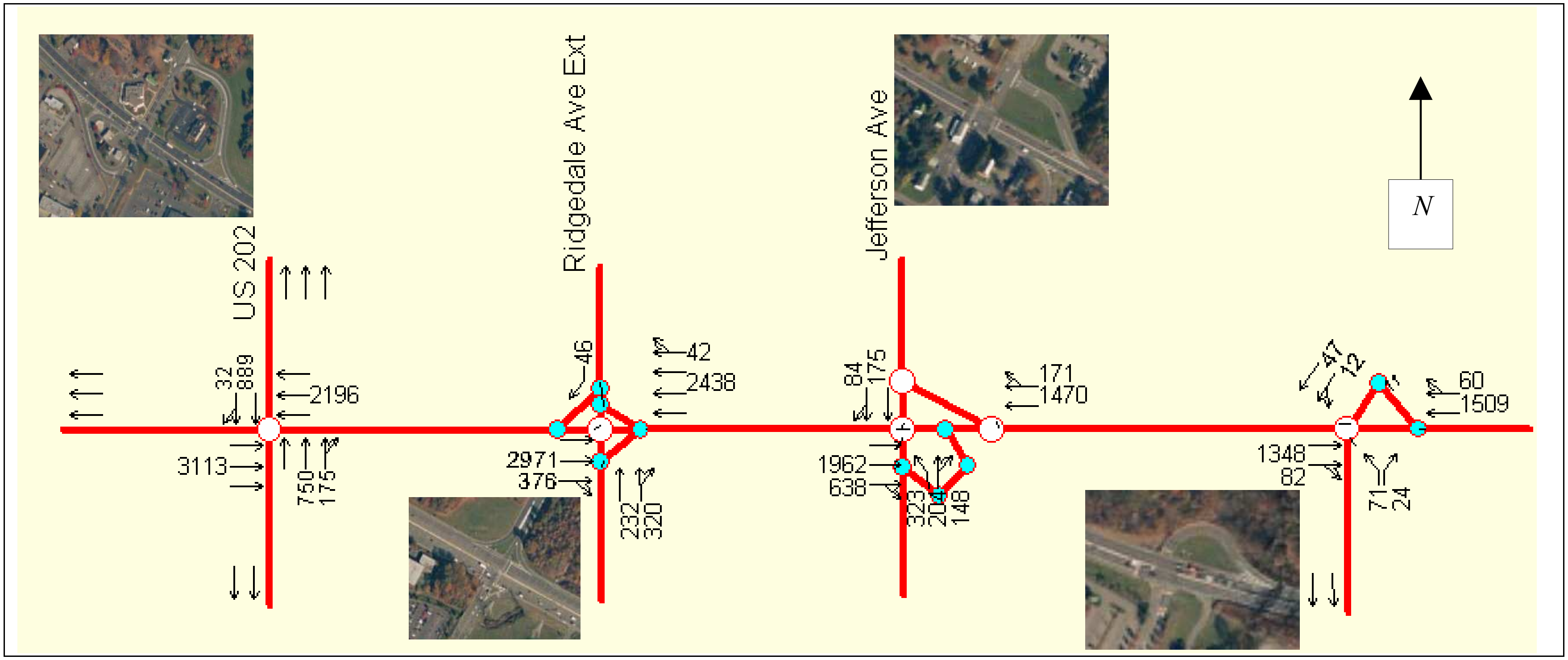


Figure 9 Route 10 study network (72)

Route 10 is a major east-west arterial located south of I-80/I-280 and north of Route 24. Route 10 network, between US 202 and the Eisenhower circle, is studied for the evaluation of the adaptive signal control strategies. Figure 9 shows the portion of network considered for the study. This roadway configuration is “grid-like” in nature and lends itself to significant alternate routing in the case of traffic incidents. Ridgedale Avenue and Eisenhower Parkway are north-south cross streets, which, in addition to I-287, connect I-280/I-80 with Route 24. There are 16 intersections on Route 10 considered for study. The Novartis complex is located in the southeastern quadrant at Ridgedale Ave. The traffic from and to this complex, combined with trips between Madison/Florham Park and East Hanover along Ridgedale Ave makes this a particularly troublesome intersection regarding capacity. Many shopping centers and strip businesses are located along both sides of Route 10 east of Ridgedale Ave to the Eisenhower Circle. Table 11 contains the length in feet of each link, between intersections. The total length of the segment is approximately 7.3 miles.

Table 11 Route 10 link lengths (feet)

Intersections	**Spacing**	**Intersections**	**Spacing**	**Intersections**	**Spacing**
US 202	Feet	Troy Hills Rd	1320	River Rd	3274
Ridgedale Ave Ext	6019	Algonquin pkwy	2270	Okener Pkwy	2481
Jefferson Rd	3960	Ridgedale Ave	2851	Walnut st	2429
Pine Plaza	2534	Jughandle	3590	Eisenhower pkwy	2006
Whippany Rd	4012	New Murray Rd	1795		

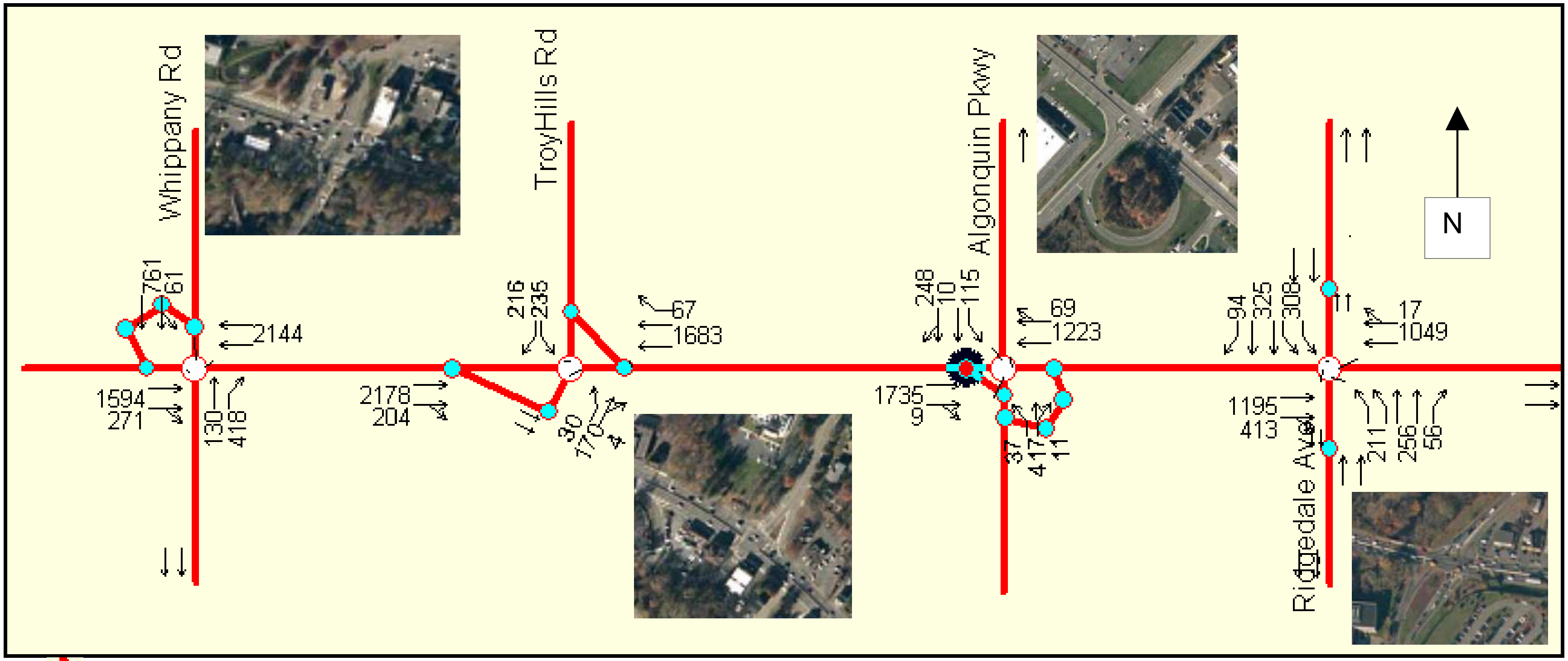


○ Signalized intersection

● Unsignalized intersection

Note: not all unsignalized intersections included in the network.

Figure 10 Route 10 study road section

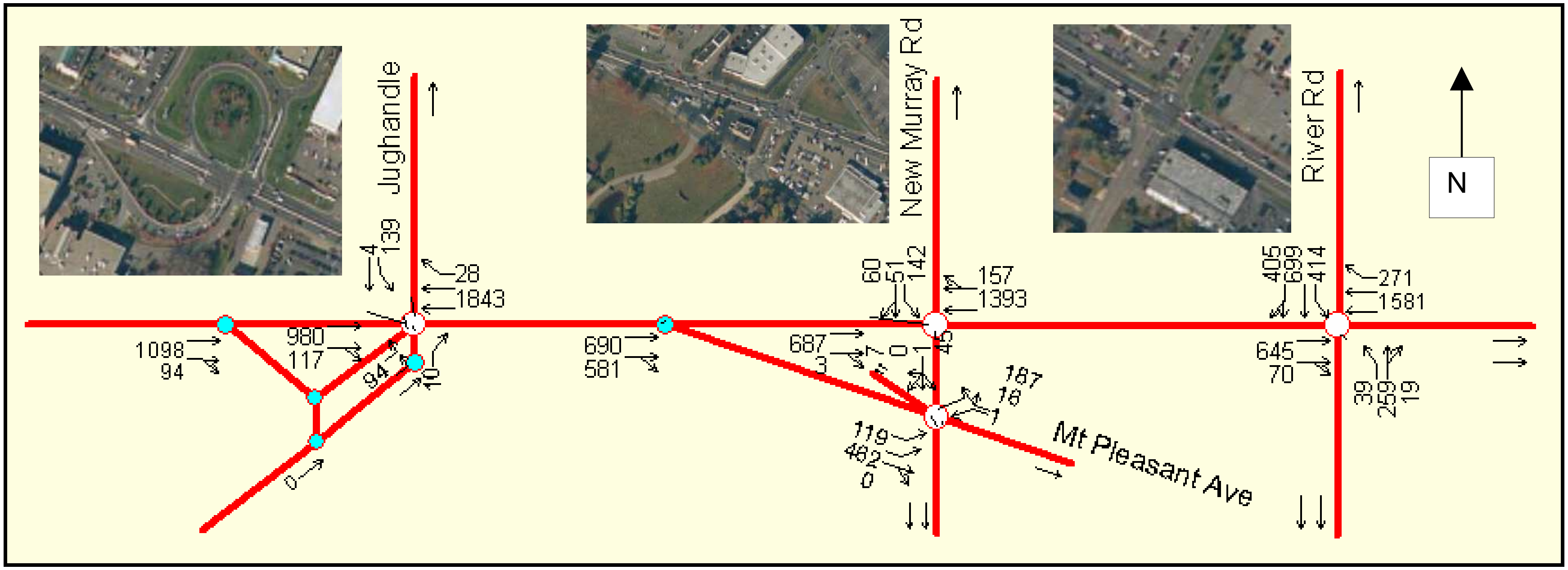


○ Signalized intersection

● Unsignalized intersection

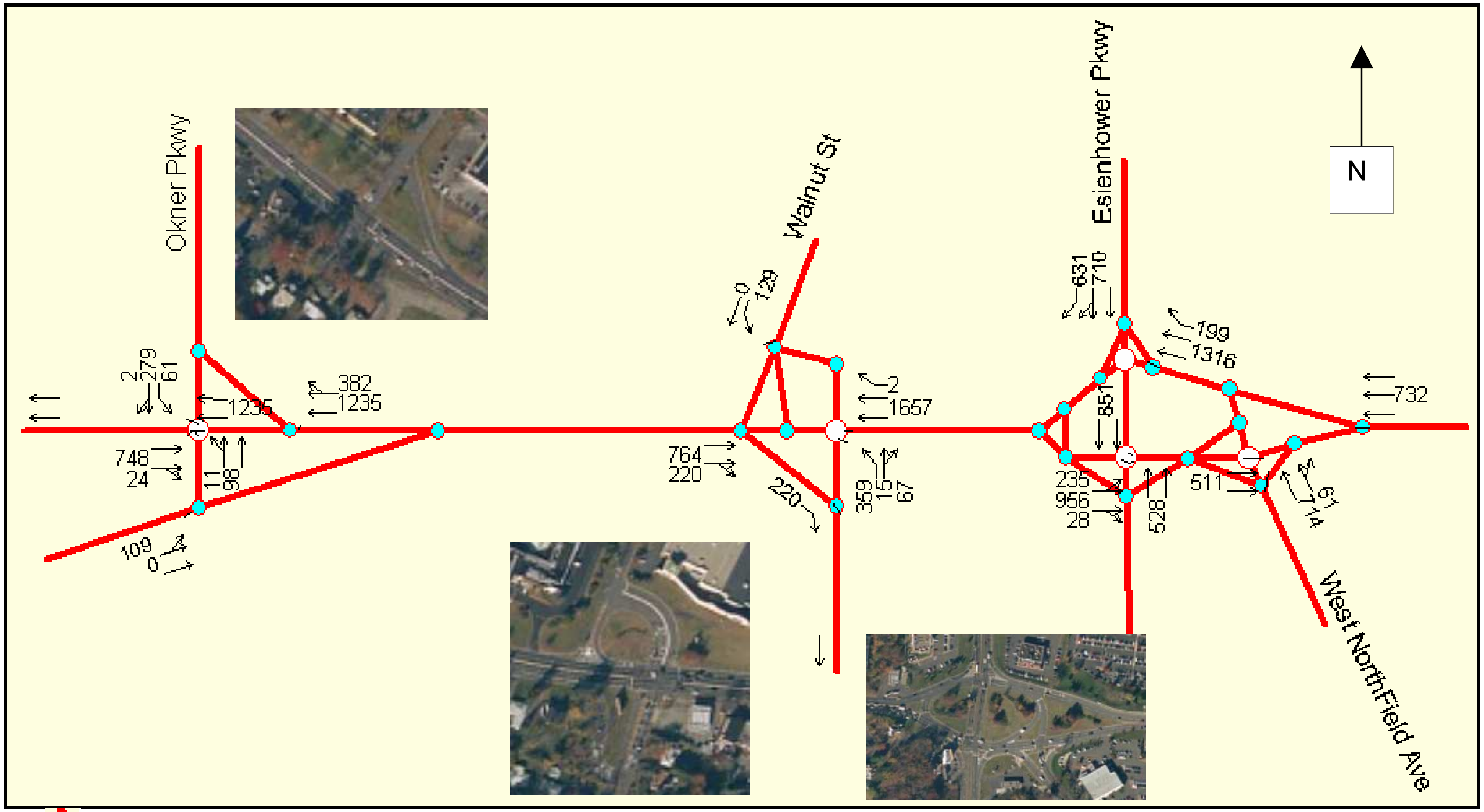
Note: not all unsignalized intersections included in the network.

Figure 11 Route 10 study road section



○ Signalized intersection
● Unsignalized intersection
Note: not all unsignalized intersections included in the network.

Figure 12 Route 10 study road section



○ Signalized intersection

● Unsignalized intersection

Note: not all unsignalized intersections included in the network.

Figure 13 Route 10 study road section

Route 23 Network

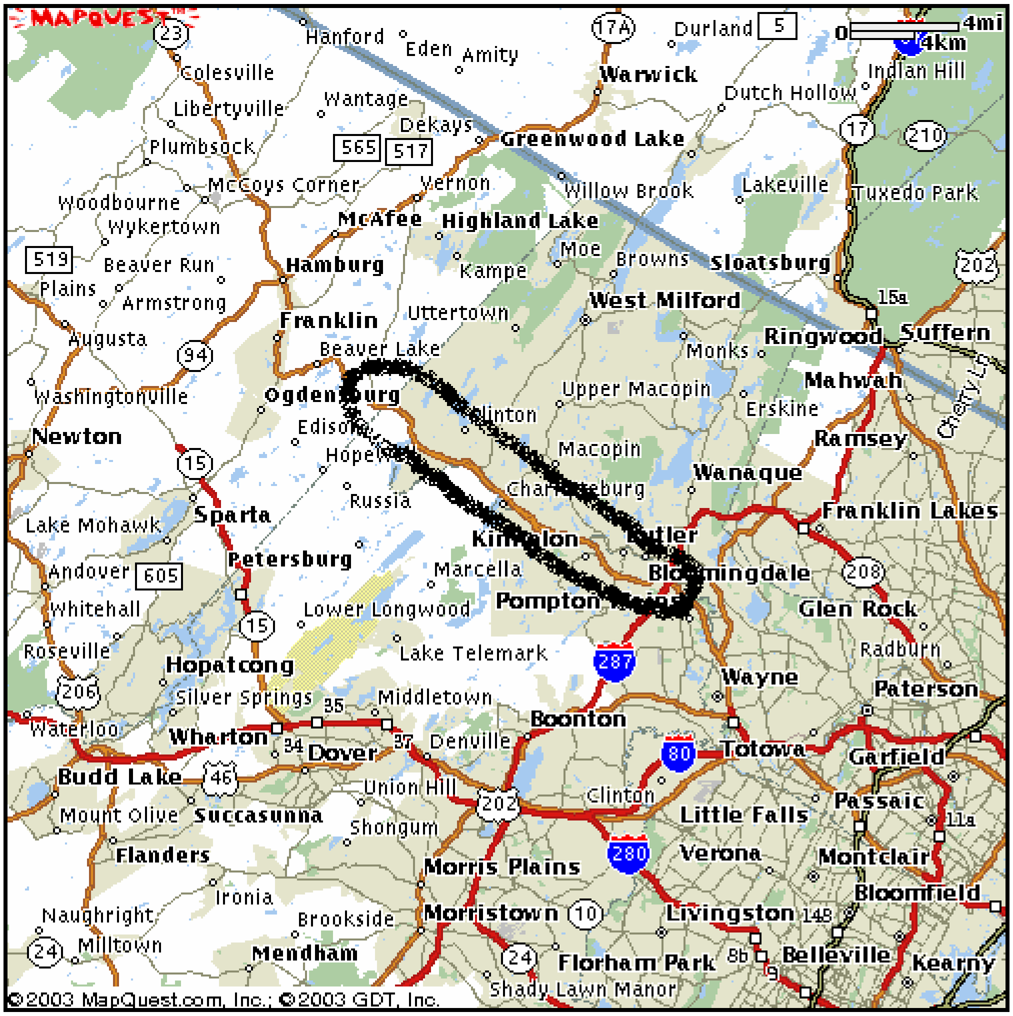
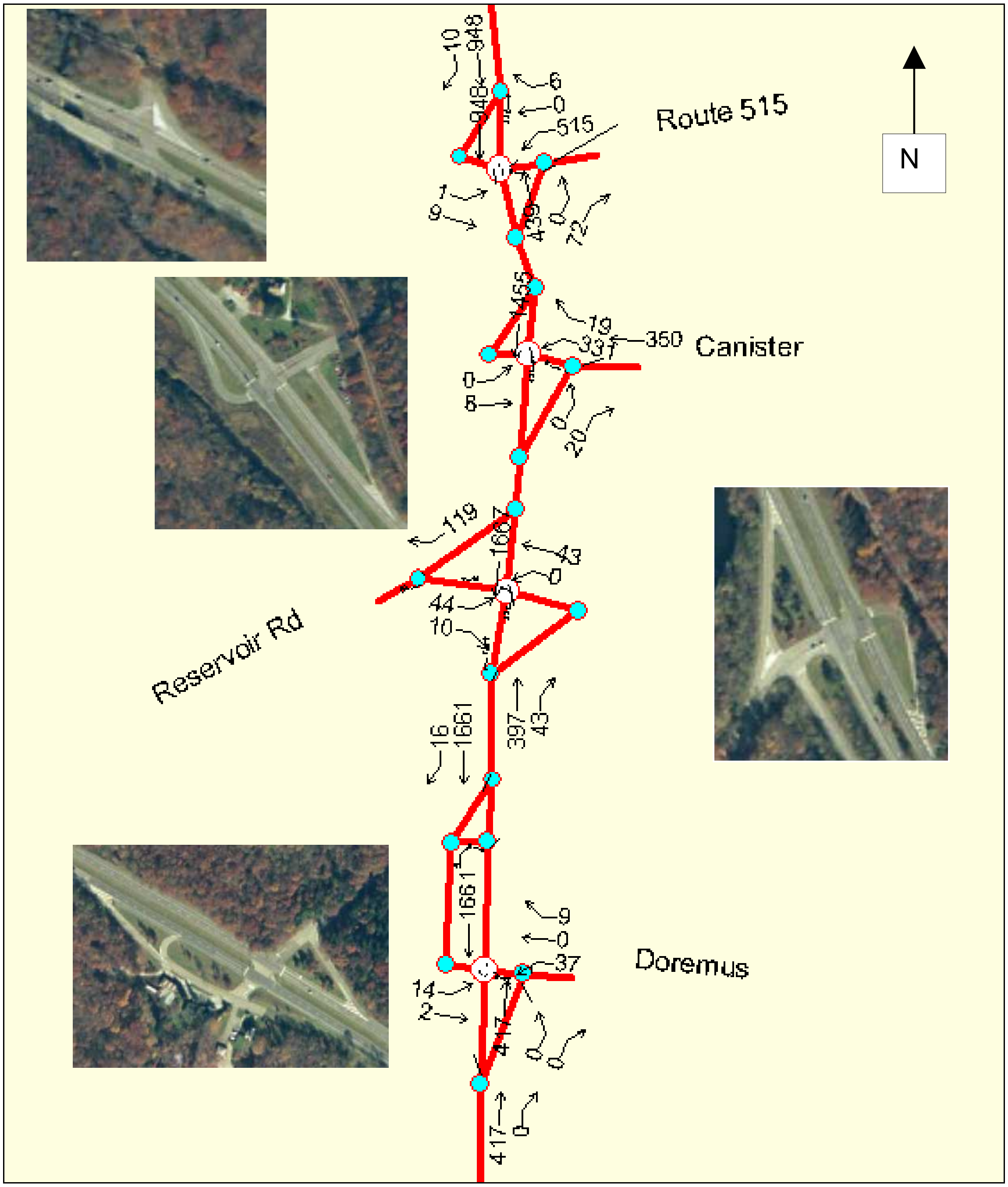


Figure 14 Route 23 study network (72)

A section of Route 23 from I-80, US 46 interchange to the Sussex Co line was studied to evaluate the adaptive control strategies project. Traffic is very directional and extremely heavy on this roadway segment. Platoons from Sussex Co vary sometimes very significantly a maximum to six minutes in duration traveling in one lane southbound in the morning. The heaviest traffic period is from September when school opens until after Thanksgiving. Traffic incidents on I-80/Route 15 generate additional amounts of traffic onto Route 23 from the Sparta area and the north. Recreational traffic from ski areas and lakes cause congestion. There was little or no overlap with traffic during the heavy traffic period of the fall. This roadway generates probably more phone calls about the signal timing and operation than any roadway in NJ during this period. Therefore, new timing plans must be generated every few years. 17 intersections on Route 23 are studied.

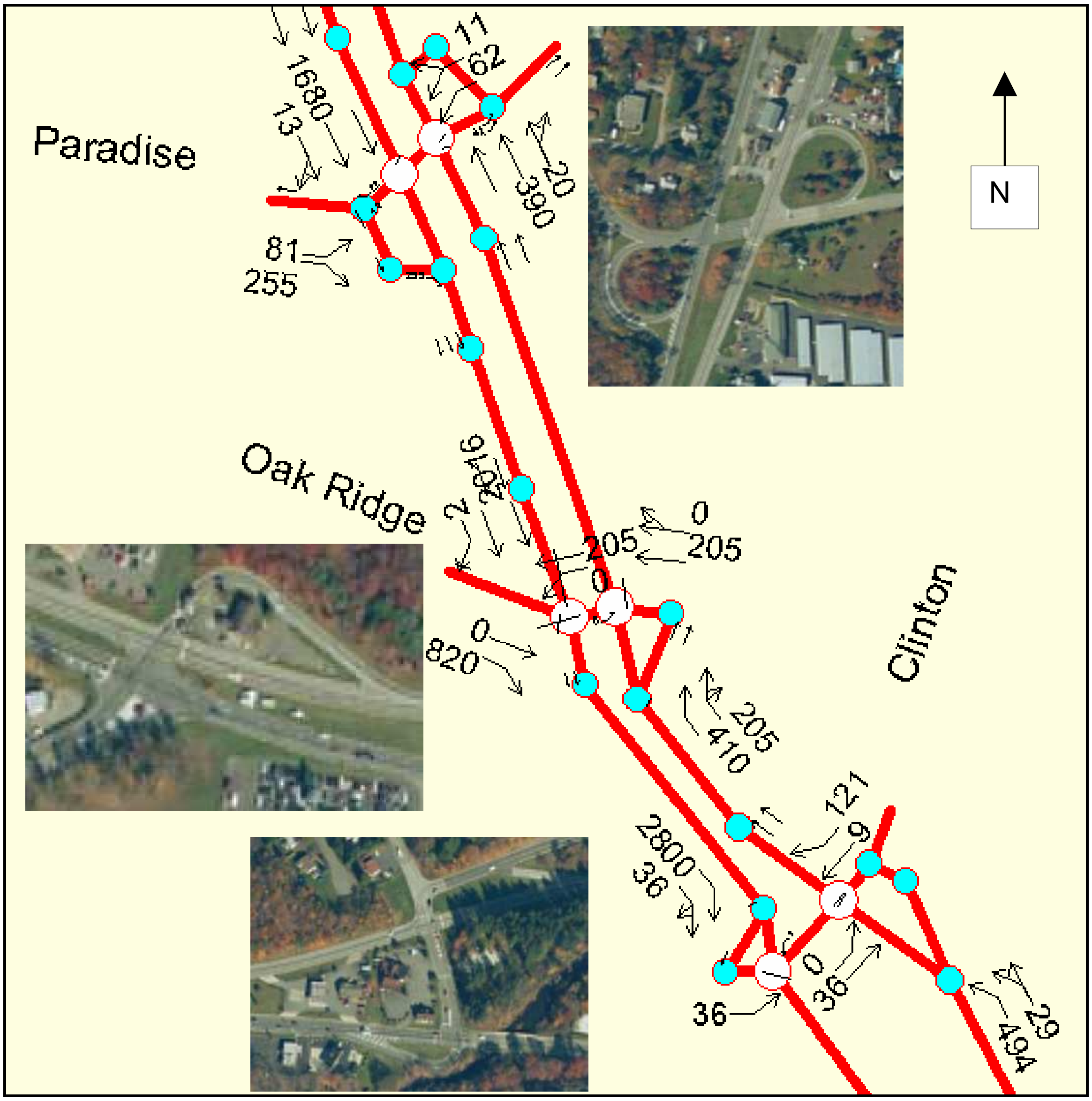
Table 12 Route 23 link lengths (feet)

Intersections	**Length**	**Intersections**	**Length**
CR 515	feet	Kanhouse Rd	2587
Canister Rd	5650	Echolake Rd	6758
Reservoir Rd	6864	Center Ct	21014
Doremus Rd	2112	Kiel Rd	2376
Paradise Rd	4700	Cascade Way	3749
Oak Ridge Rd	1795	Boonton Ave	1584
Clinton Rd	1373	Morse Ave	2059
La Rue Rd	2218	Cotluss Rd	1901



○ Signalized intersection
● Unsignalized intersection
Note: Not all unsignalized intersections are included in the network.

Figure 15 Route 23 Study road section

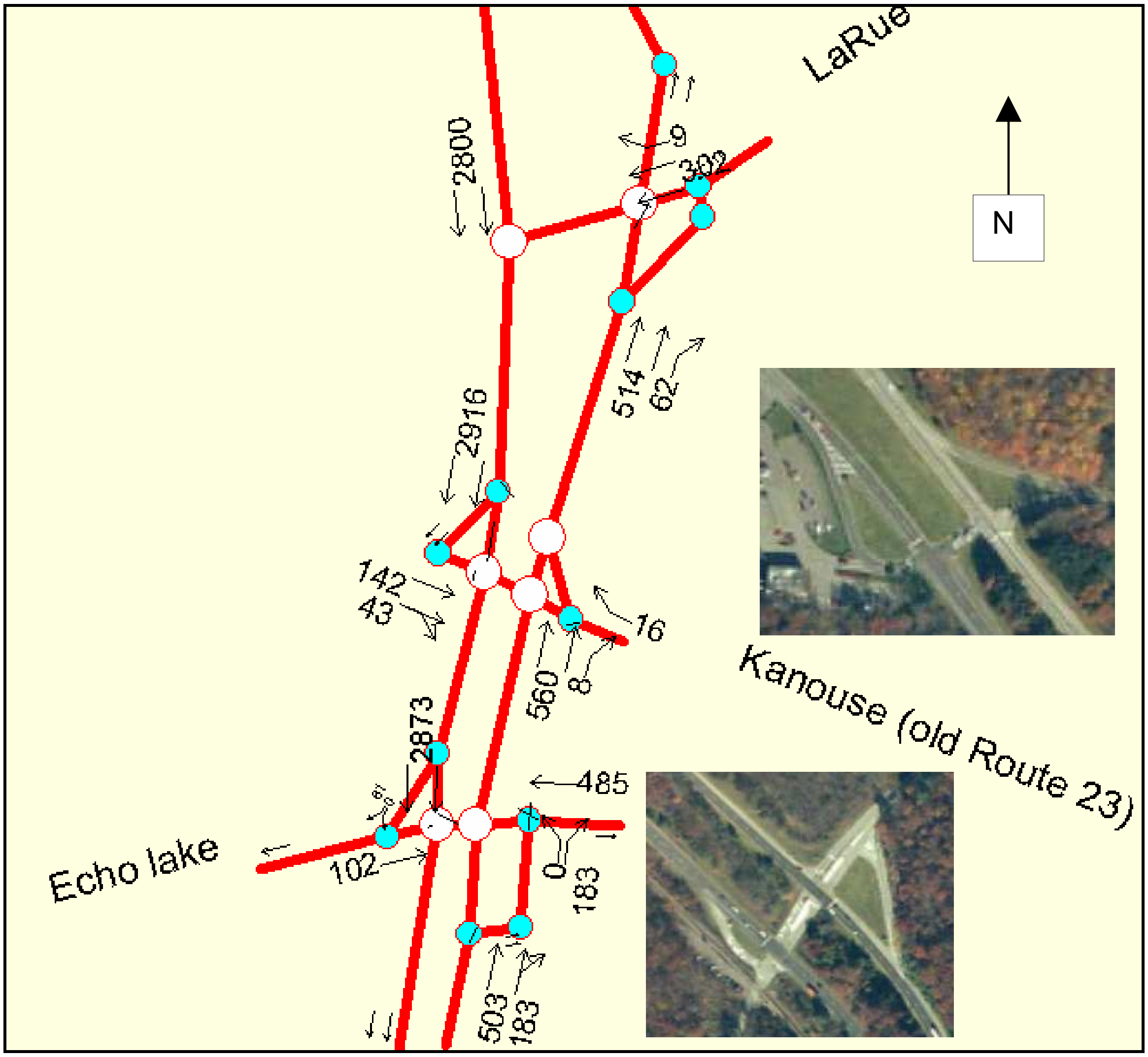


○ Signalized intersection

● Unsignalized intersection

Note: Not all unsignalized intersections are included in the network.

Figure 16 Route 23 study road section

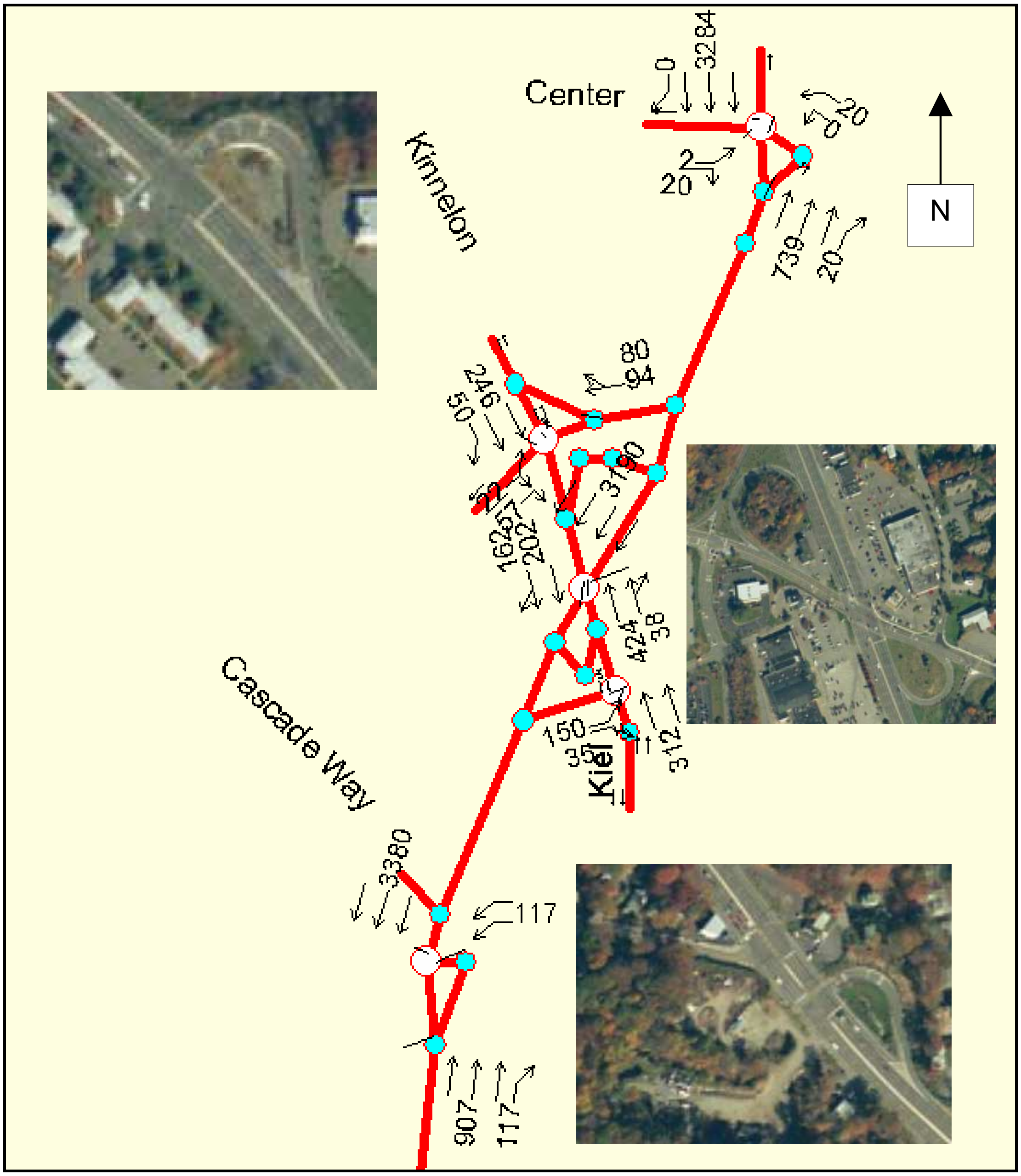


○ Signalized intersection

● Unsignalized intersection

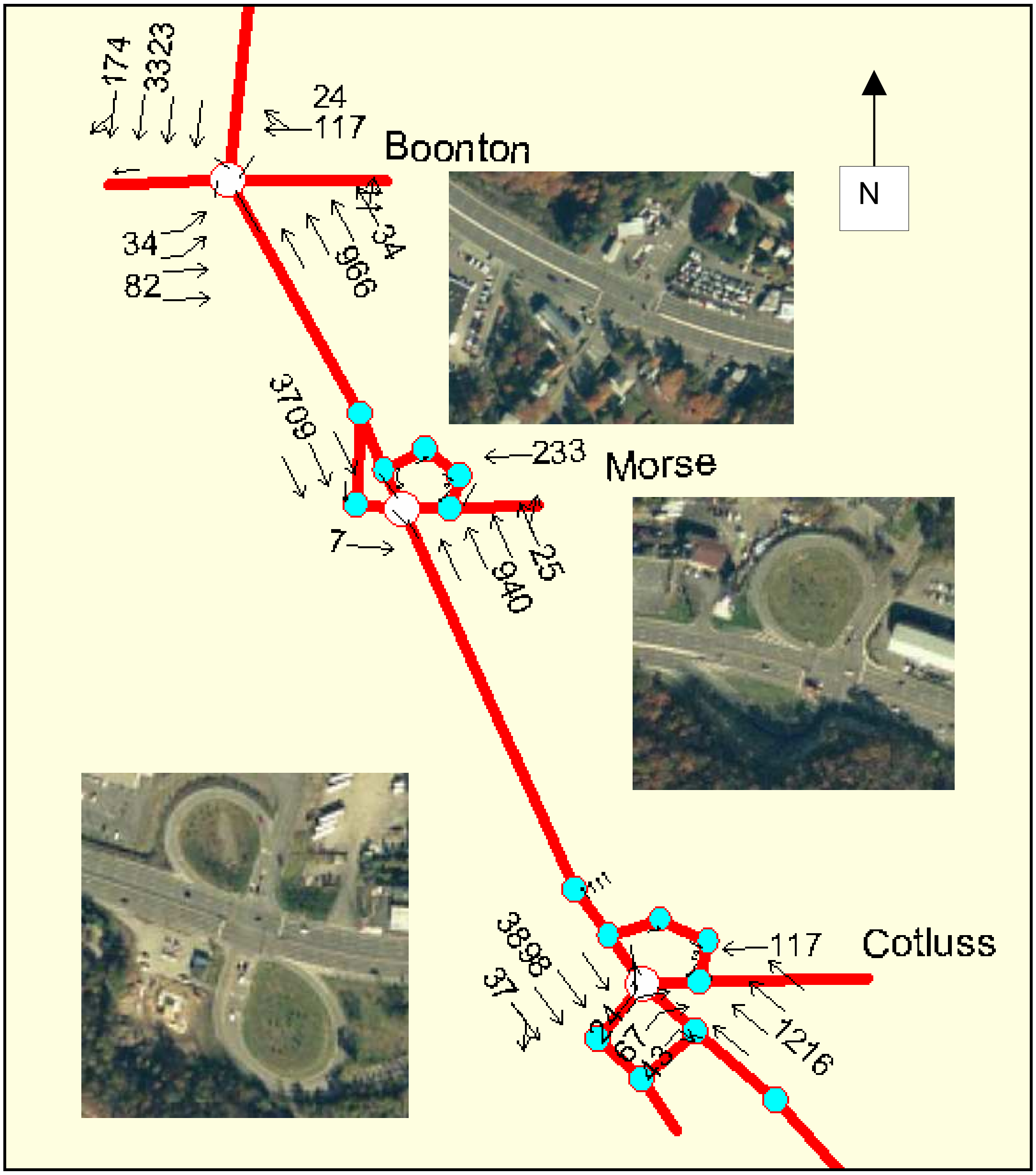
Note: Not all unsignalized intersections are included in the network.

Figure 17 Route 23 study road section



● Signalized intersection
● Unsignalized intersection
Note: Not all unsignalized intersections are included in the network.

Figure 18 Route 23 study road section



○ Signalized intersection

● Unsignalized intersection

Note: Not all unsignalized intersections are included in the network.

Figure 19 Route 23 study road section

Route 18 Network

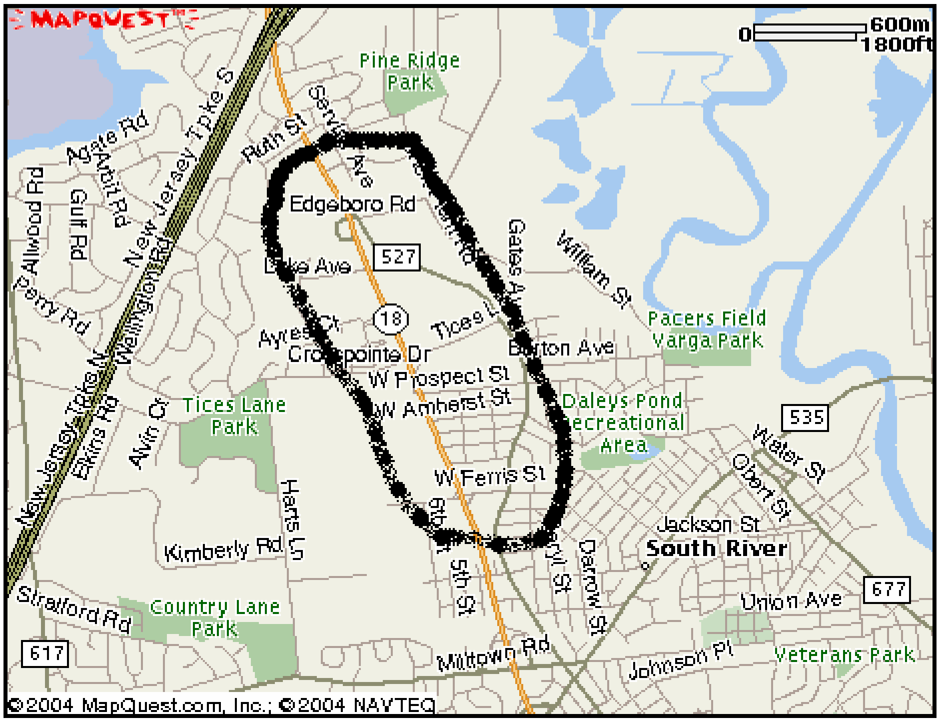


Figure 20 Route 18 study section (72)

Route 18 is an important north-south arterial in Middlesex county in NJ. The section of Route 18 considered in this study is from the Eggers/South Woodland Street intersection to the West Ferris Street intersection. A total of 5 intersections are in this arterial. The cross street demand on Old bridge Turnpike/Edgeboro Road, and, Tices Lane intersection is heavy. Spacings between each intersections are given below in Table 13.

Table 13 Intersection spacing on Route 18 intersections

Intersections	**Length (feet)**
Eggers/South Woodland Street	1637
Edgeboro/Old Bridge Turnpike	2006.4
Tices Lane	2693
West Prospect Street	581
West Ferris Street	2060

The north end of the arterial has an exit to the NJ Turnpike and US Route 1. Traffic is highly directional on this arterial during peak hours. During the morning peak, there is a high demand toward NJ Turnpike and US Route 1 (i.e. northbound), while during evening peak, there is a high demand from NJ Turnpike and US Route 1 (i.e. southbound). Flow is usually near saturation during peak hours. OPAC was installed on this route before, but because of high maintenance costs, control was switched back to actuated-control.

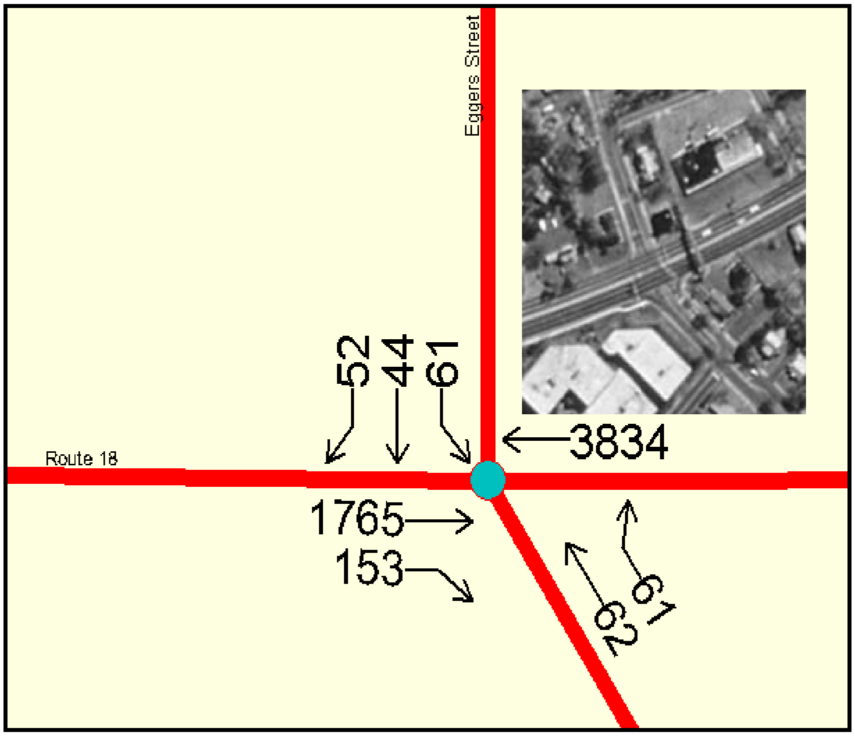


Figure 21 Route 18 study road section

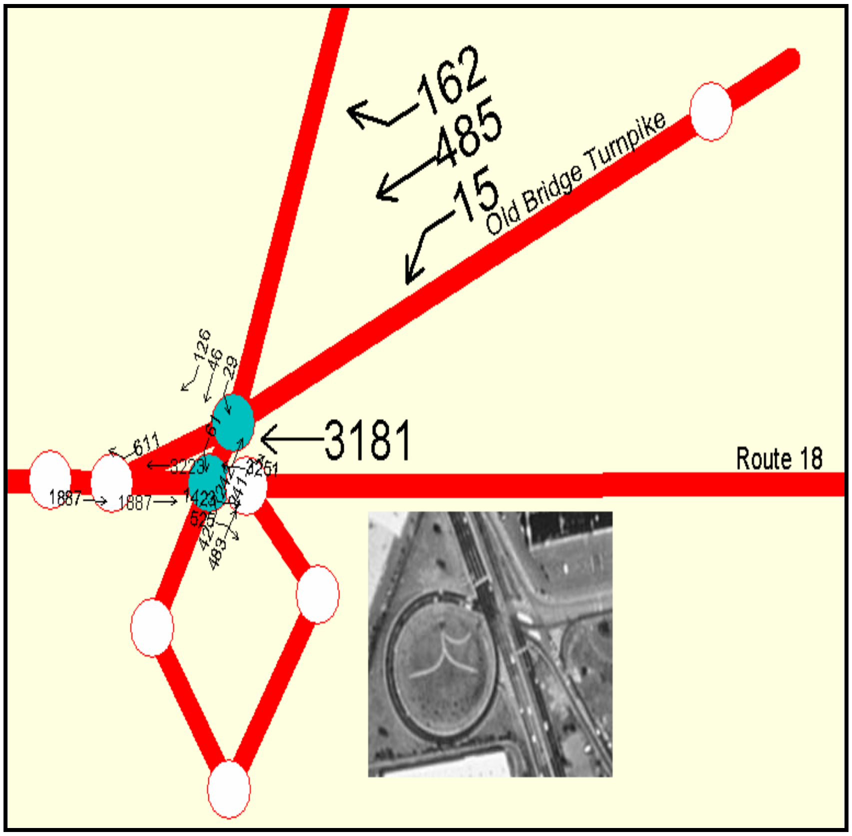


Figure 22 Route 18 study road section

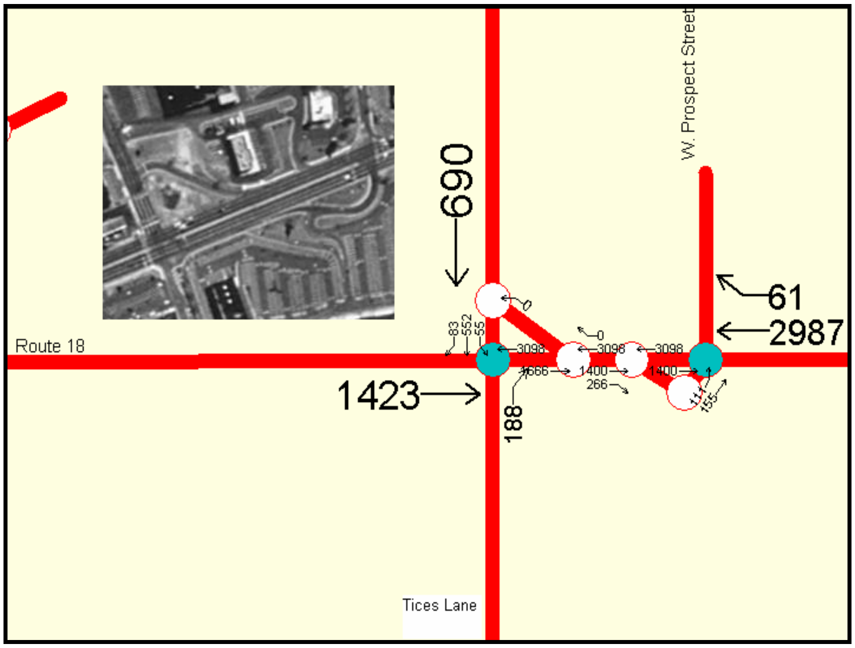


Figure 23 Route 18 study road section

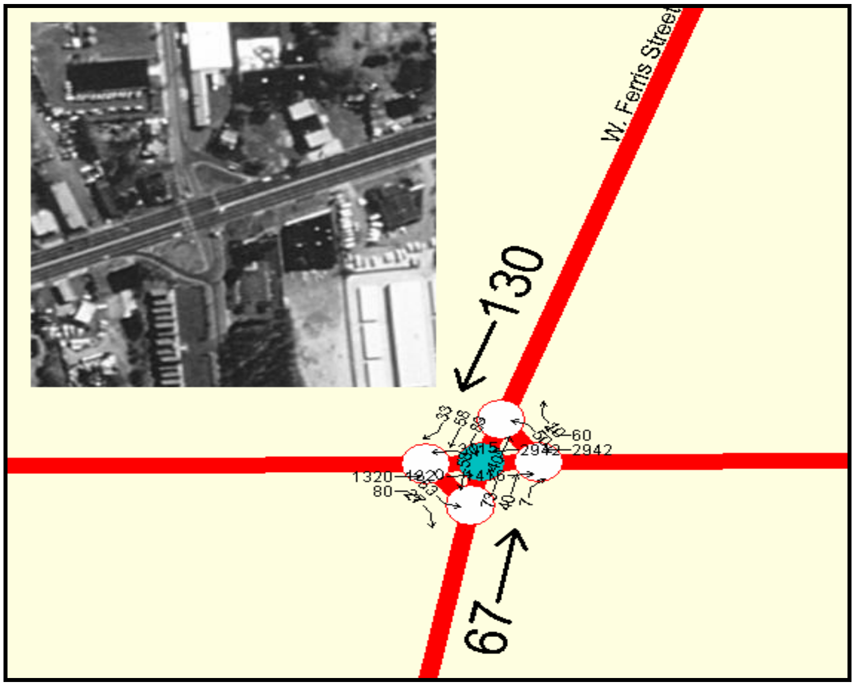


Figure 24 Route 18 study road section

Data Requirements for Modeling

The networks were created and analyzed using microsimulation softwares CORSIM, Paramics, and SimTraffic. The input data required for all three arterials are similar. Input data needed to create the network are as follows:

- Geometric components
    - Required geometry data were distance between intersections, number of lanes, lane alignment and lane channelization. Geometry data was obtained using a GIS map and straight-line diagram. Intersection details were collected using CADD drawings obtained from NJDOT. Aerial photographs shown in Figure 10 through Figure 24 were obtained from mapquest.com and terraserver.com websites. The preliminary site visit was also conducted to obtain basic information.
- Traffic demand
    - Traffic counts for each movement on every intersection were obtained from NJDOT. These traffic counts were collected on April 11, 2001 from 6 a.m. to 6 p.m. Peak-hour traffic counts, from 8 a.m. to 9 a.m. and from 7:30 a.m. to 8:30 a.m., are used for the Route 10 network and Route 23 network study respectively. For Route 18 morning peak traffic volumes were used from 8:00 a.m. to 9:00 a.m.
- Traffic Signal timings on each intersection
    - Traffic signal timings were given in DOT report in which the cycle length was fixed for each intersection whereas a range of green time for each phase was provided.

Traffic signal timings of Route 10

Fixed timings were used for the Route 10 study. Table 14 on the next page shows route 10 signal timing plans.

Table 14 Signal timing plan on NJ Route 10

Phase No.	Allowed movement		Considered	Synchro optimized	DOT
Route 10 & US 202					
1	EBT, WBT	Green time(s)	112	64	88-127
2	NBT, NBR, SBT, SBR	Green time(s)	28	26	10-31
		Cycle length	150	100	150
Route 10 & Ridgedale Ave Ext					
1	EBT, EBR, WBT	Green time(s)	99	54	97
2	NBT, NBL	Green time(s)	41	16	7-39
		Cycle length	150	80	150
Route 10 & Jefferson Ave					
1	EBT, EBR, WBT	Green time(s)	77	32	46-87
2	NBT, NBL, NBR	Green time(s)	20	47	7-28
3	SBT, SBR	Green time(s)	8	16	7-27
		Cycle length	120	100	120
Route 10 & Pine Place					
1	EBT, EBR, WBT	Green time(s)	69	35	64-70
2	SBL, SBR, SBT, NBL, NBR	Green time(s)	20	15	8-14
		Cycle length	90	60	90
Route 10 & Whippany Rd					
1	EBT, EBR, WBT	Green time(s)	57	34	50-69
2	NBT, NBR, SBT, SBL	Green time(s)	23	16	8-27
		Cycle length	90	60	90
Route 10 & Troy Hills Rd					
1	EBT, WBT	Green time(s)	56	93	40-51
2	SBL, SBR	Green time(s)	11	15	5-11

3	NBT, NBL, NBR	Green time(s)	8	17	7-12
		Cycle length	90	140	120
Route 10 & Algonquin Pkwy					
1	EBT, WBT, WBR	Green time(s)	61	25	50-63
2	SBL, SBR, SBT	Green time(s)	4	20	5
3	NBT, NBL, NBR, SBL, SBR, SBT	Green time(s)	15	15	7-20
		Cycle length	90	75	90
Route 10 & Ridgedale Ave					
Phase No.	Allowed Movement		Considered	Synchro optimized	DOT
1	EBT, EBR, WBT, WBR	Green time(s)	47	24	35-57
2	NBR, NBL, NBT	Green time(s)	13	20	7-18
3	SBL, SBR, SBT	Green time(s)	15	16	7-18
		Cycle length	90	75	90
Phase No.	Allowed movement		Considered	Synchro optimized	DOT
Route 10 & Jughandle					
1	EBT, EBR WBT, WBR	Green time(s)	52	45	39-60
2	SBL, SBR, SBT	Green time(s)	18	15	6-24
3	NBL, NBR	Green time(s)	5	15	5-8
		Cycle length	90	90	90
Route 10 & New Murray Rd					
1	EBT, EBR, WBT, WBR	Green time(s)	50	43	37-59
2	SBL, SBR, SBT	Green time(s)	12	16	7-18
3	NBT, NBL, NBR	Green time(s)	13	16	7-18
		Cycle length	90	90	90
Route 10 & Mt pleasant Ave					
1	EBT, EBR, EBL WBT, WBR, WBL	Green time(s)	45	19	36-57
2	SBL, SBR, SBT	Green time(s)	14	16	7-15

3	NBT, NBL, NBR	Green time(s)	16	15	7-16
		Cycle length	90	65	90
Route 10 & River Rd					
1	EBT, EBR, WBT, WBR	Green time(s)	45	18	34-57
2	SBL, SBR, SBT	Green time(s)	15	26	7-17
3	NBT, NBL, NBR	Green time(s)	19	16	7-20
		Cycle length	90	75	90
Route 10 & Okner Pkwy					
1	EBT, EBR, WBT	Green time(s)	60	24	47-70
2	SBL, SBR, SBT NBT, NBR	Green time(s)	20	16	7-30
		Cycle length	90	50	90
Route 10 & Walnut St					
1	EBT, WBT, WBR	Green time(s)	50	67	42-58
2	SBR, SBT	Green time(s)	11	16	6-14
3	NBT, NBL, NBR	Green time(s)	14	42	6-14
		Cycle length	90	140	90
Route 10 & Eisenhower Pkwy (EB)					
Phase No.	Allowed movement		Considered	Synchro optimized	DOT
1	EBT, EBR, EBL	Green time(s)	70	27	37-88
2	NBT, SBT	Green time(s)	40	23	12-37
		Cycle length	120	60	120
Route 10 & Eisenhower Pkwy (WB)					
1	WBT, WBR, WBL	Green time(s)	70	29	37-88
2	NBT, SBT	Green time(s)	40	21	12-37
		Cycle length	120	60	120
Route 10 & West Northfield					
1	EBT	Green time(s)	70	18	-
2	NBT	Green time(s)	55	21	-
		Cycle length	135	45	-

Traffic signal timings of Route 23

Traffic signal timings were obtained from NJDOT in which the cycle length was fixed for each intersection whereas a range of green time for each phase is provided. Table 15 shows the traffic signal timing used for the study.

Table 15 Route 23 signal timings

Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
Route 23 & route 515						
1	NBT, SBT	41	66	6	2	160
2	EBT, EBL	5	15	3	3	
3	WBL, WBR	7	15	3	3	
Route 23 & Canister Rd						
Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
1	NBT, SBT	119	119	6	2	160
2	EBT, EBL WBL, WBR	7	28	3	2	
Route 23 & Reservoir Rd						
Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
1	NBT, SBT	119	127	6	2	160
2	EBR, EBL	7	20	3	2	
3	WBL, WBT	4	20	3	2	
Route 23 & Doremus Rd						
Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
1	NBT, SBT	119	127	6	2	160
2	EBT, EBL WBL, WBR	7	20	3	2	

Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
Route 23 & Paradise Rd (SB)						
1	SBT	123	129	6	2	160
2	EBT, EBR WBL, WBT	10	18	3	2	
Route 23 & Paradise Rd (NB)						

1	NBT, NBR	123	129	6	2	160
2	EBT, EBL WBT, WBR	10	18	3	2	
Route 23 & Oak Ridge Rd (SB)						
1	SBT, SBR	100	120	6	2	160
2	EBT, EBR WBT, WBL	7	26	3	3	
Route 23 & Oak Ridge Rd (NB)						
1	NBT	100	120	6	2	160
2	EBL, WBT	7	26	3	3	
Route 23 & Clinton Rd (SB)						
1	SBT	123	140	6	2	160
2	EBL, WBL	7	7	3	2	
Route 23 & Clinton Rd (NB)						
1	NBT	123	140	6	2	160
2	EBT, EBL, WBT, WBR	7	7	3	2	
Route 23 & La Rue Rd (NB)						
1	NBT	125	133	6	2	160
2	WBT, WBR	7	14	3	2	
Route 23 & La Rue Rd (SB)						
1	SBT	125	133	6	2	160
2	WBL	7	14	3	2	
Route 23 & Kanhouse Rd						
1	SBT	127	139	6	2	160
2	EBT	7	8	3	2	
Route 23 & Kanhouse Rd (NB)						
1	NBT	127	139	6	2	160
2	EBT, EBL	7	8	3	2	
Route 23 & Echolake Rd (SB)						
1	SBT	120	123	6	2	160
2	EBT, WBT, WBL	10	24	3	2	
Route 23 & Echolake Rd (NB)						
1	NBT	120	123	6	2	160
2	EBT, EBL, WBT	10	24	3	2	
Route 23 & Center Ct						
1	NBT, SBT, SBR	120	131	5	2	160
2	EBR, EBL, WBT, WBL	7	16	4	2	
Route 23 & Kinnelon Rd						
1	NBT, SBT	100	108	5	2	160

Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
2	EBT, WBT	7	37	5	3	160
Route 23 & Cascade Way (EB)						
1	NBT, SBT, SBR	120	133	5	2	160
2	EBR	7	15	3	2	
Route 23 & Cascade Way (WB)						
1	NBT, SBT	120	133	5	2	160
2	WBL	7	15	3	2	
Route 23 & Boonton Ave						
1	NBT, SBT	120	120	5	2	160
2	EBT, EBL, WBT, WBL	10	10	4	2	
Route 23 & Morse Ave						
1	NBT, SBT	120	122	5	1	160
2	EBT, EBR,EBL, WBT, WBR, WBL	7	26	3	3	
Route 23 & Cotluss Rd						
1	NBT, NBR, NBL SBT, SBR, SBL	120	120	5	1	160
2	EBT, EBR,EBL, WBT, WBR, WBL	5	9	3	3	

Timing Plan for NJ Route 18

The timing plan for NJ Route 18 shown in the Table 16 below was generated using an optimization module in Synchro.

Table 16 Route 18 signal timings

Phase no.	Allowed movement	Min green	Max green	Amber	All red	Cycle length
Route 18 and Eggers/S.Woodland Street						
1	NBT, SBT	78	121	5	2	150
2	EBR, EBT WBT, WBL	7	21	3	2	
Route 18 and Edgeboro/Old Bridge Turnpike						
1	NBT, SBT	78	106	5	2	150
2	WBL	7	7	2	0	
3	EBL, EBT, WBL	15	21	3	2	
Route 18 and Tices Lane						
1	NBT, SBT	75	106	5	2	150
2	EBR, WBL, WBT	12	36	3	2	

Phase no	Allowed movement	Min green	Max green	Amber	All red	Cycle length
Route 18 and West Prospect Street						
1	NBT, SBT	75	118	5	2	150
2	EBL, EBT	12	24	3	2	
Route 18 and West Ferris Street						
1	NBT, SBT	75	111	5	2	150
2	EBL, EBT, WBL, WBT	12	30	3	2	

CORSIM, Paramics, and Synchro models use the concept of links and nodes to define the roadway network. The user characterizes position and attributes of nodes (e.g., intersections), and the nodes are connected with links. Then lane configurations, speeds and signal control attributes are assigned to links and nodes. Traffic counts for each movement on the intersection are required as traffic flow input in CORSIM and Synchro. Travel demand in Paramics is defined by a matrix of O-D trips. Because only traffic counts for movements on the intersection are available, it is necessary to create a Paramics friendly demand matrix using traffic counts.

O-D Matrix Generation

Traffic counts for each intersection were obtained from NJDOT. In Paramics, travel demand is defined from zone to zone. Zones are defined as geographical areas where trips start and finish. Figure 25 shows an example network with zones. Zones are assigned at each end as shown in Figure 25. Traffic counts for each intersection are available. However, the flow from zone to zone cannot be determined using these counts. For example, the traffic from zone 2 to zone 7 cannot be determined using intersection counts. Therefore, traffic counts at each intersection need to be converted into a zone-to-zone demand matrix. Because the Route 10 and Route 23 networks have 57 and 41 zones, respectively, it is not possible to use that matrix to explain the procedure. The example network with 8 zones is used to clarify the procedure of creating an O-D matrix with intersection counts.

Step 1:

To change traffic counts on each intersection into a demand matrix an excel spreadsheet is used. Traffic flow from and to each zone can be obtained from traffic counts. Assume for an intersection *"i"* there are three movements:

X_i^L = Total vehicles turning left from the intersection *i*

X_i^T = Total vehicles going through from the intersection *i*

X_i^R = Total vehicles turning right from the intersection *i*

From the first intersection in the network as shown in Figure 25, the total flow from zone j = 1, can be given as

$$f_j = X_i^L + X_i^T + X_i^R \tag{2}$$

Likewise, the total number of trips entering zone can also be obtained by counting the trips to that zone from each approach on the intersection. Using these data, total trips from and to each zone are determined. When the total number of vehicles leaving and entering the zones is determined, we create an initial matrix as shown in Figure 26.

Step 2:

Provide the values in the initial table that can be obtained from the network. For example, trips from zone 1 to 2, 1 to 3, 8 to 7, and 8 to 6 can be determined from the traffic counts on intersection. Furthermore, the traffic movements from zone 1 to 4, 1 to 2, 1 to 6, etc. are not possible; hence, trips for these O-D matrices will be zero. After fixing these values, make a second-stage demand matrix as shown in Figure 27.

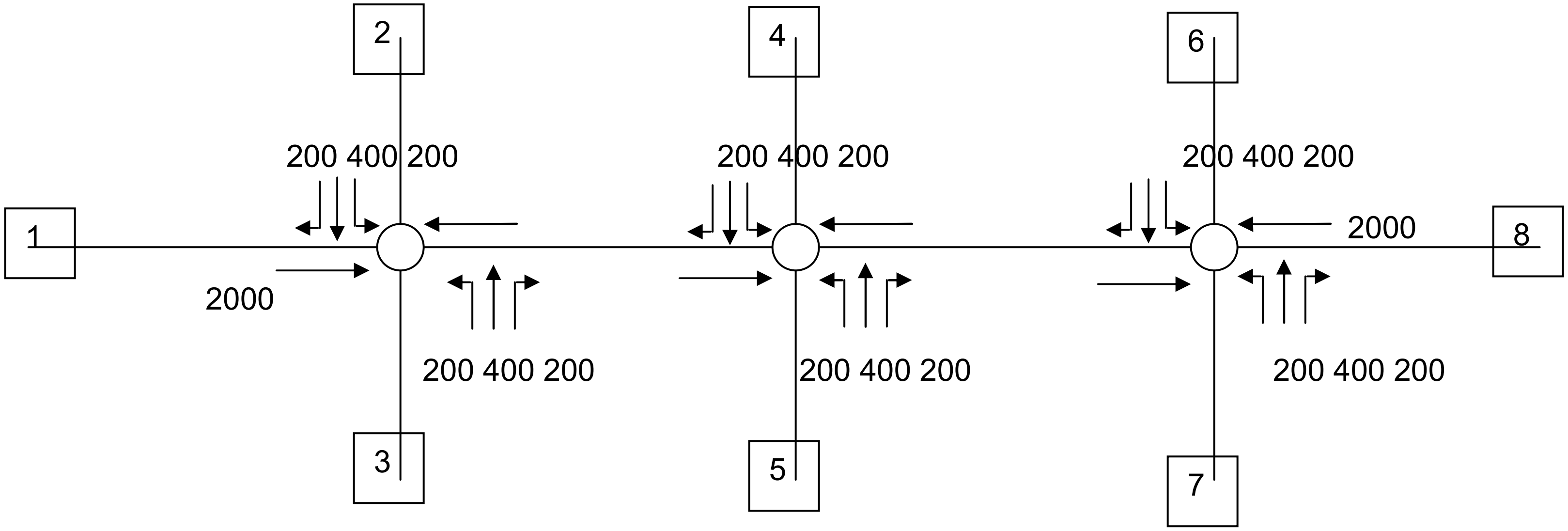


Figure 25 Example network to generate demand matrix

To From	1	2	3	4	5	6	7	8	Total
1									**2000**
2									**800**
3									**800**
4									**800**
5									**800**
6									**800**
7									**800**
8									**2000**
Total	**2600**	**600**	**600**	**600**	**600**	**600**	**600**	**2600**	

Figure 26 Demand matrix (Step 1)

To From	1	2	3	4	5	6	7	8	Total
1	0	0	200	0		0			**2000**
2	200	0	400	0		0			**800**
3	200	400	0	0		0			**800**
4			0	0	400	0	0		**800**
5			0	400	0	0			**800**
6			0		0	0	400	200	**800**
7			0		0	400	0	200	**800**
8			0		0	200	0	0	**2000**
Total	**2600**	**600**	**600**	**600**	**600**	**600**	**600**	**2600**	

Figure 27 Demand matrix (Step 2)

Step 3:

Generate the trips for the remaining of the O-D zones. Many issues must to be considered here. One must generate trips so that the sum of all trips from and to each zone equals the total number of trips given in Figure 26.

Traffic counts for each turning movement on the intersection are also given. Therefore, trips should be generated in such a way that on the intersection the traffic count for turning movements equals the given data. For example, as shown in the network, 200 vehicles from zone 5 make a left turn at the intersection. The possible movements from zone 5 to other zones to make a left turn are 5 to 1and 5 to 2. Hence assign the trips to 5 to 2 and 5 to 1 so that their sum of them is 200. At the same time, vehicles entering zone 1 or 2 should not exceed its limit. (i.e. it is 2600 for zone 1, and it is 600 for zone 2).

Repeat the last step for each zone till the matrix is balanced (Figure 28).

To From	1	2	3	4	5	6	7	8	Total
1	0	0	200	0	100	0	25	1675	**2000**
2	200	0	400	0	50	0	50	100	**800**
3	200	400	0	0	50	0	50	100	**800**
4	100	100	0	0	400	0	0	200	**800**
5	175	25	0	400	0	0	75	125	**800**
6	75	25	0	100	0	0	400	200	**800**
7	125	25	0	50	0	400	0	200	**800**
8	1775	25	0	50	0	200	0	0	**2000**
Total	**2600**	**600**	**600**	**600**	**600**	**600**	**600**	**2600**	

Figure 28 Demand matrix (Step 3)

CALIBRATION OF ARTERIAL NETWORKS IN CORSIM, SYNCHRO, AND PARAMICS

The reliability of any simulation software model depends on its ability to produce results close to actual data. When the network is created in all three models, base runs were taken to observe the model accuracy without any adjustments, and the quality of data inputs are provided. Except of NJ Route 18, which has fewer intersections, initial runs provided incompatible results for flow and intersection delay. Hence calibration was required for Route 10 and Route 23 to obtain similar results in all simulation packages.

Calibration of Network for Flow

As soon as the network is created in all three models, base runs were taken to observe the model accuracy without any adjustments and the quality of data inputs provided. Initial runs provided incompatible results for flow. Calibration was needed to get the similar flow in the simulation output compared with the data input. A widely used error measure that can provide a fairly good initial estimate of the degree of fit between the simulated and the actual traffic measurements is the Root Mean Squared Percent Error (RMSE). Equation shown below defines Root Mean Squared Percent Error (RMSE). This error gives an estimate of the total percentage error and is define as

$$\mathrm{RMSP} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - y_i}{y_i}\right)^2}\ , \tag{3}$$

where;

RMSE: Root mean squared percentage error

x_i: Simulated traffic measurement value of run *i*

y_i: Actual traffic measurement value of run *i*

To determine the number of replications required, sequential method described in Appendix 1 was used. The results discussed in these sections are mean results of several replications.

NJ Route 18

NJ Route 18 network was coded using Synchro, Paramics, and CorSim packages. An appropriate demand matrix was also created to be used in Paramics, whereas Synchro and CorSim use intersection counts. Because this network has only five intersections, calibration was easier. Initial results showed that vehicle counts for different movements across the intersections obtained using Paramics OD matrix did not match actual traffic counts, as well as simulated traffic counts obtained using Synchro and CorSim. However, after few revisions in the demand matrix, similar traffic flow was obtained for all three simulation packages. These results are summarized in Table 17.

Table 17 Flow comparison for calibrated NJ Route 18 networks

South Bound							
Intersection	Actual	SimTraffic	% Error	CorSim	% Error	Paramics	% Error
Eggers/South Woodland Street	1918	1941	-1.20	1912	0.31	1967	-2.55
Edgeboro/Old bridge Turnpike (1)	1887	1894	-0.37	1940	-2.81	1923	-1.90
Tices Lane	1423	1405	1.26	1512	-6.25	1523	-7.02
West Prospect Street	1400	1396	0.29	1520	-8.57	1412	-0.85
West Ferris Street	1320	1319	0.08	1372	-3.94	1357	-2.80
North Bound							
Intersection	Actual	SimTraffic	% Error	CorSim	% Error	Paramics	% Error
Eggers/South Woodland Street	3834	3732	2.66	3940	-2.76	3902	-1.77
Edgeboro/Oldbridge Turnpike (1)	3181	3139	1.32	3388	-6.51	3254	-2.29
Edgeboro/Oldbridge Turnpike (2)	662	659	0.45	668	-0.91	663	-0.15
Tices Lane	3098	3065	1.07	3104	-0.19	3198	-3.22
West Prospect Street	3048	3033	0.49	3092	-1.44	3032	0.52
West Ferris Street	3002	2901	3.36	2900	3.40	3002	0
East Bound							
Intersection	Actual	SimTraffic	% Error	CorSim	% Error	Paramics	% Error
Eggers/South Woodland Street	123	120	2.44	112	8.94	118	4.06
Edgeboro/Oldbridge Turnpike (1)	525	500	4.76	499	4.95	493	6.09
Edgeboro/Oldbridge Turnpike (2)	483	459	4.97	453	6.21	495	-2.48
Tices Lane	188	198	-5.32	188	0.00	189	-0.53
West Prospect Street	266	258	3.01	284	-6.77	288	-8.27
West Ferris Street	120	112	6.67	135	-12.50	117	2.5
West Bound							
Intersection	Actual	SimTraffic	% Error	CorSim	% Error	Paramics	% Error
Eggers/South Woodland Street	157	156	0.64	148	5.73	148	5.73
Edgeboro/Oldbridge Turnpike (1)	61	59	3.28	76	-24.59	69	-13.11
Edgeboro/Oldbridge Turnpike (2)	203	213	-4.93	204	-0.49	199	1.97
Tices Lane	690	670	2.90	696	-0.87	721	-4.49
West Prospect Street	-	-	-	-	-	-	-
West Ferris Street	180	172	4.44	172	4.44	174	3.33

NJ Route 23

CORSIM:

CORSIM analysis for Route 23 was carried out using a network created in Synchro. Synchro creates a required input file (.trf file) for CORSIM. Though Synchro tries to produce error-free .trf files, CORSIM is very likely to report errors in the input file. Therefore, the initial task was to correct these errors. Most of these errors are related to the lane configuration. Synchro produced a .trf file in which the lane number and lane chanelization were not correct. When these errors were corrected, the network was checked and verified by converting it into a TRAFED file. TRAFED is a GUI that allows user to create a network with links and nodes. An initial run was taken to observe the output flow obtained using CorSim with observed flow without changing any default parameters. It provided inconsistent results. The difference was higher for southbound and northbound flows after the intersection at Echolake Road. Initial calibration was carried out by checking the lane chanelization. A permitted left turn for westbound flow on Echolake Road made it difficult for these vehicles to discharge quickly because of a shorter green time. To obtain the correct flow, an acceptable gap in oncoming traffic for left turn was kept 1 second. The mean start-up delay and mean discharge headway were also decreased and kept as 1.0 second and 1.4 seconds respectively. When these changes were made, it was found that not all vehicles discharged from Echolake Road could discharge from Center Court and that caused less flow on the remaining intersections. After discussion with TSIS (CORSIM software developers), it seemed that the problem stemmed from the test for equilibrium. At the end of every interval during the initialization period, CORSIM compares the number of vehicles entering the network with the number of vehicles exiting the network. When the difference is small enough for two consecutive intervals, it assumes that equilibrium has been achieved. Sometimes this occurs too early. In version 5.0, it is not possible to force the initialization period to last for as long as specified so simulation runs were taken for 2 hours, and second-hour results were considered. When these changes were made CORSIM gave comparable results and the root mean square percentage error obtained was between 2.9 to 5.3.

Paramics:

When the network was created in Paramics, the initial run was taken without changing the default parameters. The initial run gave inconsistent results. Initial calibration was carried out by checking the network geometry, lane chanelization, and turning movements. Some changes were required in lane chanelization and turning movements. The difference was high for southbound flow. To get the required flow, the mean headway and mean reaction time was set to 0.6 second. In Paramics, the network is not loaded with vehicles when the simulation begins, instead they are generated after the simulation start. To avoid discrepancy in flow values due to this, the simulation run was taken for two hours. It was observed that vehicles are evenly spread across the network after one hour and hence results for the second hours were considered for the analysis. When all these changes were made, results provided the correct flow, Paramics gave comparable results and Root mean square percentage error obtained was between 3 and 6.

***Synchro/SimTraffic*:**

Initial simulation results of SimTraffic were inconsistent with observed flow. For the higher volume-to-capacity ratio SimTraffic created congestion. To get the correct flow, headway and gap acceptance factors were decreased, and the turning speed for the left turn on Echolake Road was increased. After these changes were made, SimTraffic gave comparable results, and the root mean square percentage error obtained was between 2.5 to 5.6.

Table 18 through Table 21 show the flow obtained from CORSIM, SimTraffic and Paramics simulation runs for Route 23.

Table 18 Route 23 flows on each link (Southbound)

Direction	**Actual**	**CORSIM**	**% Error**	**SimTraffic**	**% Error**	**Paramics**	**% Error**
Route 515	**948**	949	**0.1**	914	**-3.6**	1009	**6.4**
Cannister	**1455**	1436	**-1.3**	1410	**-3.1**	1527	**4.9**
Reservoir	**1667**	1616	**-3.1**	1615	**-3.1**	1746	**4.7**
Doremus Rd	**1661**	1621	**-2.4**	1628	**-2.0**	1776	**6.9**
Paradise Rd	**1693**	1647	**-2.7**	1662	**-1.8**	1751	**3.4**
Oak Ridge	**2018**	1931	**-4.3**	1954	**-3.2**	2103	**4.2**
Clinton Rd	**2800**	2707	**-3.3**	2705	**-3.4**	2856	**2.0**
La Rue Rd	**2800**	2718	**-2.9**	2745	**-2.0**	2863	**2.3**
Kanhouse	**2916**	2831	**-2.9**	2835	**-2.8**	2991	**2.6**
Echo Lake	**2813**	2756	**-2.0**	2802	**-0.4**	2891	**2.8**
Center Ct	**3284**	3369	**2.6**	3089	**-5.9**	3326	**1.3**
Kiel Rd	**3190**	3272	**2.6**	2985	**-6.4**	3204	**0.4**
Cascade	**3380**	3366	**-0.4**	3159	**-6.5**	3450	**2.1**
Boonton Ave	**3497**	3425	**-2.1**	3300	**-5.6**	3563	**1.9**
Morse	**3709**	3677	**-0.9**	3513	**-5.3**	3851	**3.8**
Cotluss	**3935**	3854	**-2.1**	3732	**-5.2**	4027	**2.3**
RMSP		**2.4**		**4.2**		**3.4**	

Table 19 Route 23 flow on each link (Northbound)

Direction	**Actual**	**CORSIM**	**% Error**	**SimTraffic**	**% Error**	**Paramics**	**% Error**
Route 515	**439**	441	**0.5**	430	**-2.1**	470	**7.1**
Cannister	**421**	403	**-4.3**	403	**-4.3**	440	**4.5**
Reservoir	**397**	376	**-5.3**	387	**-2.5**	411	**3.5**
Doremus Rd	**417**	390	**-3.8**	418	**0.2**	442	**6.0**
Paradise Rd	**410**	374	**-6.3**	410	**0.0**	419	**2.2**
Oak Ridge	**410**	373	**-4.1**	414	**1.0**	402	**-2.0**
Clinton Rd	**494**	453	**-4.3**	471	**-4.7**	494	**0.0**
La Rue Rd	**514**	469	**-4.9**	485	**-5.6**	509	**-1.0**
Kanhouse	**560**	545	**-2.7**	553	**-1.3**	557	**-0.5**
Echo Lake	**503**	490	**-2.6**	510	**1.4**	503	**0.0**
Center Ct	**739**	713	**-3.5**	753	**1.9**	724	**-2.0**
Kiel Rd	**759**	754	**-0.7**	773	**1.8**	728	**-4.1**
Cascade	**907**	929	**2.4**	898	**-1.0**	875	**-3.5**
Boonton Ave	**1000**	987	**-1.3**	1008	**0.8**	980	**-2.0**
Morse	**965**	1039	**7.7**	992	**2.8**	959	**-0.6**
Cotluss	**1216**	1215	**-0.1**	1232	**1.3**	1214	**-0.2**
RMSP		**3.9**		**2.5**		**3.0**	

Table 20 Cross street flow (Westbound)

Cross street	**Actual**	**CORSIM**	**% Error**	**SimTraffic**	**% Error**	**Paramics**	**% Error**
Route 515	**521**	504	**-3.26**	531	**1.92**	545	**4.61**
Cannister Rd	**350**	337	**-3.71**	331	**-5.43**	366	**4.57**
Reservoir Rd	**43**	46	**6.98**	41	**-4.65**	45	**4.65**
Doremus Rd	**46**	48	**4.35**	44	**-4.35**	49	**6.52**
Paradise Rd	**73**	78	**6.85**	70	**-4.11**	75	**2.74**
	62	65	**4.84**	60	**-3.23**	64	**3.23**
Oak Ridge Rd	**205**	201	**-1.95**	211	**2.93**	216	**5.37**
	205	201	**-1.95**	211	**2.93**	216	**5.37**
Clinton Rd	**130**	136	**4.62**	124	**-4.62**	133	**2.31**
	9	10	**11.11**	8	**-11.11**	10	**11.11**
La Rue Rd	**311**	317	**1.93**	302	**-2.89**	327	**5.14**
	302	306	**1.32**	295	**-2.32**	311	**2.98**
Kanhouse Rd	**-**						
Echo Lake Rd	**485**	470	**-3.09**	472	**-2.68**	498	**2.68**
	441	430	**-2.49**	431	**-2.27**	455	**3.17**
Center Ct	**20**	22	**10.00**	17	**-15.00**	23	**15.00**
Kiel Rd	**462**	450	**-2.60**	442	**-4.33**	475	**2.81**
Cascade Way	**117**	121	**3.42**	108	**-7.69**	122	**4.27**
Boonton Ave	**422**	412	**-2.37**	402	**-4.74**	435	**3.08**
Morse	**233**	225	**-3.43**	236	**1.29**	245	**5.15**
Cotluss	**127**	124	**-2.36**	129	**1.57**	122	**-3.94**
RMSP		**4.9**		**5.6**		**5.8**	

Table 21 Cross street flow (Eastbound)

Cross street	Actual	CORSIM	%Error	SimTraffic	%Error	Paramics	%Error
Route 515	**10**	11	**10.00**	9	**-10.00**	11	**10.00**
Cannister	**8**	9	**12.50**	7	**-12.50**	9	**12.50**
Reservoir	**54**	52	**-3.70**	56	**3.70**	58	**7.41**
Doremus Rd	**16**	15	**-6.25**	15	**-6.25**	17	**6.25**
Paradise Rd	**93**	90	**-3.23**	95	**2.15**	99	**6.45**
	53	51	**-3.77**	55	**3.77**	56	**5.66**
Oak Ridge Rd	820	807	**-1.59**	839	**2.32**	791	**-3.54**
	0						
Clinton Rd	**36**	34	**-5.56**	37	**2.78**	34	**-5.56**
	36	34	**-5.56**	37	**2.78**	34	**-5.56**
La Rue Rd	**-**						
	-						
Kanhouse Rd	**185**	181	**-2.16**	192	**3.78**	196	**5.95**
	142	138	**-2.82**	147	**3.52**	148	**4.23**
Echo Lake Rd	**108**	104	**-3.70**	101	**-6.48**	112	**3.70**
	106	102	**-3.77**	100	**-5.66**	109	**2.83**
Center Ct	**22**	21	**-4.55**	23	**4.55**	21	**-4.55**
Kiel Rd.	**364**	372	**2.20**	355	**-2.47**	352	**-3.30**
Cascade	**43**	41	**-4.65**	46	**6.98**	41	**-4.65**
Boonton	**262**	255	**-2.67**	268	**2.29**	259	**-1.15**
Morse	**44**	46	**4.55**	45	**2.27**	41	**-6.82**
Cotluss	**134**	139	**3.73**	137	**2.24**	131	**-2.24**
RMSP		**5.3**		**5.3**		**6.0**	

NJ Route 10

As soon as the network was created in all the three simulation software packages, initial runs were taken to observe the flow without changing default parameters. Initial results were inconsistent with the observed flows. Initial calibration was carried out checking lane chanelization, network geometry, and turn movements. The initialization period was also kept higher for all three models to reach equilibrium. After making the initial changes, it was found that a comparable flow could not be obtained for some of the links in CORSIM and SimTraffic. However, Paramics was not giving the comparable flow for the whole network. Through visualization, it was found that because of less green time for cross streets on Jefferson Avenue, US 202, and River Rd, vehicles could not discharge quickly. In CORSIM, the acceptable gap for oncoming traffic for left and right turns was decreased. For left turn it was 4 s to 2.7 s for driver type 1 to type 10 and for right turn it was 6 s to 3.6 s. The mean start-up delay and mean discharge headway was also decreased and kept at 1.0 second and 1.4 second respectively for cross streets on which flow was high. In SimTraffic, headway at 50 mph was decreased and kept 1.6 to 0.93 for different driver types. The gap acceptance factor was also decreased and the value was from 1.0 to 0.85 for different driver types. When these changes were made, CORSIM and SimTraffic produced comparable results. In Paramics, to balance the flow between two intersections, extra links were added. These extra links represent the entrance to and exit from existing shopping centers, parking lots or any minor streets in reality. Initially, mean headway and mean reaction time was set to 0.3 second. In Paramics animation, the congestion was found at junction where extra link joined the main route. To calibrate the flow in Paramics these links were positioned as they are in reality. Because Paramics does not provide an initialization period a higher portion of the flow from the intermediate links was generated during the first 15 to 20 minutes to reach equilibrium. After these changes were implemented, Paramics gave results that were consistent with real-world counts.

Table 22 through Table 25 show the flow obtained from CORSIM, SimTraffic and Paramics simulation runs for Route 10.

Table 22 Route10 flow on each link (Eastbound)

Direction towards	**Actual**	**CORSIM**	**% Error**	**SimTraffic**	**% Error**	**Paramics**	**% Error**
Route 202	**3113**	3108	**-0.2**	3221	**3.5**	3044	**-2.2**
Ridgedale Ext	**3347**	3380	**1.0**	3491	**4.3**	3196	**-4.5**
Jefferson Ave	**2600**	2677	**3.0**	2758	**6.1**	2483	**-4.5**
Pine Place	**1430**	1455	**1.7**	1420	**-0.7**	1382	**-3.4**
Whippany Rd	**1865**	1849	**-0.9**	1850	**-0.8**	1752	**-6.1**
Troy Hills Rd	**2178**	2155	**-1.1**	2239	**2.8**	2063	**-5.3**
Algonquin Pkwy	**1744**	1675	**-4.0**	1878	**7.7**	1698	**-2.6**
Ridgedale Ave	**1608**	1569	**-2.4**	1501	**-6.7**	1588	**-1.2**
U-Turn	**1098**	1055	**-3.9**	1089	**-0.8**	1019	**-7.2**
Mt. Pleasant Conn	**690**	701	**1.6**	712	**3.2**	722	**4.6**
Mt. Pleasant Ave	**581**	546	**-6.0**	577	**-0.7**	545	**-6.2**
River Rd	**715**	715	**0.0**	707	**-1.1**	724	**1.3**
Okner Pkwy	**772**	752	**-2.6**	702	**-9.1**	800	**3.6**
Walnut St	**764**	714	**-6.5**	680	**-11.0**	801	**4.8**
Eisenhower Pkwy	**1226**	1197	**-2.4**	1160	**-5.4**	1212	**-1.1**
W. Northfield Ave	**446**	463	**3.8**	482	**8.1**	463	**3.8**
RMSP		**3.1**		**5.5**		**4.2**	

Table 23 Route10 flow on each link (Westbound)

Direction towards	Actual	CORSIM	% Error	SimTraffic	% Error	Paramics	% Error
Route 202	**2196**	2091	**-4.8**	2056	**-6.4**	1997	**-9.1**
Ridgedale Ext	**2438**	2394	**-1.8**	2295	**-5.9**	2370	**-2.8**
Jefferson Ave	**1490**	1463	**-1.8**	1400	**-6.0**	1455	**-2.3**
Pine Place	**1509**	1462	**-3.1**	1429	**-5.3**	1481	**-1.9**
Whippany Rd	**2144**	2133	**-0.5**	2071	**-3.4**	2073	**-3.3**
Troy Hills Rd	**1683**	1697	**0.8**	1597	**-5.1**	1594	**-5.3**
Algonquin Pkwy	**1292**	1270	**-1.7**	1302	**0.8**	1357	**5.0**
Ridgedale Ave	**1066**	1105	**3.7**	1006	**-5.6**	1114	**4.5**
U-Turn	**1871**	1862	**-0.5**	1792	**-4.2**	1840	**-1.7**
Mt. Pleasant Conn	**1550**	1546	**-0.3**	1650	**6.5**	1456	**-6.1**
Mt. Pleasant Ave	**203**	205	**1.0**	196	**-3.4**	222	**9.4**
River Rd	**1852**	1894	**2.3**	1775	**-4.2**	1757	**-5.1**
Okner Pkwy	**1235**	1264	**2.3**	1125	**-8.9**	1172	**-5.1**
Walnut St	**1659**	1734	**4.5**	1706	**2.8**	1569	**-5.4**
Eisenhower Pkwy	**1316**	1333	**1.3**	1214	**-7.8**	1331	**1.1**
W. Northfield Ave							
RMSP		**2.4**		**5.4**		**5.1**	

Table 24 Cross street flow (Northbound)

Cross street	Actual	CORSIM	% Error	SimTraffic	% Error	Paramics	% Error
Route 202	**925**	925	**0.0**	965	**4.3**	985	**6.5**
Ridgedale Ext	**552**	550	**-0.4**	597	**8.2**	597	**8.2**
Jefferson Ave	**675**	630	**-6.7**	611	**-9.5**	681	**0.9**
Pine Place	**95**	94	**-1.1**	97	**2.1**	93	**-2.1**
Whippany Rd	**548**	546	**-0.4**	545	**-0.5**	537	**-2.0**
Troy Hills Rd	**204**	195	**-4.4**	200	**-2.0**	164	**-19.6**
Algonquin Pkwy	**465**	429	**-7.7**	420	**-9.7**	468	**0.6**
Ridgedale Ave	**523**	522	**-0.2**	514	**-1.7**	508	**-2.9**
U-Turn	**93**	88	**-5.4**	96	**3.2**	105	**12.9**
Mt. Pleasant Conn	**278**	289	**4.0**	286	**2.9**	292	**5.0**
Mt. Pleasant Ave							
River Rd	**317**	309	**-2.5**	330	**4.1**	327	**3.2**
Okner Pkwy	**109**	109	**0.0**	112	**2.8**	113	**3.7**
Walnut St	**441**	443	**0.5**	408	**-7.5**	439	**-0.5**
Eisenhower Pkwy	**528**	527	**-0.2**	500	**-5.3**	511	**-3.2**
W. Northfield Ave	**775**	717	**-7.5**	648	**-16.4**	739	**-4.6**
RMSP		**3.9**		**6.7**		**7.1**	

Table 25 Cross street flow (Southbound)

Cross street	DOT	CORSIM	% Error	SimTraffic	% Error	Paramics	% Error
Route 202	**921**	878	**-4.7**	915	**-0.7**	914	**-0.8**
Ridgedale Ext	**45**	45	**0.0**	48	**6.7**	45	**0.0**
Jefferson Ave	**259**	257	**-0.8**	263	**1.5**	282	**8.9**
Pine Place	**60**	62	**3.3**	55	**-8.3**	62	**3.3**
Whippany Rd	**828**	826	**-0.2**	828	**0.0**	862	**4.1**
Troy Hills Rd.	**451**	448	**-0.7**	466	**3.3**	462	**2.4**
Algonquin Pkwy	**373**	375	**0.5**	371	**-0.5**	369	**-1.1**
Ridgedale Ave	**727**	727	**0.0**	738	**1.5**	714	**-1.8**
U-Turn	**144**	140	**-2.8**	144	**0.0**	140	**-2.8**
Mt. Pleasant Conn	**253**	251	**-0.8**	260	**2.8**	266	**5.1**
Mt. Pleasant Ave	**53**	57	**7.5**	50	**-5.7**	55	**3.8**
River Rd	**1518**	1278	**-15.8**	1423	**-6.3**	1463	**-3.6**
Okner Pkwy	**342**	340	**-0.6**	352	**2.9**	333	**-2.6**
Walnut St	**129**	128	**-0.8**	129	**0.0**	125	**-3.1**
Eisenhower Pkwy	**648***	677	**4.5**	720	**11.1**	677	**4.5**
W. Northfield Ave							

RMSP	**4.9**	**4.7**	**3.7**

Conclusion of Calibration Efforts

This section presents a comprehensive approach for calibrating simulation models and an explanation for the inconsistency among delay estimates obtained from different simulations.

The conclusions drawn from the analysis of calibrating the network are as follows:

1. The first step in calibrating the network is to verify the geometry of the network. It was observed that incorrect lane channelization even at one link can be the cause of erroneous flows.
2. Different softwares have different driver behavior and vehicle behavior calibration parameters. However, most frequently used parameters to calibrate the flow are headway between vehicles and gap acceptance for right and left turns.
3. The other important factor to be considered for calibrating the network for flow is the initialization period. It is studied that initialization period in CORSIM, SimTraffic, and Paramics influences the simulation results to a great extent.
4. To obtain statistically acceptable results, simulation runs must be taken for different seed values.

MACROSCOPIC SIMULATION OF ADAPTIVE CONTROL STRATEGIES

Introduction

The lack of readily available computer implementations of popular adaptive signal control algorithms required the development of prototypes of these algorithms from scratch using the information available in the literature. Prototypes of adaptive control strategies of SCOOT, SCATS, and OPAC were thus developed using two different programming languages (C and Visual Basic) to perform macroscopic simulations. It is faster to run simulation in C language compared with any of the other simulation packages mentioned previously. These faster simulations would also help in understanding the limitations of the prototypes and developing solutions to the possible problems that might arise. This situation was especially encountered in OPAC where three prototypes were being developed before conducting OPAC simulations in the Paramics simulation model described in the previous chapter. These prototypes were then connected to the expert system developed using the GeoMedia Pro GIS package and Synchro to macroscopically evaluate the behavior of different control strategies at various NJ intersections.

Each of these prototypes was prepared from mathematical formulations available in the literature. They are explained in the next sections of this chapter. A flow chart for each algorithm also depicts how these prototypes were implemented. A prototype for a pre-timed signal control strategy that works as part of the macroscopic simulation tool was also developed. It was used to compare the performance of the optimized pre-timed signal control technique with these adaptive control strategies. Finally, the simulation runs were performed using the developed prototypes at a test intersection to compare the individual performance of each strategy. Simulation results are presented and discussed in the last section of this chapter.

Research Methodology

Two types of simulations were performed to study the performance of adaptive prototypes. Macroscopic simulation models were developed using C and Visual Basic Programming Language. An API was developed using Paramics Programmer V3.0 to be used with Paramics simulation package for the microscopic simulation. Macroscopic simulation models were tested first, because these simulations run faster compared to microscopic simulation model. Finally these prototypes are applied to calibrate New Jersey State Route intersections discussed in the previous section. The simulation results for these intersections are analyzed to see under what type of network conditions the selected adaptive signal control strategies give benefits when compared with pre-timed signal control and when they fail to generate any benefits. Finally, benefit-cost analysis is performed to study the cost effectiveness of the developed prototypes.

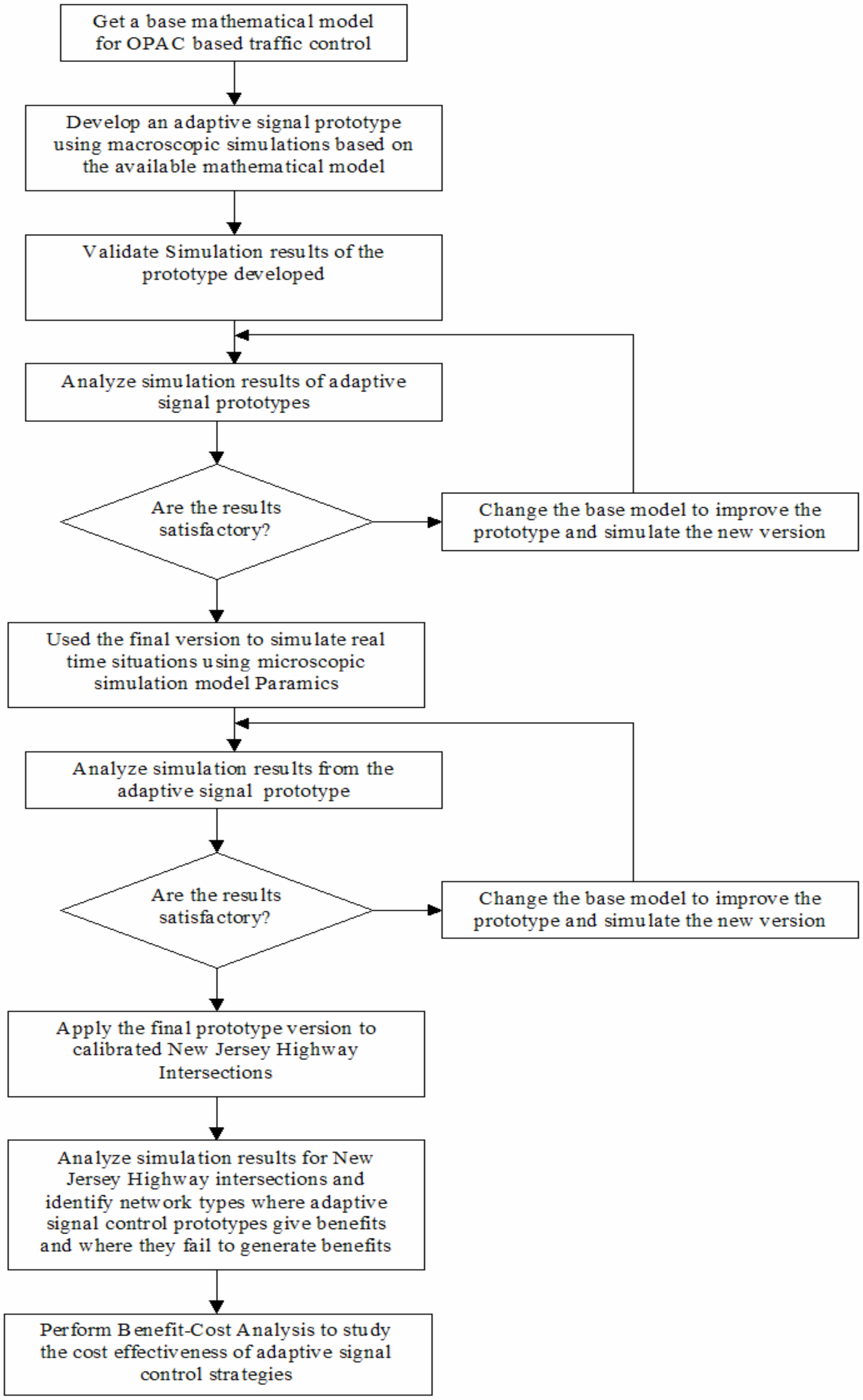


Figure 29 Research methodology for evaluation of adaptive signal prototypes

Pre-timed Signal Control Prototype

To compare the performance of adaptive signal prototypes with pre-timed signal control, a pre-timed signal control prototype was first developed. The algorithm used to develop this prototype is described below:

Step 1 – Generate arrivals at the intersection based on the total volume per lane per approach at the intersection.

Step 2 – Apply a user-defined timing plan to depart vehicles from the intersection based on a defined saturation flow rate.

Step 3 – Vehicles would depart from the intersection if the signal status is green; otherwise, arrivals in the mean time will be added to the existing queue. Development of the pre-timed signal control prototype was essential to validate the macroscopic model to be used for simulation and analysis of adaptive signal prototypes. There are many ways of checking the validity of a model:

- Compare data results obtained from the model with actual data obtained in the field.

- Compare results with a second model, whose results have been proven to resemble real-life situations.

The second approach was used to validate the results of the developed macroscopic model. Synchro was used to compare the results of the developed prototype. Synchro is one of the most reliable and realistic traffic simulation models. The results were compared for similar conditions and identical parameter values (e.g., saturation flow rate, green time etc.) in both Synchro and the developed prototype. The results from the two models were then compared for the validation process. Table 26 shows the accuracy of the developed prototype compared with Synchro:

Table 26 Comparison of simulation results of pre-timed control prototype with Synchro

Volume (vph)		**Synchro**				**Pre-timed Control Prototype**			
		Delay (seconds/vehicle)		**Delay (vehicle-hours)**		**Delay (seconds/vehicle)**		**Delay (vehicle-hours)**	
Phase 1	**Phase 2**	**Phase 1**	**Phase 2**	**Phase 1**	**Phase 2**	**Phase 1**	**Phase 2**	**Phase 1**	**Phase 2**
700	200	9.2	16.2	1.79	0.90	10.14	14.09	1.97	0.78
800	200	10.9	18.8	2.42	1.04	11.21	18.46	2.49	1.03
900	200	16	18.8	4.00	1.04	13.96	15.94	3.49	0.89
1000	200	15.2	24.2	4.22	1.34	12.05	22.87	3.35	1.27
1100	200	17.3	29.6	5.29	1.64	13.36	28.69	4.08	1.59
1200	200	22.4	35.7	7.47	1.98	16.71	38.38	5.57	2.13

According to Table 26 we see that the macroscopic model provides similar results compared with Synchro. At higher volumes, the prototype model gave lower delay than Synchro because it considers only stopped delay, whereas Synchro considers both stopped and control delay. At higher volumes, control delay is significant and hence the results vary. Overall, the results were comparable and the base prototype for pre-timed control was used for the implementation of adaptive signal control prototypes.

Macroscopic Simulation of OPAC (11)

The OPAC model was first developed by PB-Farradyne and the University of Massachusetts-Lowell. OPAC is based on the idea that if the vehicle arrival rate at the intersection is known, it would be possible to decide a signal switching strategy that will optimize a measure of effectiveness (delay, emission, etc.) for the intersection.

For practical implementation of this concept, OPAC uses the rolling horizon technique, which optimizes a given performance index over a “horizon length.” The horizon length is divided into equal time intervals, typically 5 seconds. The horizon is divided into two parts – head/roll period (typically 15 seconds) and tail/projection period (typically 60 seconds). The roll period uses detector (or real-time) data for arrivals, whereas the tail period uses arrivals from a model based on historic data. These arrivals are used to calculate the switching strategy to optimize a performance index.

This strategy can operate as an independent smart controller, or as part of a coordinated system. It considers a single intersection with two phases and traffic coming from two directions. The strategy is then extended to ‘n’ number of phases and *n* lanes in each phase. Modifications are made in the control strategy after studying the simulation results. As a result, three prototypes of OPAC have been developed, namely OPAC1, OPAC2, and OPAC3. These prototypes are purely based on mathematical formulations available in literature and may not exactly represent the version of OPAC currently in use.

OPAC-Like 1 Prototype

A basic version of OPAC-Like was developed using available mathematical formulations. This version is called OPAC-Like 1. This control makes the decision to change or not to change the current signal status at short and fixed time intervals. This decision is purely based on the current status of the signal (green or red), queue length at each approach at the beginning of each interval and as well as the number of vehicles that will arrive on each approach during the interval. If the decision to change a signal is made, no movement is assumed on either intersection approaches for one interval (which is similar to amber and all red times). The mathematical formulation available in literature for such a strategy is given below:

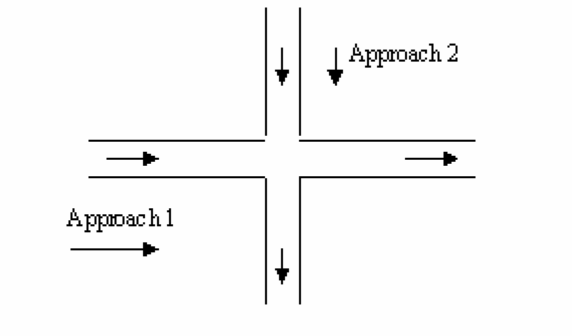


Figure 30 Isolated intersection with two approaches and two phases used in OPAC-Like prototype simulation

Let,

$q_{ij}=$A state variable describing the length of the queue on approach *j*, *j*=1,2.

$s_{ij}=$A state variable describing the status of the signal on approach *j*, *j*=1,2. A value of 0 corresponds to green and 1 to red. Note that if $s_{i1}=0$, then $s_{i2}=1$ and vice versa.

$a_{ij}=$A state variable describing the number of arrivals on approach j during interval *i*, *i* = 1,2…..*N*

$d_i=$The decision variable associated with interval *i*. A value of 0 corresponds to a decision of no switch while a value of 1 corresponds to a decision to switch the signal.

$Q_i=[q_{i1},q_{i2}]$= Vector containing length of the queue on approach j (j=1,2) for interval *i*.

$A_i=[a_{i1},a_{i2}]$= Vector containing arrivals on each approach for interval *i*.

$S_i=[s_{i1},s_{i2}]$ = Vector containing signal status at the beginning of interval *i*.

$C(Q_i,S_i,A_i,d_i)=$Cost function incurred at stage i, as a result of decision d_i

Then,

$$C(Q_i,S_i,A_i,d_i)=\begin{cases} T*(q_{i1}+q_{i12}+\frac{1}{2}(a_{i1}+a_{i2})) & d_i=1 \\ T*(q_{i1}+\alpha_{i2}+\frac{1}{2}(a_{i1}+\max(0,a_{i2}-\alpha_{i2}))) & d_i=0, s_{i1}=1 \\ T*(q_{i2}+\alpha_{i1}+\frac{1}{2}(a_{i2}+\max(0,a_{i1}-\alpha_{i1}))) & d_i=0, s_{i1}=0 \end{cases} \quad (4)$$

Where, *T* = length of the discrete time interval.

$$\alpha_{ij} = \begin{cases} \left(2 * \frac{T}{5}\right) - q_{ij} & \text{If, } \left[\left(2 * \frac{T}{5}\right) - q_{ij}\right] \geq 0 \\ 0 & \text{Otherwise.} \end{cases} \quad (5)$$

(Assumption: The discharge rate is 2 vehicles per 5 seconds, this is however changed later in the paper where the discharge rate is function of the saturation the flow rate).

The queue length after each interval on each approach would depend on the decision variable as well as the initial signal status. The queue length can be defined as:

$$q_{i+1,j} = \max(0, q_{ij} + a_{ij} - 2[(1 - s_{ij})\{1 - (s_{ij} + d_i)_{MOD2}\}] \quad (6)$$

The signal status after each decision variable will be

$$s_{i+1,j} = (s_{ij} + d_i)_{MOD2} \quad (7)$$

Based on the dynamic programming principle of optimality, the best decision d_i at each stage *i*, will be the one that minimizes the performance measure from stage *i* to stage *i* = *N*.

$$OP_i = \min_{d_i = 0,1} C(Q_i, S_i, A_i, d_i) + OP_{i+1} \quad (8)$$

(where OP_i is the optimal value of performance measured from stage *i* onwards). The flowchart used to implement the above algorithm is shown in Figure 31.

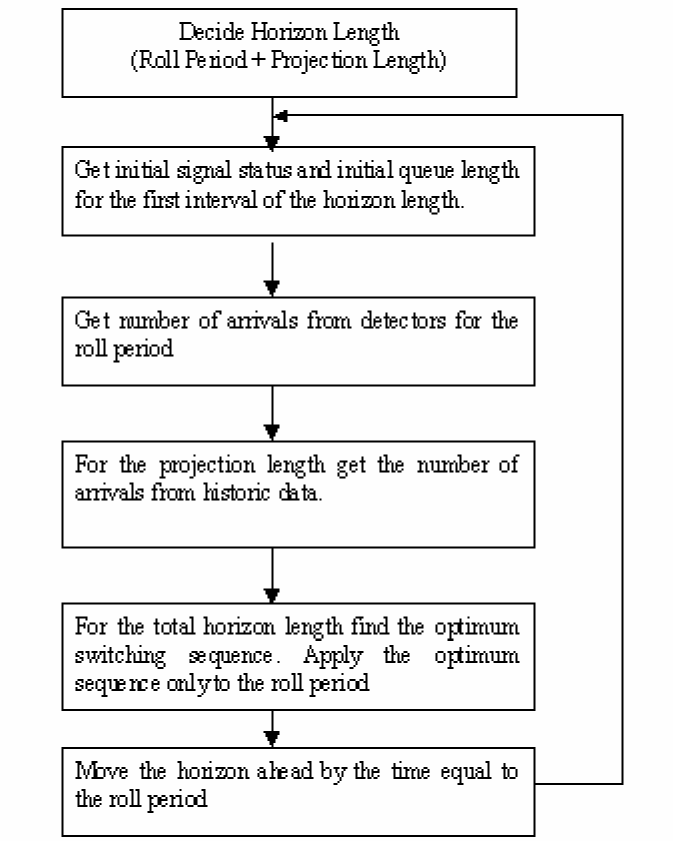


Figure 31 Flowchart for implementation of OPAC-Like 1 strategy

On the whole, the results for OPAC-Like 1 were not good even though this strategy was able to reduce traffic delay by a considerable amount compared with optimized pre-timed. In-depth analysis of this strategy highlighted its shortcomings and the OPAC-Like 1 strategy was modified to be more practical to implement. Table 32 through Table 37 and Figure 45 through Figure 50 show experimental results compared with Pre-timed control. The computational experiences with OPAC-Like 1 simulations are listed below:

1. The OPAC-Like 1 strategy used here does not consider cycle length. This results in serious drawbacks. For an isolated intersection with heavy traffic, OPAC-Like 1 gives undesirably high cycle length (around 3 cycles in one hour of simulation) as shown in Figure 32. Modern day controllers (such as NEMA) can support a cycle length up to a maximum of 255 seconds, and hence, such high values of cycle lengths are not desired.

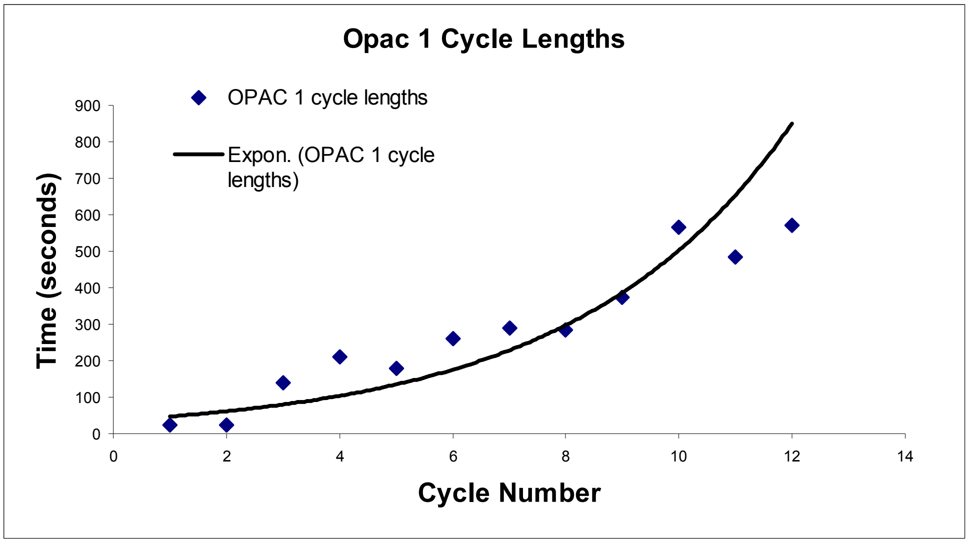


Figure 32 OPAC-Like 1 cycle length for test intersection with three phase and three lanes with traffic 1000, 700 and 200 vphpl

2. When applying the algorithm to a congested arterial, it becomes necessary to have cycle lengths of all intersections based on critical intersection cycle length, thus increasing the throughput. OPAC-LIKE 1 fails to control the cycle length.

3. OPAC-Like 1 tends to optimize the total delay on an intersection. Even though OPAC-Like 1 effectively utilizes capacity by assigning proper “green times” to both approaches at the proper time, thus reducing the overall delay, delay on cross streets where the traffic is low is undesirably high. This may not be practical in real use.

4. Because of the nature of the algorithm, the signal status may oscillate between green and red for consecutive time steps if the traffic scenario (queue lengths and future arrivals) on two approaches are similar. This means that there are chances that after each time step a decision to change the signal would be taken. The condition is shown in the output file table (Table 27), generated from macroscopic simulation.

Table 27 Oscillatory nature of OPAC-Like 1

Simulation	Queue		Future Arrivals		Signal Status	
Time (sec)	Approach 1	Approach 2	Approach 1	Approach 2	Approach 1	Approach 2
1320	0	2	0	1	1	0
1325	0	0	0	0	1	0
1330	0	0	0	0	0	1
1335	0	0	0	0	1	0
1340	0	0	0	0	0	1
1345	0	0	0	0	1	0
1350	1	0	1	0	0	1
1355	0	0	1	0	0	1

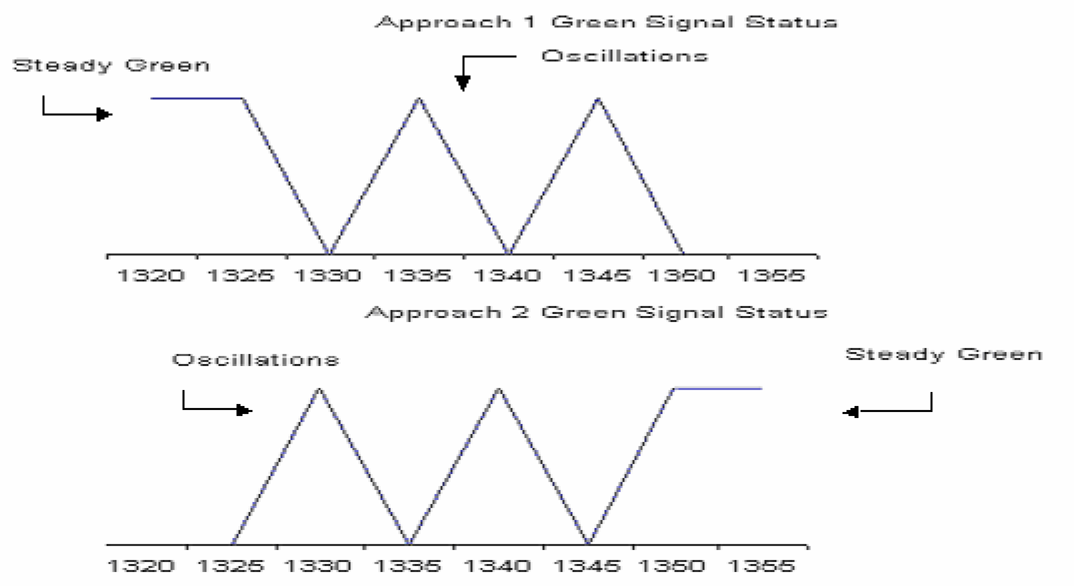


Figure 33 Oscillatory behavior of OPAC-Like 1 algorithm

5. At a high demand level on the main street and a low demand on the cross street, OPAC-Like 1 does not dissipate the cross street queue completely before changing its signal status to red. This might be justified with the fact that the cross street queue will dissipate during off-peak hours, however, this nature is not be desired unless the intersection is highly over saturated.
6. The above algorithm has too many assumptions, which makes it impossible to work well in real time conditions. Initial Paramics simulations failed to generate any benefits.

The above shortcomings discussed for OPAC-Like 1 lead to some modifications in the original algorithm so that it is practical in implementation. OPAC-Like 2 (with minimum green time constraints) and OPAC-Like 3 (with both minimum green and maximum green time constraints) were then tested in a macroscopic simulation program written in C and the microscopic Paramics traffic simulation package. Macroscopic simulation results showed that OPAC-Like 1 has less delays than OPAC-Like 2 and OPAC-Like 3, however OPAC-Like 3 was found to be practical to implement in Paramics.

OPAC-Like 2 Prototype

OPAC-Like 2 introduced the concept of minimum green time. This means that once the signal status is changed it will remain green for a specified time interval until OPAC-Like 2 module can act on it. This is especially needed at the cross street for queue dispersion under moderately heavy demand as well as to take care of the oscillatory behavior under similar demand on main and cross streets. The objective here is to minimize the sum of delay over the entire horizon for all possible phase sequences with an additional constraint of minimum green times also. The sequence, which gives the minimum delay, is then selected for the roll period.

Objective Function: *min* $\mathrm{C}(\mathrm{Q}_{\mathrm{i}}, \mathrm{A}_{\mathrm{i}}, \mathrm{D}_{\mathrm{i}}, \mathrm{t})$ (9)

Subject to: $t \geq (\min green)_k$, where *t* represents possible switching times (after minimum green time is elapsed).

To check whether OPAC-Like 2 keeps cycle length within acceptable limits, simulation was abruptly ended when cycle length is too long. The experimental results of OPAC-Like 2 prototype are compared with OPAC-Like 1 and OPAC-Like 3 and Optimized Pre-timed control in Table 32 through Table 37 and Figure 45 through Figure 50. The flowchart used to implement OPAC-Like 2 algorithm is shown in Figure 35.

Computational experiences with OPAC-Like 2 are listed below:

1. With the introduction of minimum green time, OPAC-Like 2 is able to keep cycle time within acceptable cycle length for some demand conditions, whereas OPAC-Like 1 is unable to limit its value. Hence introduction of minimum green time has a positive effect.
2. At high demands on main street and low demands on cross streets the cross street green time never exceeds minimum green time. Hence, for optimum control, the minimum green time of the cross street must be large enough such that it dissipates initial queue.
3. At high demand conditions the OPAC-Like 2 cycle length exceeds maximum allowable cycle length, which is not desired.
4. No oscillations were found in the output file for any cases in OPAC-Like 2.

Overall OPAC-Like 2 performs better than OPAC-Like 1 if minimum green time is selected properly, however, its inability to control the cycle length for high demand levels makes it impractical to implement in real-time conditions.

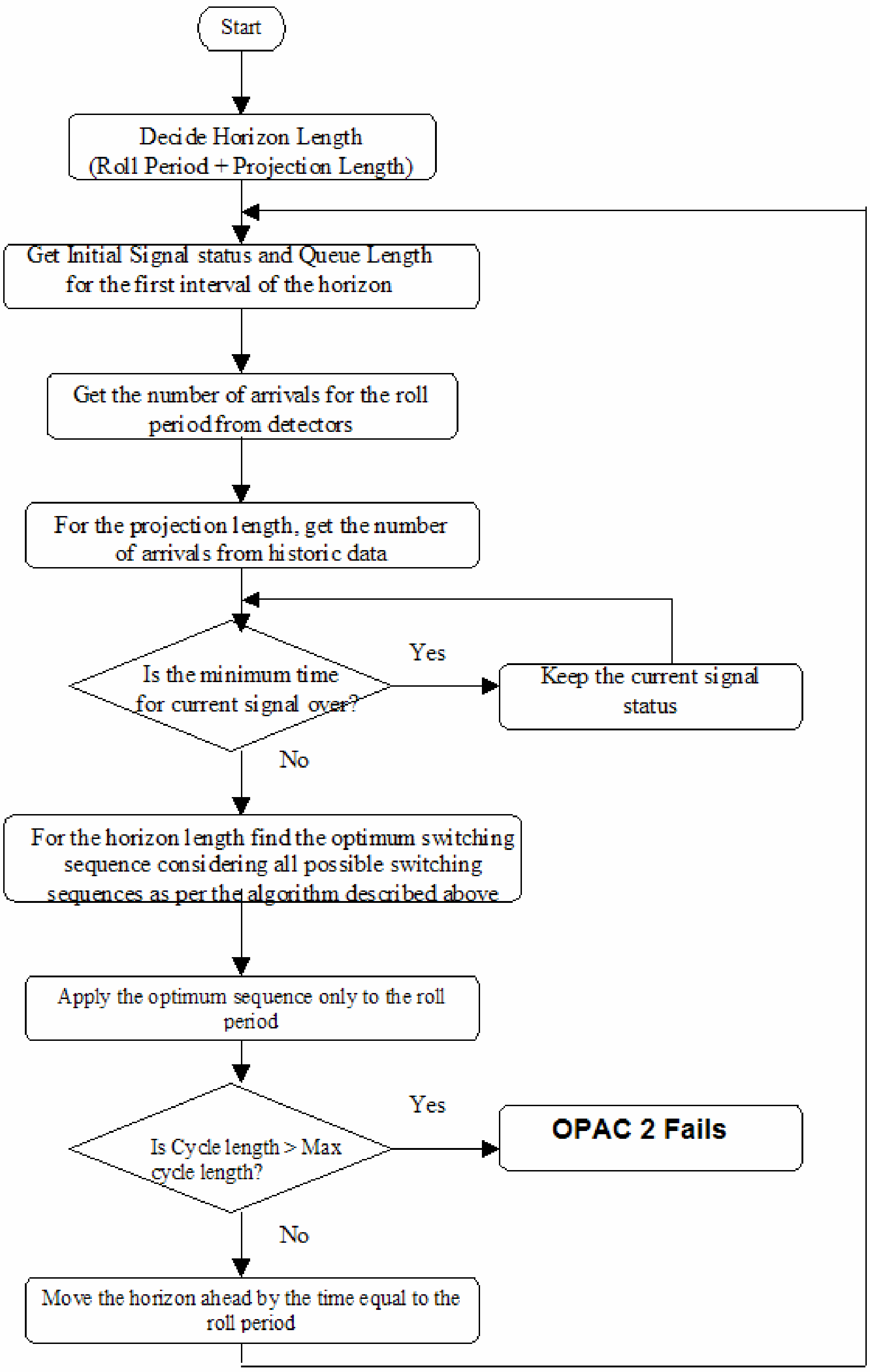


Figure 34 Flow chart used to implement OPAC-Like 2 algorithm

OPAC-Like 3 Prototype

OPAC-Like 3 uses one more constraint of maximum green time. This means that as soon as the signal remains green for a particular phase for pre-specified time (maximum green time), the signal changes to give green to the other phase. The additional constraint was introduced mainly to limit the cycle length, give enough green time to the cross-street, and possibly support network-wide operation.

The mathematical formulation has an additional constraint that the signal should switch when maximum green time of a phase is reached. The optimization process is shown below:

Objective Function: *min* $\mathrm{C}(\mathrm{Q}_{\mathrm{i}}, \mathrm{A}_{\mathrm{i}}, \mathrm{D}_{\mathrm{i}}, \mathrm{t})$ (10)

Subject to: 1) $t \geq (\min green)_k$

2) $t \leq (\max green)_k$

Where *t* represents possible switching times (between minimum green time and maximum green time for each phase). The results of the comparison of OPAC-Like 3 prototype with OPAC-Like 1, OPAC-Like 2 and Optimized Pre-timed control are shown in Table 32 through Table 37 and Figure 45 through Figure 50. Maximum green time was assumed to be 2.5 times the pre-timed green time but the maximum cycle length was 150 seconds. The computational experiences with OPAC-Like 3 simulations are listed below:

1. According to the above experimental results it is seen that there is not much difference in delays between the OPAC-Like 2 and OPAC-Like 3 strategy. Also with OPAC-Like 3 the cycle length be could limited.
2. For high demand level on main street and low demand level on cross street, it was observed that green time for main street is usually its maximum green time while for the cross street it is the minimum green time for that phase.
3. For high demand levels on both streets, the phase remains green for maximum green times.
4. With an increase in maximum green time the delays with OPAC-Like 3 decreases. However increasing the maximum green time beyond a certain value may not make any difference since OPAC-Like 3 will not increase the green time beyond a certain limit. It is important to determine this limit for the maximum green time.
5. OPAC-Like 3 is able to give lesser delays compared to Optimized Pre-timed control.

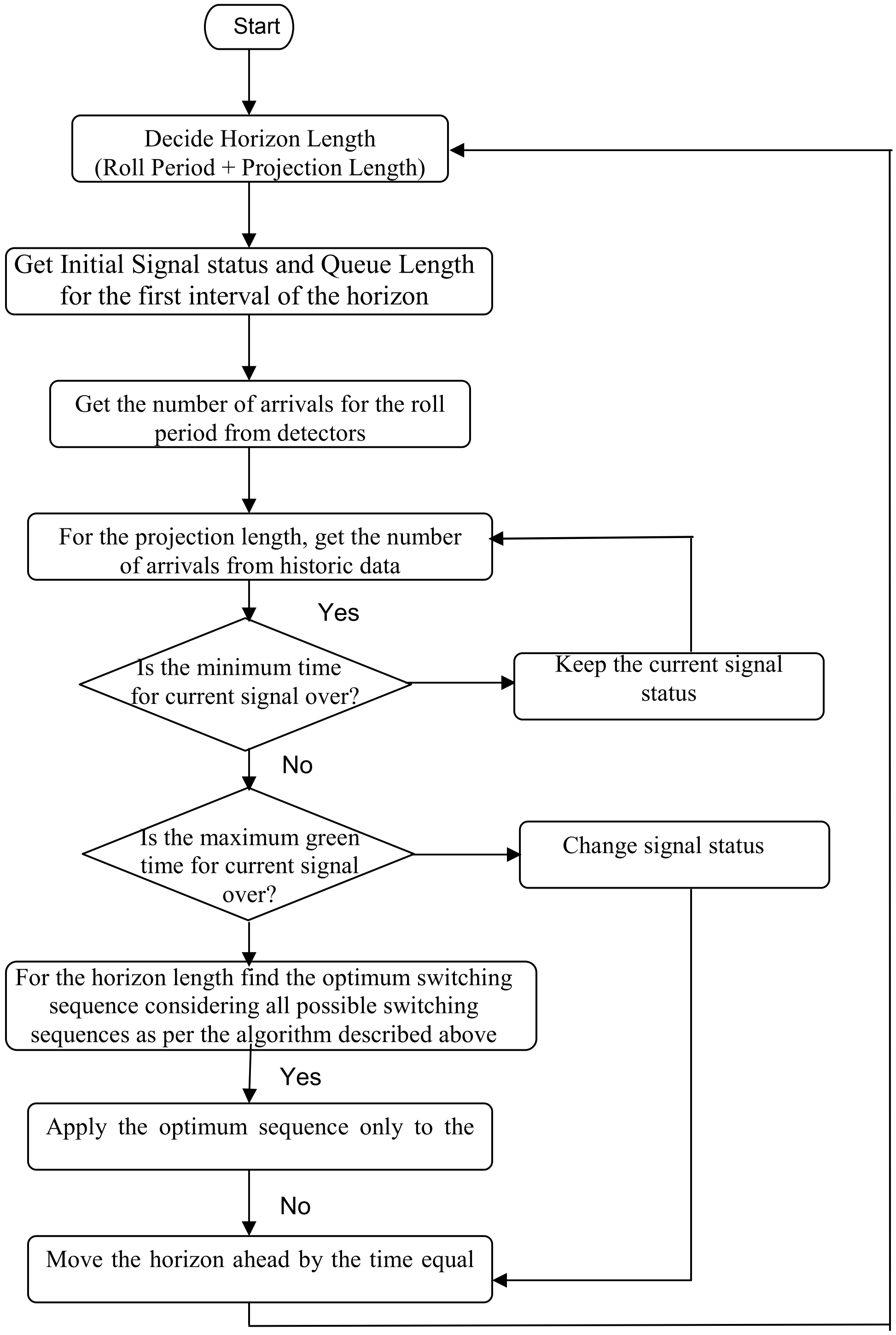


Figure 35 Flowchart used to implement OPAC-Like 3 algorithm

Macroscopic Simulation of SCOOT (5)

SCOOT optimizes phase timing by using data collected by detectors on the upstream of intersections. The SCOOT traffic model uses time-varying traffic data from detectors. This data is used to predict traffic queues, delays, and stops for optimizing splits, cycle lengths, and offsets. The detectors are located as far as possible from the signal stop lines. Ideally, immediately downstream of the adjacent signalized intersections, if detectors in this position, can monitor all major traffic streams that approach the stop line. This is shown in Figure 36.

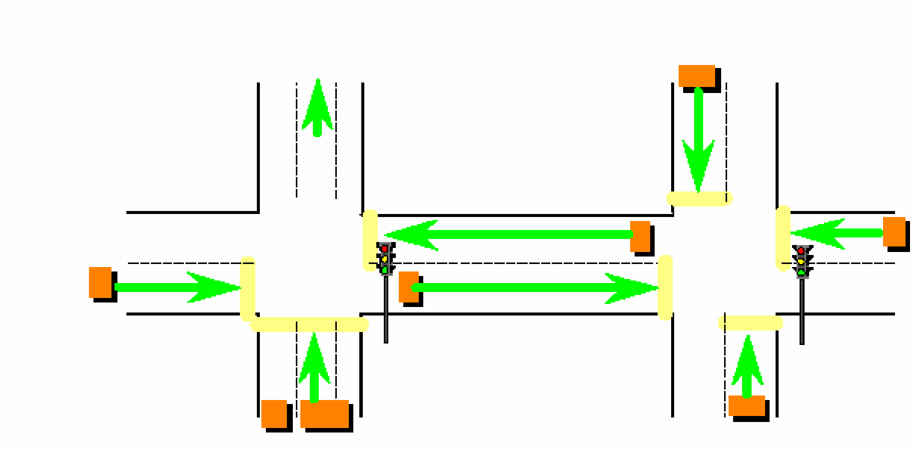


Figure 36 Location of detectors (19)

The data from detectors is stored in the SCOOT computer, in the form of “cyclic flow profiles” for each approach. Figure 7 shows an example of such a profile. (5) The cyclic flow profiles are based on the cycle time of the upstream signalized junction. These profiles are fundamentally important to the operation of SCOOT. The profile consists of a histogram that records how the traffic flow rate varied during one cycle time of the upstream signals. Queues, delays, and stops are predicted by using this file.

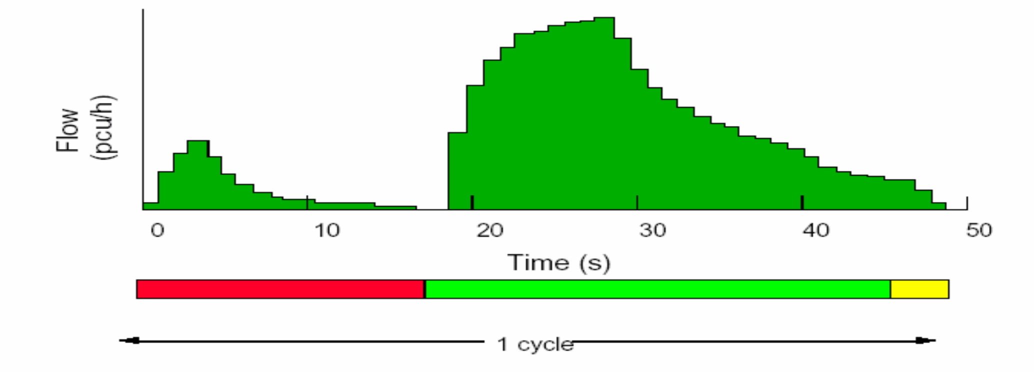


Figure 37 Cyclic flow profile (19)

According to the cyclic flow profiles on each section of street, the SCOOT traffic model predicts the current value of the queue at the downstream stop line. The maximum back-of-queue length is calculated as shown in Figure 38.

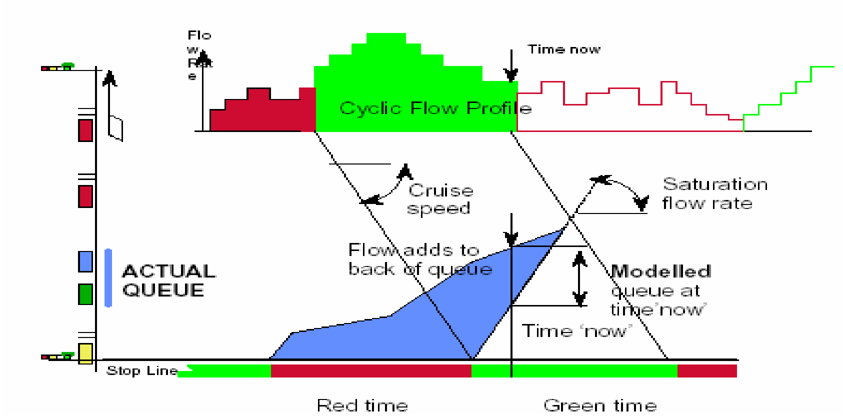


Figure 38 Queue modeling (19)

SCOOT calculates an average value for the sum of the queues at each signal stop line. Vehicles that reach the stop line during the red time are added onto the back of the queue, continually into the next green time until the queue clears. Vehicles discharge at the saturation rate when the signals are green, until all queues are cleared.

The performance index (PI) is introduced for measures of traffic behavior. PI can be a partial or all combination of the weighted average queue, total number of vehicle stops, and measured congestion.

$$PI = w_q Q + w_s N + w_c P \qquad (11)$$

Where,

w_q = link-specific weighting factor for queue.
Q_i= average queue for the sum of the queue, (vehicles).
w_s = weighting factor for stops.
N= total number of stops under the area.
w_c = weighting factor for measurement of congestion.
P = total of the proportion of the cycle time that the detector is occupied by a queue under the subsystem.

Cycle Time Optimization

Signalized junctions in an area controlled by SCOOT are grouped into sub areas. Usually, all intersections within a sub-area are operated by SCOOT on a common cycle length. In response to the change in demand, the SCOOT cycle optimizer varies the cycle length of each sub-area by a small amount (usually a few seconds). The frequency of this optimization is 5 minutes, and not less than 2.5 minutes. (5)

The SCOOT traffic model continuously measures the current degree of saturation for each movement in the sub-area. The most heavily loaded intersection determines the change of cycle length in the sub-system. The optimization process follows the below criteria:

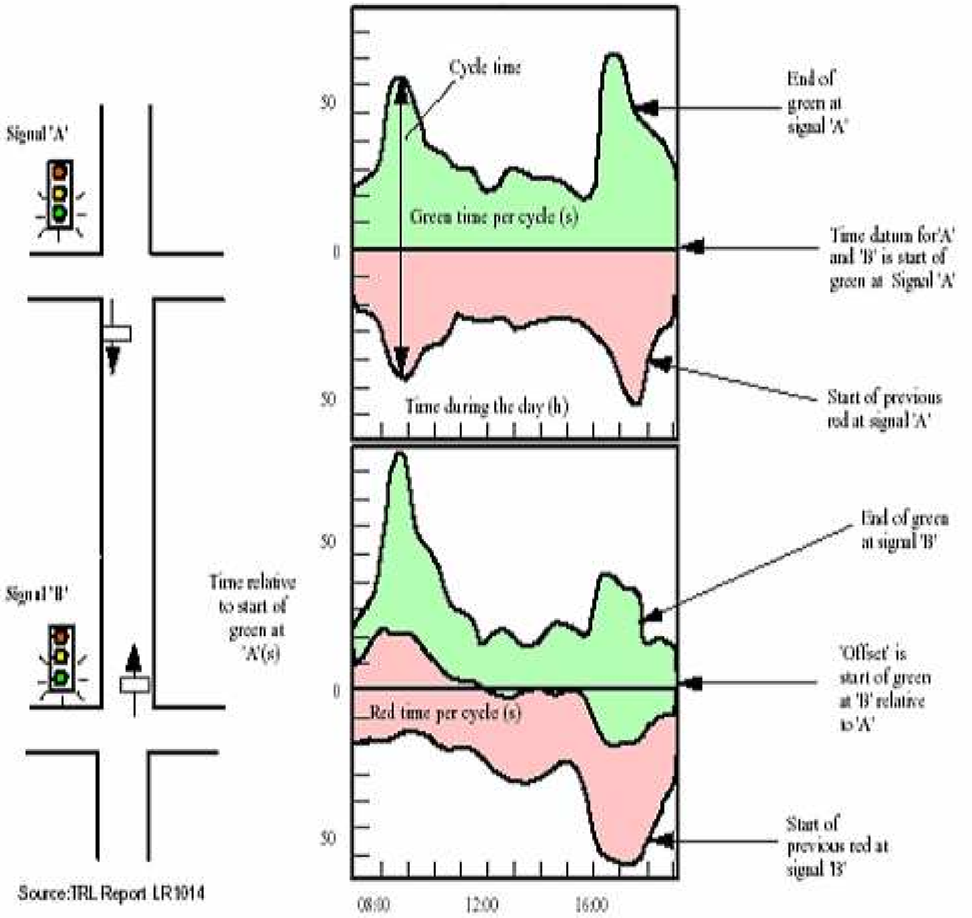
1. If the maximum $DS \le 80\%$ in the sub-system, the cycle optimizer will make reductions in the cycle length by 4, 8, 16 seconds. This decrease is considered to be 4 seconds in this study.
2. If $DS \ge 90\%$, i.e., at most the heavily loaded intersections the cycle optimizer will increment the cycle length by 4, 8, or 16 seconds to increase capacity. This increase is considered to be 4 seconds in this study.
3. If the maximum degree of saturation is in the range of 0.8 to 0.9, the cycle length maintains the same value.
4. The cycle length is constrained by the maximum cycle time of the region and minimum cycle time of the node.

Green Durations

The signal split optimizer estimates whether it is beneficial to make a change earlier, as scheduled, or later according to the degree of saturation and PI. The objective of this optimization is to minimize the maximum degree of saturation on the approaches to that junction. Any decision by the optimizer may alter a scheduled stage change time by no more than a few seconds. Usually the split time is temporarily changed by –4, 0, +4 and permanently changed by +1, 0, -1 seconds. Optimizers make a decision 5 seconds before the scheduled stage change. The duration of green time is constrained by the minimum green time, maximum green time, and fixed green time lengths.

Offset Optimization

The offset optimizer compares the sum of the PIs on all adjacent streets for the scheduled offset with offsets that occur a few seconds (-4, 0, +4) earlier or later. The offset with the minimal PI is selected. Figure 39 shows the change within a day. The changes of offset are constrained by fixed and biased offsets.



Source:TRL Report LR1014

Figure 39 Two coordinated signals during one day of SCOOT control (14)

The flow chart in Figure 40 shows the implementation logic of the SCOOT algorithm using the Visual Basic programming language. The same flow chart was used to write an API in Paramics to perform microscopic simulations.

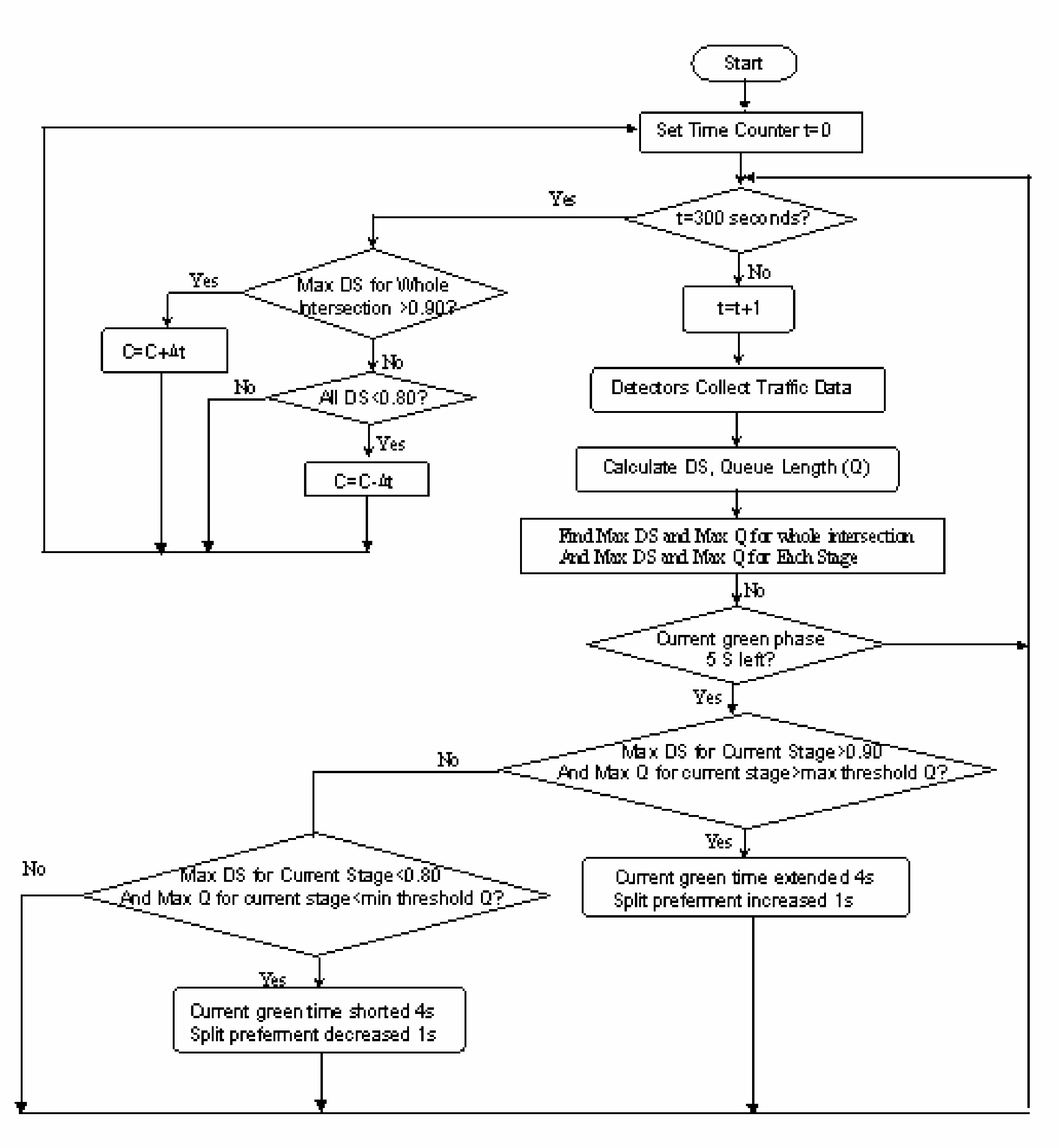


Figure 40 Flowchart for the implementation of SCOOT-Like prototype

MACROSCOPIC SIMULATION OF SCATS-LIKE ALGORITHM (9)

SCATS collects actual traffic data within the system from vehicle detectors on the road and then chooses the appropriate signal plan (pre-optimized) for each signal installation based on the current traffic situation. SCATS uses data from detectors that are used to calculate the degree of saturation. The location of the detectors is different from that in the implementation of SCOOT. They are placed immediately before the stop-line at each lane. (9) It is shown in Figure 41.

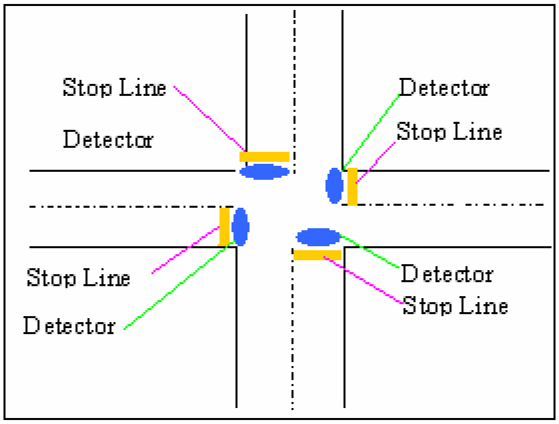


Figure 41 Location of detector on SCATS

Degree of saturation

The most important traffic parameter used by the SCATS algorithms is one analogous to the degree of saturation. It is defined as the ratio of effectively used green time to the total available green time. Mathematically, it is given as (1):

$$DS = g^{'} / g \tag{12}$$

$$g^{'} = (3600 / S) * n$$

where,
DS = degree of saturation,
g’ = effective green (seconds),
g = total available green (seconds),
n = number of cars pass that detector during green time,
S = saturation flow.

Synthesized Flow

The SCATS detector length (4.5m) prevents accurate counting of vehicles, especially during congested traffic conditions and problems of obtaining a Passenger Car Unit (pcu) equivalent flow and of the two valued nature of density as a function of flow, synthesized flow that calculated from the measured DS and maximum flow rate for each strategic detector is introduced.

$$q' = DS * g * \frac{s}{3600} \tag{13}$$

When DS=1, the synthesized flow is s/3600 vehicles per second of green. When DS > 1, q'/g > s/3600. This means that the flow exceeds the saturation flow.

Design of Phase Split Plan

Usually, there are four or eight green split plans for each intersection. SCATS specifies the percentage of stretch cycle length to be allocated to each phase. One phase must be nominated as a "stretch" phase. From the current measured degree of saturation, SCATS calculates the expected degree of saturation for each plan. To select a more appropriate plan, it is essential that the increments between phase times are small enough to achieve equal degrees of saturation. In fact, to ensure reliable plan voting, SCATS requires that the increments between plans at the stretch cycle length are limited to about 7 seconds. An example (32), for explanation green split plans, stretch phase and stretch cycle length is shown below. Assuming the following peak period phase splits are required:

Phase	Pm Peak
A	40
B	90
C	20
Total	150 seconds

These can be achieved by making "B" phase the stretch phase in the P.M. peak plan. Assuming the increments between plans are 5 seconds, the amount of stretch required can be calculated by

$$stretch = X - (Y + PSR), \tag{14}$$

where X = stretch phase time,
Y = other phase time,
PSR = phase split range (=3*increment for 4 plans).

In this example, stretch = 90 - (40+3*15) = 35

Hence, stretch cycle length =150 - 35 = 115 seconds.

When the cycle length equals the stretch cycle length, the following phase split applies:

Phase	Pm Peak
A	40
B	55
C	20
Total	115 seconds

Hence, the phase times for 4 plans at the stretch cycle length will equal 115 seconds.

Phase	1	2	3	4
A	40	35	30	25
B	55	60	65	70
C	20	20	20	20

Expressed as percentage of stretch cycle length, they are as follows

Phase	1(%)	2(%)	3(%)	4(%)
A	35	30	26	22
B	48	52	57	61
C	17	17	17	17

Split Plan Selection

A “split plan vote” is performed at each cycle. One split plan will be selected among all the split plans at the end of a cycle. If one split plan gets two votes for three subsequent cycles, the allocation of the cycle length will be according to this split plan. Otherwise the algorithm will keep the current cycle’s split plan to allocate the cycles. To perform vote, expected DS for each plan will be calculated according to the current measured degree of saturation. The formula used is as follows.

$$DS_{ij} = DS_{ic} * g_{ic} / g_{ij} \tag{15}$$

where,

DS_{ij} = degree of saturation on movement *i* under phase timing plan *j*,
DS_{ic} = degree of saturation on movement *i* under current phase timing plan,
g_{ij} = green time on movement *i* under phase timing plan *j*,
g_{ic} = green time on movement *i* under current phase timing plan,

First the maximum DS on each plan should be found. Then, the minimum DS whose plan gets a vote, is found among the maximum DS of each plan. If a plan gets two votes in three subsequent cycles, the allocation of a new cycle length to each phase will be according to this plan. Otherwise, the allocation of the new cycle length will be according to the current plan. The following is an example is provided to explain this idea. If a plan gets a vote in a cycle, “1” will be stored as a count. Otherwise, “0” will be stored. At the end of three cycles, the total count of each plan is calculated. This is shown in Table 28.

Table 28 Methodology for voting of phase timing plan

Plan	Cycle count			Total Votes
	1	2	3	
1	1	0	1	2
2	0	1	0	1
3	0	0	0	0
4	0	0	0	0

Calculation of Cycle Length

Signalized intersections are grouped into sub-systems. All intersections within a sub-system operate on a common cycle length. The sub-system cycle length (C) is a function of the highest "eligible" DS measured in the sub-system during the previous cycle. The cycle length change and cycle length are calculated as follows:

$$C'' = C + C'$$

$$C' = 60 * (Max(DS) - f(C)) \tag{16}$$

$$f(C) = \frac{0.9 - 0.5}{C_{\max} - C_s}(C - C_s) + 0.5$$

Where,
C'' = new cycle length (seconds),
C = previous cycle length (seconds),
C' = changes of cycle length (seconds),
$C_{\max} =$ maximum cycle length (seconds),
$C_s =$ medium cycle (seconds).

Investigating the function of f(c), if C = C_s, the value of f(c) is 0.5. Whereas if, C= $C_{\max}$, the value of f(c) is 0.9. Therefore, cycle length increments are more easily achieved at shorter cycle lengths. The function f(c) is approximately linear and shown in Figure 42.

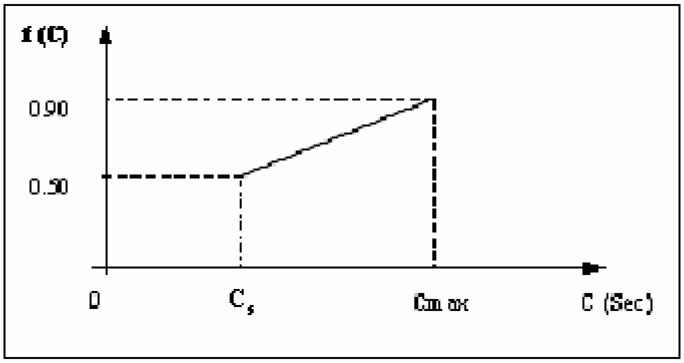


Figure 42 Function f(c)

The increment or decrement of the cycle length must be in the interval of [-6 s, 6 s]. If the calculated increment is greater than 6 s, it will then be assumed 6 s. On the other hand, if the calculated decrement is less than –6 s, it will be assumed to be –6 s.

Principle of Allocation of an Increment in a Cycle

Before implementing the allocation of an increment in a cycle, some parameters need to be defined.

$C_{\max} =$ maximum cycle (seconds)
$C_{\min} =$ minimum cycle (seconds)
$C_x =$ stretch cycle length (seconds)
$C_s =$ medium cycle (seconds)

The allocation of the increment to different splits in a cycle follows the rules below:

1. $C_{\min} \le C \le C_s$: If the rate of flow measured by the nominated strategic detectors falls below a preset value, the cycle length can only operate on $C_{\min}$. If the split of each phase equals the $C_{\min}$, multiply the percentage of each phase in the selected phase split plan.
2. $C_s \le C \le C_x$: Cycle length is allocated to each phase according to the percentage of each phase in the selected timing plan.
3. $C_x \le C \le C_{\max}$: C_x is allocated to each phase according to the percentage of each phase in the selected phase. The difference of $C - C_x$ goes to stretched phase.

Offset plans

In this version of implementation, each intersection is modeled as an isolated intersection without using the offset. However, for the sake of completeness, we give a description of SCATS offset logic is produced. Offset in SCATS is considered to be the time difference between the start of particular phases at adjacent intersections.

Five internal offset plans and five external offset plans are provided for each intersection in the sub-system and for between sub-systems, respectively. Offset plans are determined based on speed on the link, distance between adjacent intersections, and queue at the stop line.

Theoretically, because the offset is indicated in seconds, it is intrinsically independent of the cycle length. However, considering queuing or link speed changes during heavy traffic, it may be modified as a function of the cycle length.

For example, two of the offset plans are defined for highly directional flow patterns such as those experienced during the morning and evening peak periods. However, the two offset plans may be selected at the condition that flow is heavily biased in one direction but of less quantity. The basic offset in these two plans will not be entirely appropriate for any traffic condition and thus will need to be modified. The modified offset is given as a function of cycle length, expressed as follows:

$$P' = P[1 + A * g(C)] \qquad (17)$$

Where:

P' = modified offset in seconds
P = basic offset in seconds
A = specified modifying factor that could be positive or negative
$g(C)$ = a linear function of cycle length

When $C = C_{\max}$, $g(C)$ = 0; when $C \leq 0.75C_{\max}$, $g(C)$ =1.

The function $g(C)$ is shown in Figure 43.

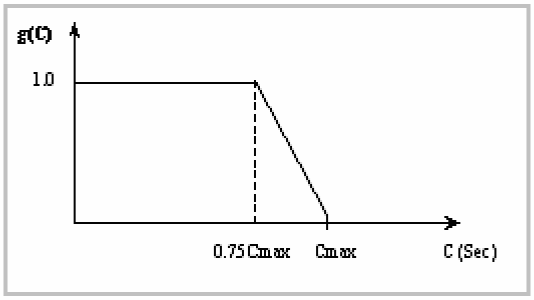


Figure 43 Function $g(C)$

Selection of Offset

Rules for selection of the offset are described as follows.

1. $C = C_{\min}$: Offset plan 1 is implemented at the intersection.
2. $C_s < C = C_s + 10$: Offset plan 2 is selected.
3. Selection of an offset plan among the remaining three plans is based on synthesized flow; it is performed once per cycle. Getting four votes for the same plan in any five consecutive cycles results in the selection of that plan. Before conducting selection, a plan vote bias figure is given for each strategic approach at each plan 3, 4, and 5. The procedures for the selection are as follows:
    - Select the approaches with the highest bias on the same plan.
    - Select the approach with the highest flow among approaches with highest bias.
    - Multiply the flows on the selected approaches by the corresponding bias.

- Sum the products obtained by the above procedure.

To explain the procedure of the offset selection, an example is provided below. In this intersection, there are four strategic approaches. The selection is from three offset plans, namely 3, 4, and 5. Biases for each strategic approach at each offset plan are given in the Table 20. And flows on each strategic approach are shown in the Table 30. The bold numbers in Table 29 represent the highest bias for each plan. According Table 29, the strategic approaches 2, 3, and 4 have the highest bias 0.3. Further observing the flow on the approach 2, 3, and 4 in the Table 30, strategic approaches 2 and 3 achieve the highest flow. Bias on strategic approaches 2 and 3 multiplied by the corresponding flow and sum the products. Similarly, summations of the products for offset plans 4, and 5 are calculated following the same procedure. The products are shown in the Table 31. Offset plan 5 with the highest product receives the vote.

Table 29 Bias factors

Strategic approach Offset Plan	1	2	3	4
3	0.1	**0.3**	**0.3**	**0.3**
4	0.4	**0.4**	0.10	0.10
5	0.15	**0.35**	**0.35**	0.15

Table 30 Flow on strategic approach

Strategic approach	1	2	3	4
Flow	120	**1000**	**1000**	100

Table 31 Product of flow and bias of selected approach

Offset Plan	Product
3	600
4	72
5	700

The flow chart in Figure 44 gives the implementation of the SCATS algorithm in the Visual Basic programming language. The same flow chart was used for writing an API in Paramics to conduct microscopic simulations.

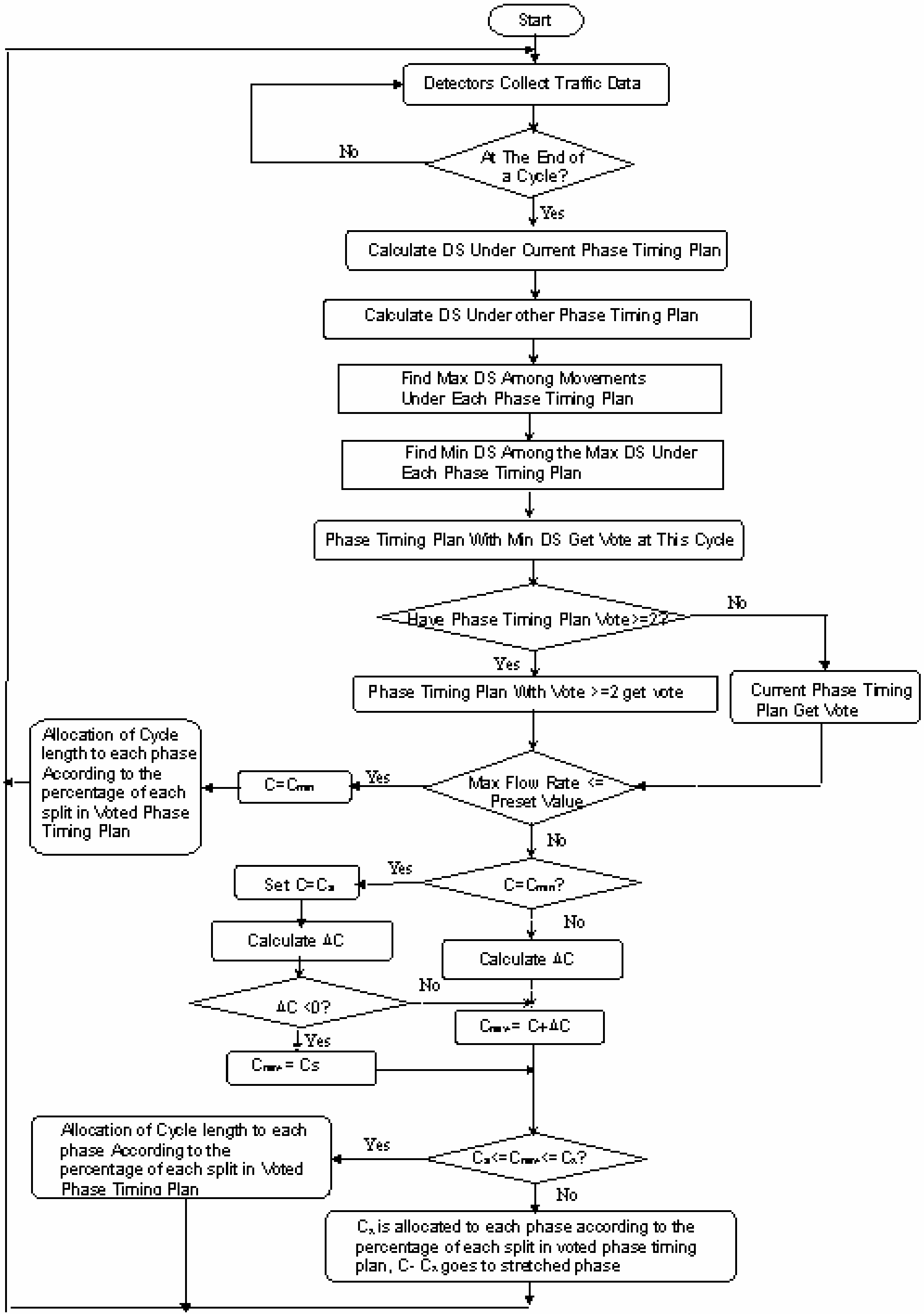


Figure 44 Flow chart of SCATS algorithm

MACROSCOPIC SIMULATION RESULTS

Simulation runs were carried out with SCOOT, SCATS and OPAC prototypes at a test intersection with two approaches and two lanes as shown in Figure 30. These simulations helped to compare the performance of three adaptive control strategy prototypes and with a base scenario (pre-timed control). The macroscopic simulation environment has less stochastic fluctuations compared with microscopic simulations. Three cases were considered for macroscopic simulations:

1. Two Phases; Main Street Demand of 700 vph to 1200 vph and Cross Street Demand 200 vph (LOW)
2. Two Phases; Main Street Demand of 700 vph to 900 vph and Cross Street Demand 700 vph (HIGH)
3. Three Phases; Main Street Demand of 700 vph to 1200 vph and Cross Street Demand 200 vph (LOW) for two remaining phases.

The results summarized in this section are taken from the final version of the prototypes developed.

Case 1

Table 32 Total delay for adaptive control systems for Case 1

Cross street demand: Low (200 vphpl)												
Main Street Traffic (vphpl)	Phase 1				Phase 2				Total Delay			
	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed
	Vehicle Hours				Vehicle Hours				Total Delay			
700	1.33	0.84	0.502	1.71	0.62	1.71	0.737	1	1.96	2.55	1.2	2.71
800	1.62	1.56	0.729	2.04	0.63	1.33	0.765	0.95	2.25	2.89	1.5	3
900	2.57	2.05	0.728	2.5	0.71	1.03	1.197	0.87	3.28	3.08	1.9	3.37
1000	2.25	2.3	0.884	3.06	1	1.5	1.477	1.12	3.25	3.8	2.4	4.18
1100	2.41	2.14	1.118	3.12	1.21	2.55	1.825	1.56	3.63	4.69	2.9	4.69
1200	2.5	1.79	1.5	3.8	2.02	2.87	2.33	1.77	4.52	4.66	3.8	5.57

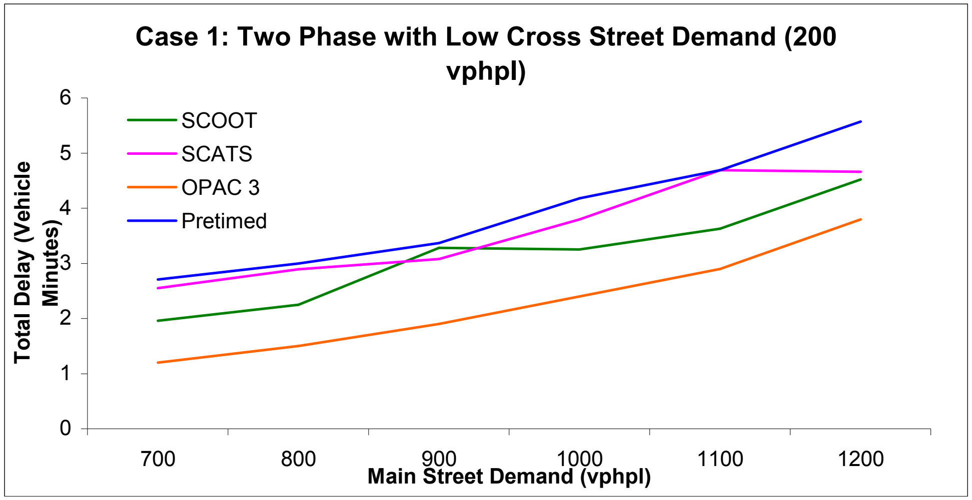
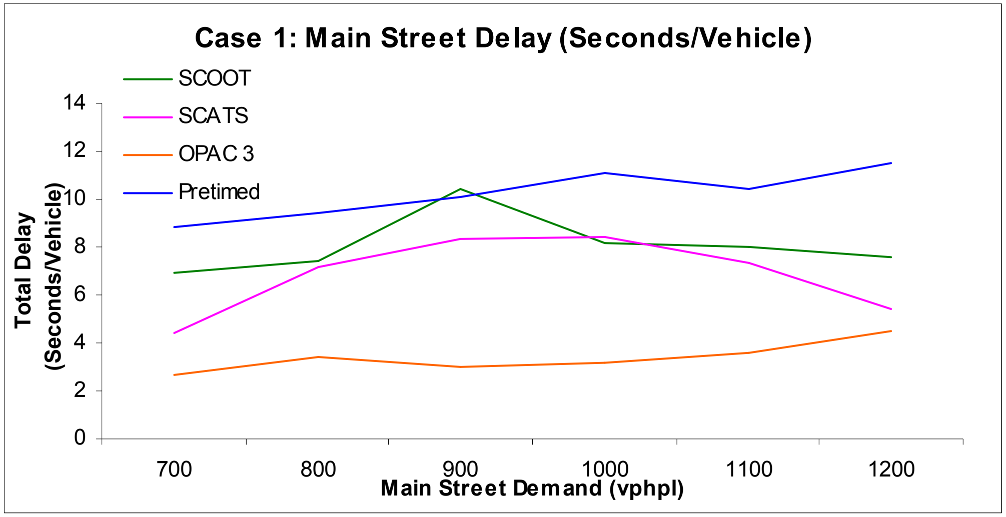


Figure 45 Total delay for adaptive control system for Case 1

Table 33 Average delay (seconds/vehicle) for adaptive control systems for Case 1

Cross street demand: Low (200 vphpl)								
Main Street Traffic	Phase 1				Phase 2			
	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed
(vphpl)	Seconds/Vehicle				Seconds/Vehicle			
700	6.89	4.39	2.7	8.85	11.45	31.34	13.7	18.4
800	7.45	7.18	3.4	9.4	10.97	23.28	13	16.66
900	10.39	8.3	3	10.1	13.64	19.6	22.1	16.85
1000	8.2	8.39	3.2	11.1	18.72	28.11	25.7	21
1100	8.04	7.31	3.6	10.41	22.16	46.31	35.7	28.49
1200	7.61	5.45	4.5	11.5	37.5	53.3	42.4	32.89



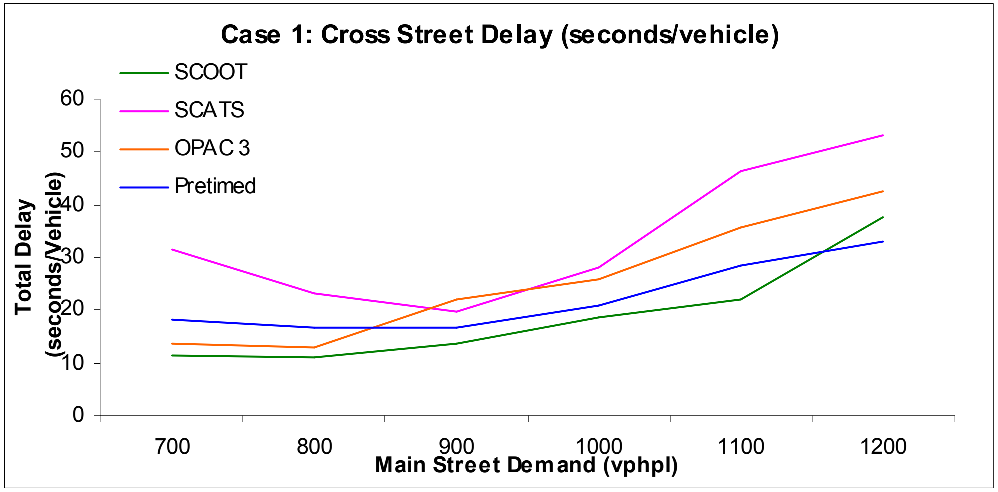


Figure 46 Average delay (seconds/vehicle) for adaptive control system for Case 1

Case 2

Table 34 Total delay for adaptive control systems for Case 2

Cross Street Demand: Low (700 vphpl)												
Main Street Traffic (vphpl)	Phase 1				Phase 2				Total Delay			
	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed
	Vehicle Hours				Vehicle Hours				Total Delay			
700	3.46	4.74	2.42	3.92	2.86	5.23	2.411	3.55	6.32	9.96	4.831	7.47
800	3.84	6.33	3.856	5.16	4.49	7.04	3.98	5.19	8.33	13.37	7.836	10.35
900	6.25	8.41	3.679	6.59	5.97	7.39	4.32	5.94	12.22	15.7	7.999	12.53

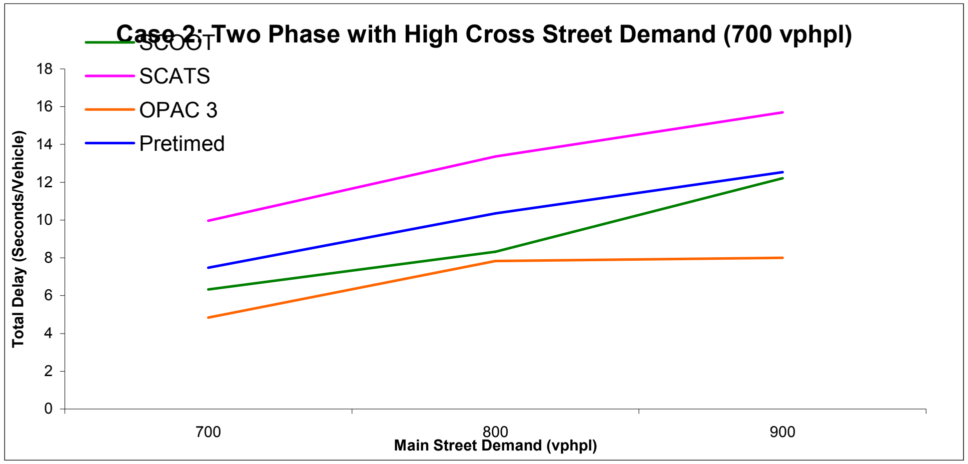
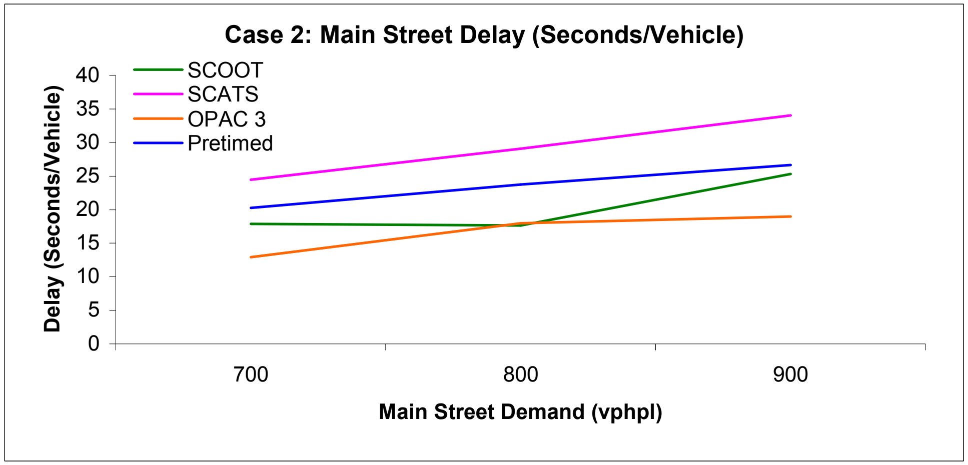


Figure 47 Total delay for adaptive control systems for Case 2

Table 35 Average Delay (seconds/vehicle) for Adaptive Control Systems for Case 2

Cross street demand: High (700 vphpl)								
Main Street Traffic	Phase 1				Phase 2			
	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed
(vphpl)	seconds/vehicle				seconds/vehicle			
700	17.89	24.48	12.928	20.26	15.15	27.69	12.122	18.82
800	17.66	29.1	17.96	23.75	23.09	36.17	19.286	26.64
900	25.33	34.04	14.983	26.67	31.77	39.25	22.51	31.54



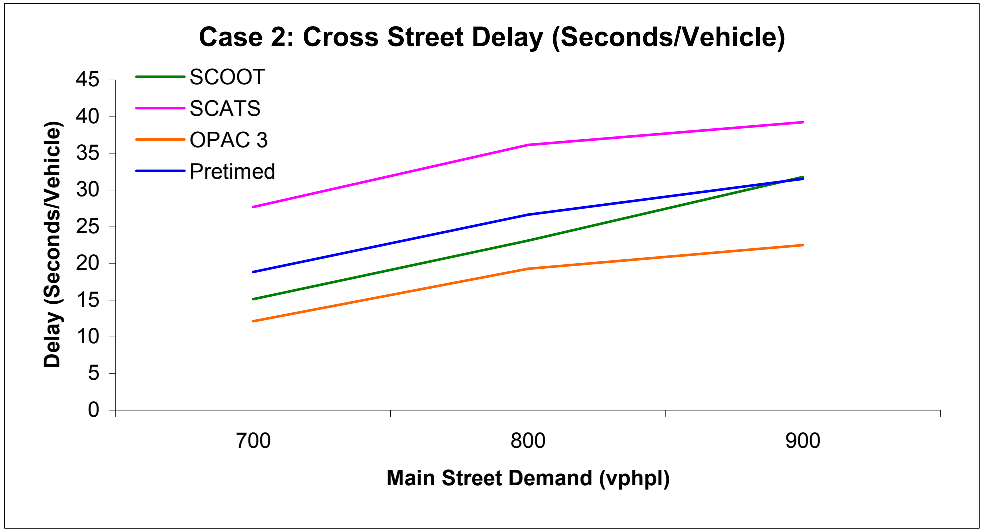


Figure 48 Average delay (seconds/vehicle) for adaptive control system for Case 2

Case 3: -

Table 36 Total delay for adaptive control systems for Case 3

Cross street demand: Phase 2: Low (200 vphpl), Phase 3: Low (200 vphpl)																
Main	**Phase 1**				**Phase 2**				**Phase 3**				**Total Delay**			
Street Traffic	**SCOOT**	**SCATS**	**OPAC 3**	**Pre-timed**	**SCOOT**	**SCATS**	**OPAC 3**	**Pre-timed**	**SCOOT**	**SCATS**	**OPAC 3**	**Pre-timed**	**SCOOT**	**SCATS**	**OPAC 3**	**Pre-timed**
(vphpl)	**Vehicle Hours**				**Vehicle Hours**				**Vehicle Hours**				**Total Delay**			
700	2.94	2.86	1.786	4.55	1.04	1.97	1.481	1.67	1.15	1.83	1.437	1.55	5.12	6.67	4.704	7.77
800	2.35	3.61	2.116	5.16	1.64	2.57	1.815	1.84	1.71	2.22	1.543	1.75	5.7	8.4	5.474	8.75
900	3.41	4.66	2.243	7.39	1.66	2.32	1.993	1.67	1.68	2.47	1.862	1.87	6.73	9.45	6.098	10.93
1000	3.9	4.58	2.652	6.39	2.09	2.71	2.611	2.4	2.55	3.21	2.309	2.58	8.55	10.51	7.572	11.37
1100	5.36	5.57	4.125	7.71	3.18	3.74	2.691	2.77	3	3.79	2.788	2.96	11.54	13.3	9.604	13.44
1200	6.02	7.21	6.68	6.67	4.5	3.94	3.386	4.54	4.53	4.81	3.501	4.98	15.07	15.97	13.567	16.2

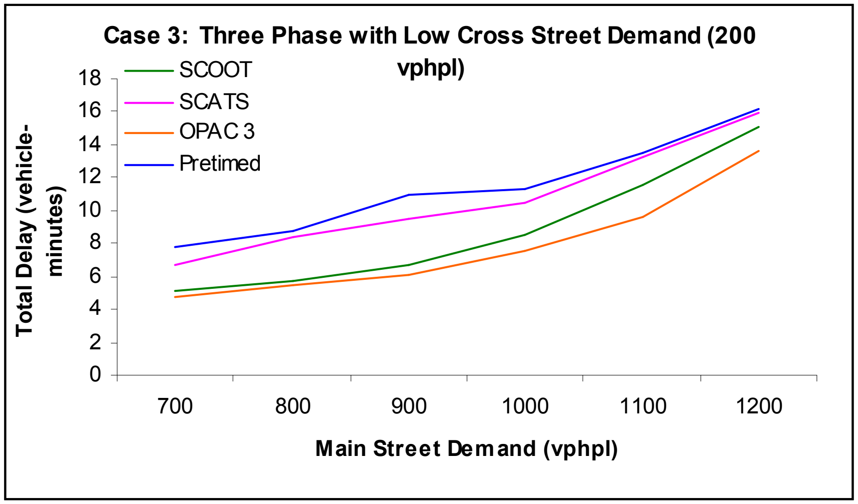
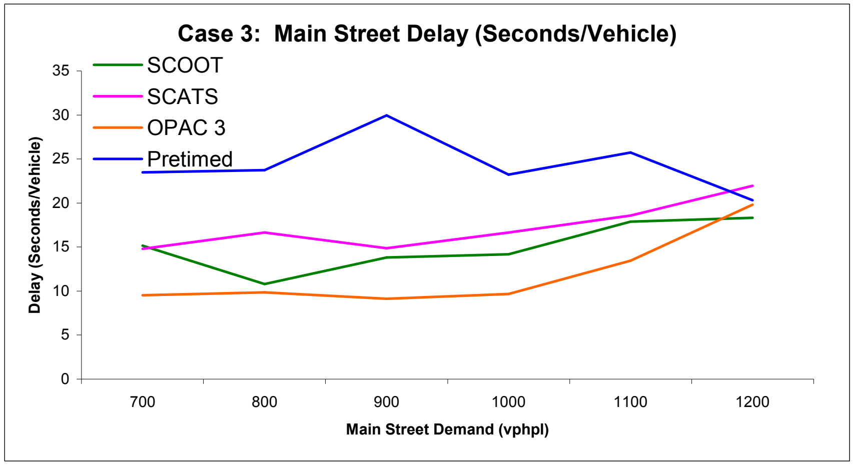


Figure 49 Total delay for adaptive control systems for Case 3

Table 37 Average delay (seconds/vehicle) for adaptive control systems for Case 3

Cross street demand: Phase 2: Low (200 vphpl), Phase 3: Low (200 vphpl)												
Main Street Traffic	Phase 1				Phase 2				Phase 3			
	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed	SCOOT	SCATS	OPAC 3	Pre-timed
(vphpl)	seconds/vehicle											
700	15.16	14.79	9.511	23.48	19.05	36.29	27.5	30.76	22.4	35.73	24.41	30.24
800	10.81	16.63	9.857	23.75	28.69	44.97	30.825	32.09	31.96	41.22	29.236	32.68
900	13.8	14.86	9.134	29.94	31.58	61.68	36.794	31.82	30.32	73.12	35.664	34.01
1000	14.18	16.66	9.656	23.24	38.96	50.68	45.41	44.9	47.59	59.66	41.575	47.84
1100	17.89	18.57	13.438	25.73	57.78	67.99	52.663	50.38	52.17	69.04	49.702	51.53
1200	18.33	21.96	19.826	20.32	83.62	73.22	61.565	84.41	84.21	89.37	60.893	92.42



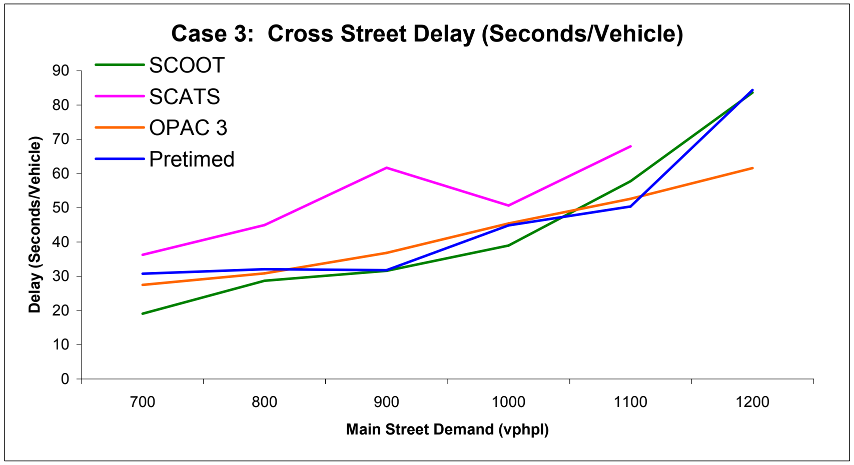


Figure 50 Average delay (seconds/vehicle) for adaptive control systems for Case 3

MICROSCOPIC SIMULATION OF SCOOT, SCATS AND OPAC USING PARAMICS

Introduction

Microscopic simulations for SCOOT, SCATS, and OPAC were performed in Paramics Project Suite. The main goal was to evaluate the performance of adaptive control algorithms using realistic and well-calibrated microscopic simulations. As mentioned earlier, macroscopic simulations were carried out under a highly controlled environment, where parameters such as vehicle arrivals and departures, as well as, green time for a signal, were known without any uncertainty that stemming from the stochastic nature of the traffic and were thus easily controlled. Macroscopic simulations gave a preliminary idea of what to expect from adaptive control strategies in terms of their performance under different traffic conditions. It was, however, necessary to test these systems in a highly stochastic environment such as the Paramics simulation program. The results obtained were also used for developing the rules of the rule base of the knowledge-based expert system developed as a part of this project.

The Paramics Project Suite consists of Paramics Modeller, Paramics Processor, and Paramics Analyser. The Paramics Programmer consists of a functional interface, or API, and a data interface. The users are allowed additional modules referred to as “plugins.” API functions can be subdivided into two main groups, namely control function and callback functions. Control functions include two types of control, override control functions and overload control functions. Paramics provides a default standard override control simulation model. If the user defines an override control function in a plugin, the newly defined override control function will replace the internal override function. The overload control functions are defined in the standard simulation loops, which are accessible to the user. Users can add more codes to the Paramics simulation loop. An overloaded control function can be defined in more than one plugin, and Paramics will call each of them in turn.

Paramics and API interact to run simulations. Some input files should be customized to run SCOOT-Like, SCATS-Like and OPAC-Like algorithms. Formats of these files are explained in Appendix B titled “Paramics API for Adaptive Control”. The APIs are capable of generating timing plans according to the specific adaptive control algorithm and can override the default Paramics settings. Figure 51 shows the data exchange while using an API.

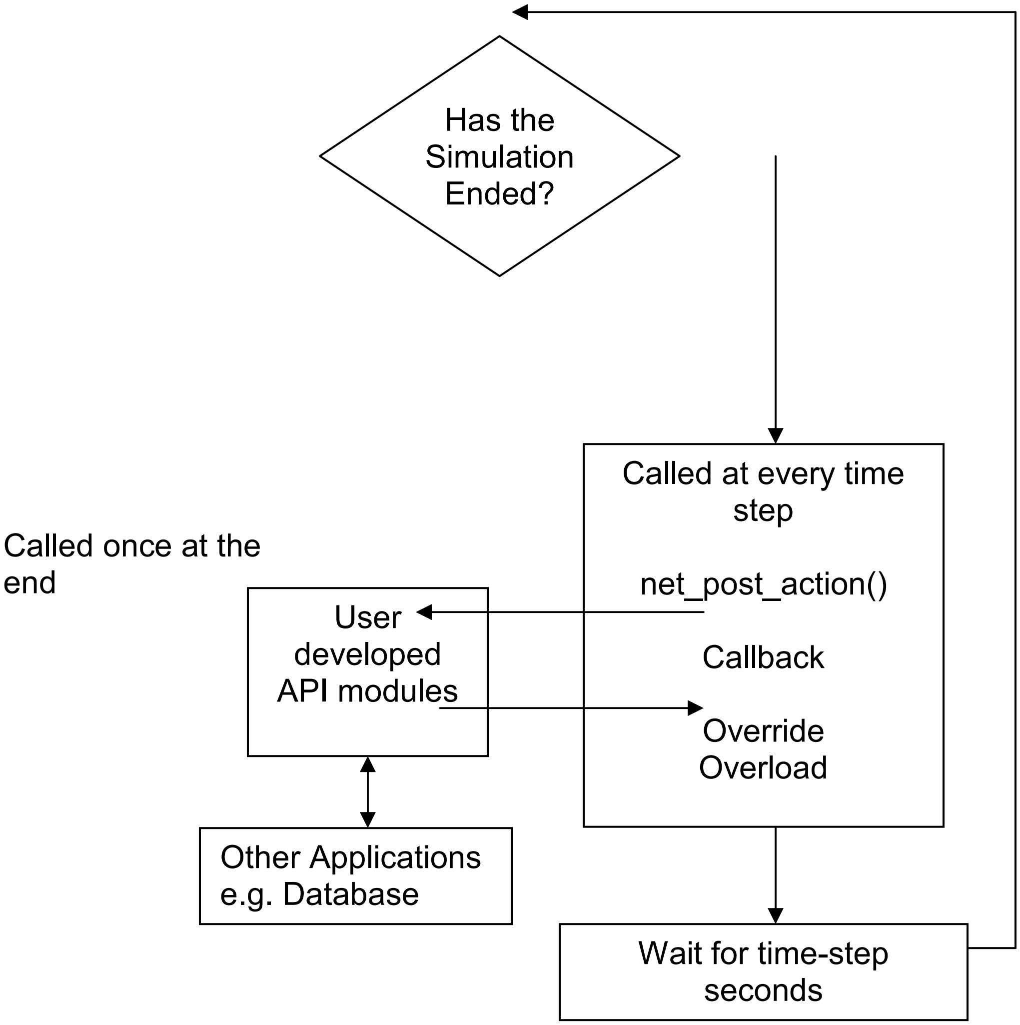


Figure 51 Data Exchange Diagram (65)

Development of an Object Oriented Modeling Approach for API Programming

The algorithms for OPAC, SCOOT and SCATS were not readily available from their developers. Thus, the research team had to develop these algorithms from scratch using the information available in the open literature. The main difficulty in such an endeavor, beyond the development of the algorithm for the specific signal control approach, is in its integration with the microscopic simulation tool. For the integration, the research team had to use Paramics API functionality. One of the main drawbacks that have prevented widespread use of Paramics API is the steep learning curve involved. This can be traced to fact that development is performed in C language, a procedural programming language. In the programming language community, it is well known that in a procedural programming language, data and program are interspersed. Operations that modify a data item may be spread over the entire code. This creates dependencies among several portions of the program. These dependencies lead to three drawbacks:

- **Difficult to understand:** Understanding a function may require knowledge of all the dependent functions.
- **Difficult to modify:** Dependence between various functions in the code means that changes may not be localized to a certain part, instead being spread over multiple parts of the program.
- **Difficult to re-use**: Re-using a specific function in a different code is not straightforward because of the dependencies.

To overcome these difficulties in developing an API, a simpler programming environment called **EZParamics** (60) is developed using the object oriented (OO) programming concept. The aim was to make programming in Paramics easier without any reduction in flexibility and capability. OO programming is fundamentally different from procedural programming in that it based on the notion of *objects*. An object in OO programming encapsulates both the data and the operations that manipulate the data (the code). Conceptually, an object in OO programming is analogous to a real-world object having a state and behavior. For example a vehicle in a traffic network has the following state: its current speed, identifier of link it is currently on, etc. Using the brake and gas pedal, a driver can accelerate/decelerate the vehicle. These are examples of the behavior of the brake and gas pedal. Since traffic simulation involves interaction among multiple objects (vehicles, signals, detectors, etc), it lends itself naturally to development using an OO programming language.

EZParamics imparts an object-oriented look and feel to Paramics. This is a simple but significant change to Paramics because OO programming promotes re-use of the existing code, and makes understanding and modifying the code easier. An API for SCOOT, SCATS and OPAC were made using the EZParamics program structure. More information about EZParamics is available in (60).

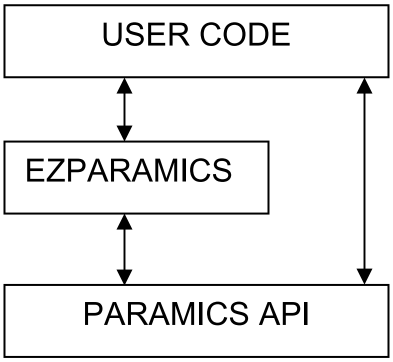


Figure 52 Schematic diagrams illustrating the structure of EZParamics

OPAC Simulations Using Paramics

The OPAC-Like 3 strategy was implemented using the Paramics microscopic simulation model. An API was written for the signal control using the EZParamics structure. This was different than the previous program written in C because data were to be extracted from detectors as in real time. Detector placement was also important, and the final placement of detectors was based on several experiments and as per the requirements of OPAC. To determine the arrivals for the head period of the horizon, count detectors were placed at 5, 10, and 15 seconds upstream of the stop line. To obtain future arrivals for the head period, the count of the upstream detector was subtracted from the count of the downstream detector. For the tail period, arrivals were generated using exponential inter-arrival times.

The detector at the stop line, which was used to find the queue length, needed to be selected properly. The following trials were run to obtain the most accurate result.

Trial 1: On stop line

Advantages:	1. Simple to get queue counts from API file
Disadvantages:	1. Vehicles stopping at the stop line made an increment in the loop count, sometimes giving wrong queue value when vehicle counts from upstream detectors were subtracted.

Trial 2: On downstream link near the node

Advantages:	1. Helps in calculating the correct queue length.
Disadvantages:	1. Paramics API will be more complicated since detectors will be placed on different links. 2. Right turn and left turn vehicles will not be counted hence, it will give a wrong queue value.

Trial 3: Few feet downstream of the stop line such that turning vehicles do not add up to counts of other links

Advantages:	1. Simple to get queue counts from API file. 2. Helps in calculating correct queue length
Disadvantages:	1. Method might not work in case of intersections having less distance between stop lines and point of curvature of curb.

Finally, the stop-line detectors were placed a few feet downstream of the stop line.

Figure 53 shows the test network layout as well as with detector placement along the link, for OPAC requirements in Paramics. The detector placement for stop-line detector is as shown in Figure 54.

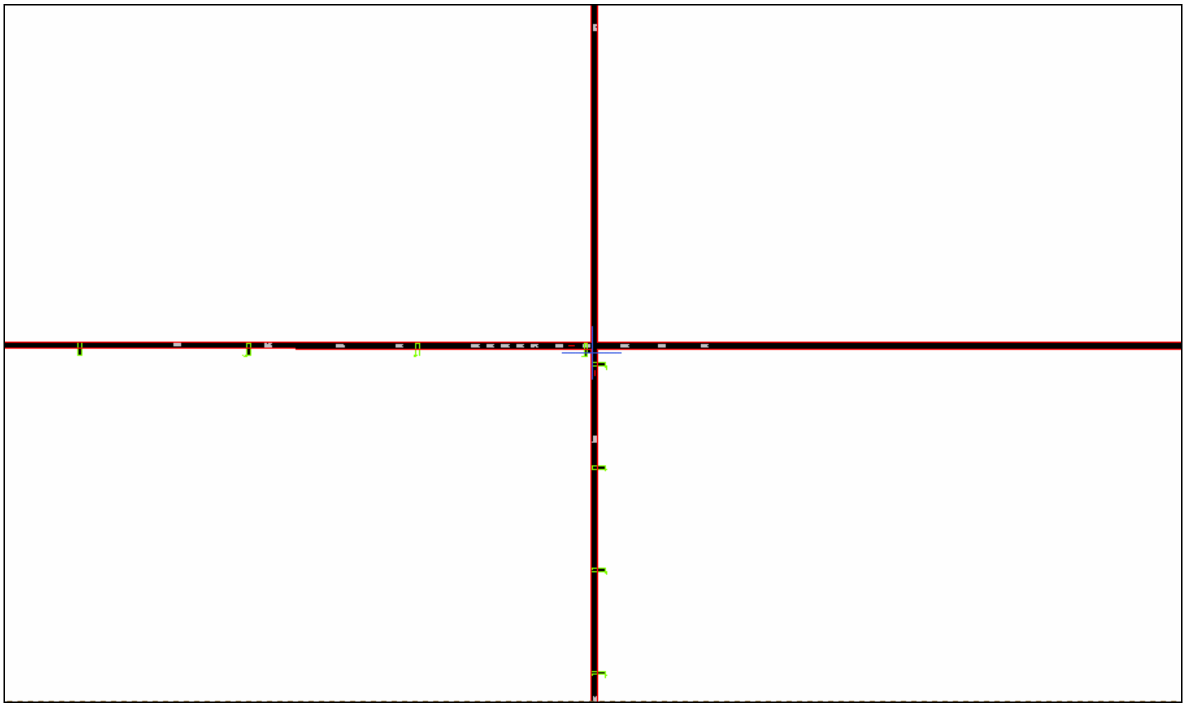


Figure 53 Overview of test network Paramics with detector placements

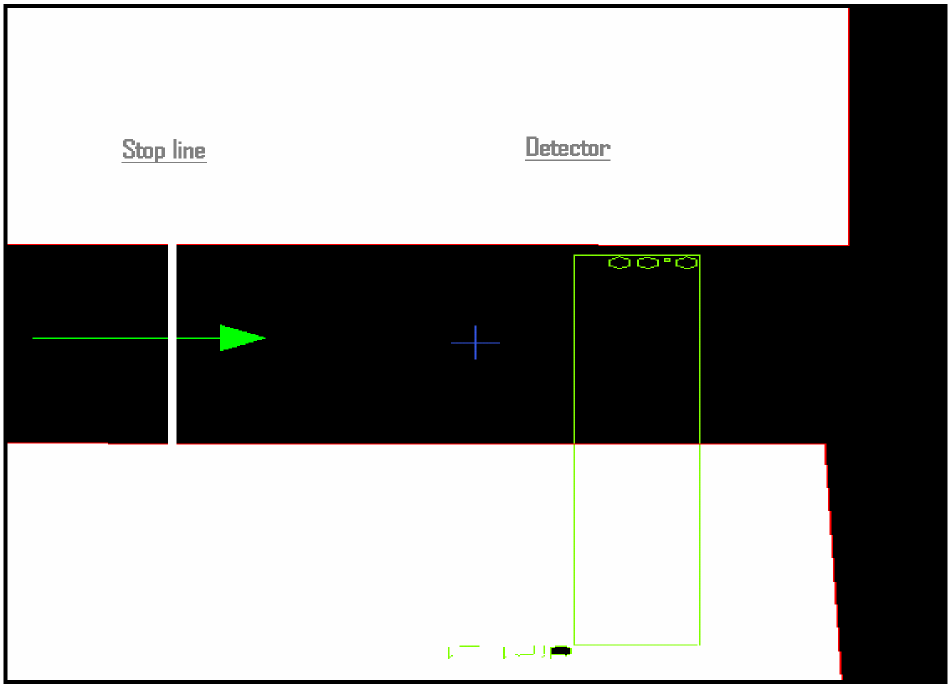


Figure 54 Detector placements near stop line

Simulations were run for the Figure 54 network in Paramics using the OPAC-Like 3 algorithm. Maximum green time for each phase was set to be approximately 2.5 times the optimized green time. Inter-arrival times for the tail period were considered to be exponential. For the head period, future arrivals were taken by subtracting two successive detector counts. Initial simulation results were not encouraging, owing to abnormal results and no comparison with C simulation results. The results for low cross street traffic and the two-phase scenario are shown in Table 38 and Figure 55 through Figure 57 below.

Because Paramics simulations with different seed values give different results, it is essential to use a suitable output analysis method to determine the number of replications required for the simulation of each intersection. The number of replications is determined using the sequential method. (56) This statistical procedure aims to obtain the mean $\mu = E(X)$ of the selected performance measure X, within a specified precision.

If $\overline{X}$ is estimated such that $\left|\overline{X} - \mu\right| / |\mu| = \gamma$, then γ is called the relative error of $\overline{X}$. The specific objective of this procedure is to obtain an estimate of μ with a relative error of γ and a confidence level of $100(1-\alpha)$ %. If the half-length of the confidence interval denoted by $\delta(n,\alpha)$, then the sequential procedure is as follows:

1. Make n_0 replications of the simulation and set $n = n_0$.
2. Compute $\overline{X}(n)$ and $\delta(n,\alpha)$ from $X_1, X_2, ..., X_n$.
3. If $\delta(n,\alpha)/\left|\overline{X}(n)\right| \leq \gamma'$, use $\overline{X}(n)$ as the point estimate for μ and stop. If not, replace n by $n+1$, make an additional replication of the simulation and go to step 1.

where, $\gamma' = \gamma/(1-\gamma)$.

An excel spreadsheet was used to implement this method. Simulations were stopped as soon as the relative error was within 10%. Simulations were also stopped when the error in the mean was within acceptable limits.

Table 38 Initial OPAC simulation results using Paramics (Low cross street traffic and two phases)

Cross street demand: Low (200 vphpl)						
Main Street Traffic	**Phase 1**		**Phase 2**		**Total Delay**	
	OPAC 3 (Paramics)	**Pre-timed**	**OPAC 3 (Paramics)**	**Pre-timed**	**OPAC 3 (Paramics)**	**Pre-timed**
(vphpl)	**seconds/vehicle**		**seconds/vehicle**		**seconds/vehicle**	
700.00	0.02	0.79	16.45	5.84	3.68	1.91
800.00	0.30	0.86	12.73	8.42	2.79	2.37
900.00	0.00	0.90	21.69	11.22	3.94	2.77
1000.00	0.40	0.90	31.33	13.53	5.56	3.01
1100.00	0.45	0.92	38.77	18.61	6.35	3.64
1200.00	0.76	0.96	36.87	24.55	5.92	4.33

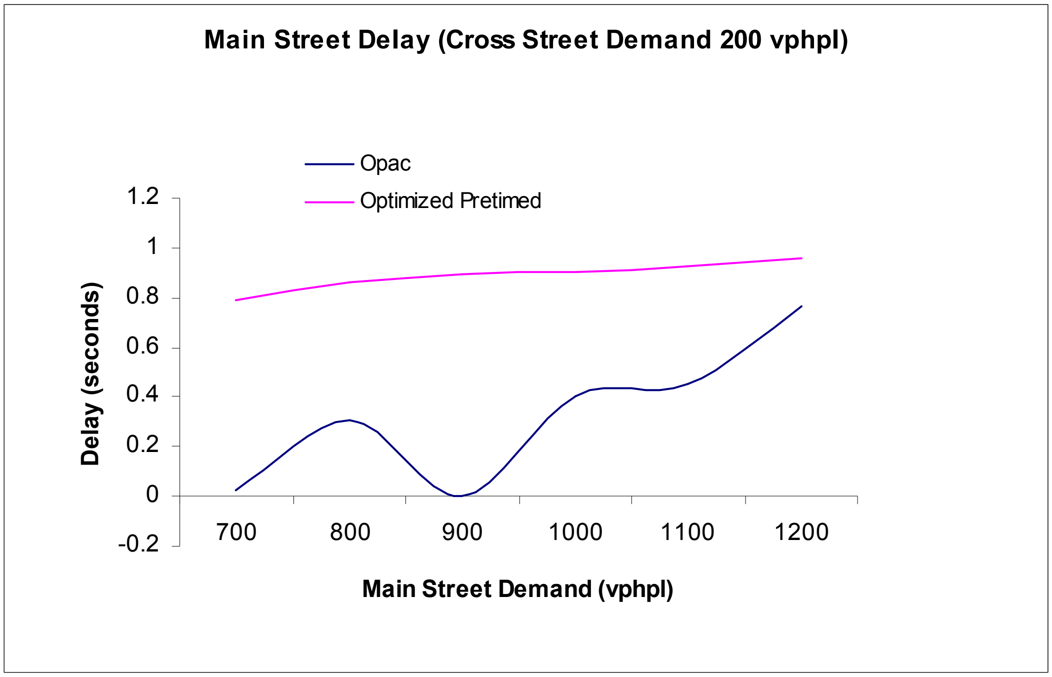


Figure 55 Main street delay for initial OPAC-Like 3 simulations in Paramics

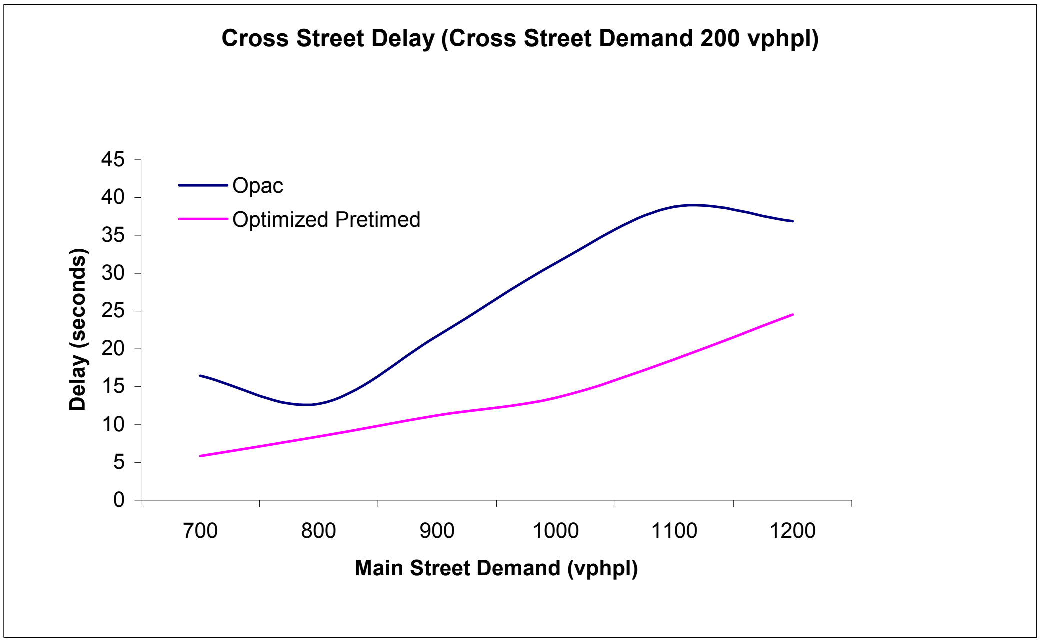


Figure 56 Cross street delay for initial OPAC-Like 3 simulations in Paramics

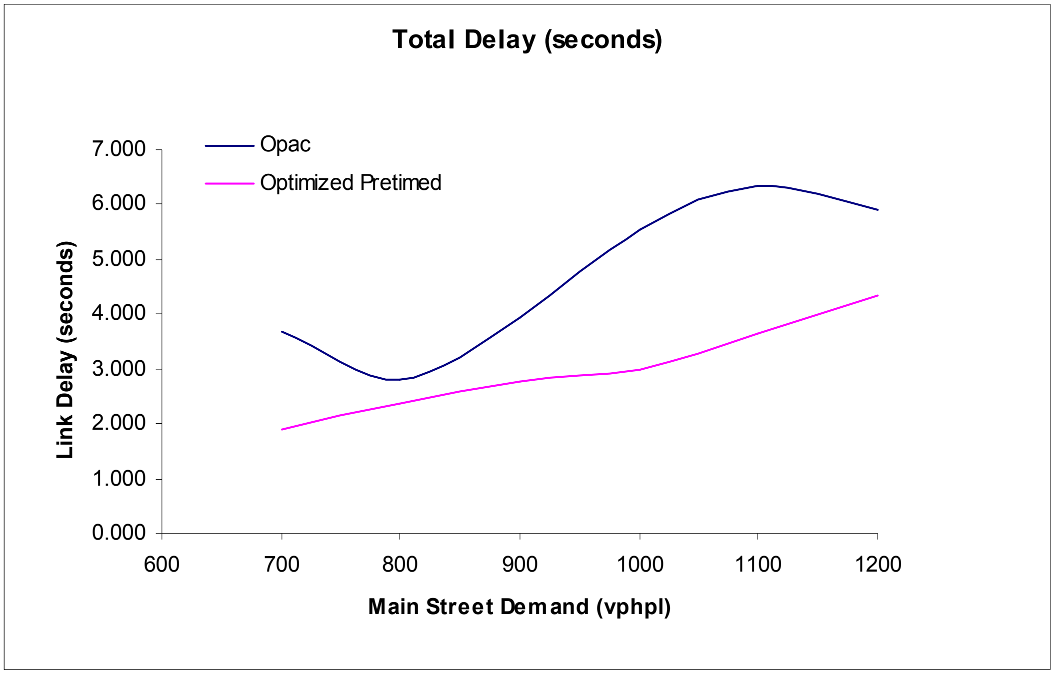


Figure 57 Total delays for initial OPAC-Like 3 simulations in Paramics

According to Figure 55 through Figure 57 the behavior of OPAC-Like 3 algorithm under real-time conditions simulated by Paramics is unusual. More simulation runs were conducted to see the contributing factors that lead to higher than expected delays when Paramics results are compared with the results of macroscopic C program simulations. These factors are:

1. Sensitivity analysis for maximum and minimum green times,
2. Effect of known and unknown future arrivals,
3. Effect of departure rates.

Sensitivity Analysis for Maximum and Minimum Green Times

To determine the actual minimum and maximum green-time settings that will give lower delays compared with pre-timed signal control under real- time conditions, minimum and maximum green times were varied for the test network. The main street demand was kept at 1000 vphpl, and the cross street demand was kept at 200 vphpl. The optimized pre-timed timing plans were generated using Synchro. The minimum and maximum green times were varied as shown in Table 39. Again the sequential method described in the previous section was used to determine the number of replications needed.

Table 39 Effect of maximum and minimum green time on OPAC implemented in Paramics

Main Street Traffic: 1000 vphpl				
Cross Street Traffic: 200 vphpl				
Optimized Pre-timed		**Main Street**	**Cross Street**	**Total Delay**
46	16	0.901428571	13.53	3.006190476
Opac Timings		**Main Street**	**Cross Street**	**Total Delay**
45	15	0.053	13.347	**2.268**
50	20	1.550	12.920	3.445
	25	3.198	12.750	4.790
	30	4.223	12.310	5.571
60	20	0.735	19.050	3.788
	25	2.408	17.180	4.870
	30	3.850	15.465	5.786
70	20	0.268	22.736	4.012
	25	1.823	20.410	4.920
	30	3.448	19.568	6.135
80	20	0.016	25.790	4.312
	25	1.288	22.720	4.860
	30	3.300	22.330	6.472
90	20	0.021	28.895	4.833
	25	0.848	27.130	5.228
	30	2.440	24.470	6.112

According to these results in Paramics, OPAC-Like 3 gives a minimum delay when the maximum green time is close to the optimized green time. The same results were obtained with other demand conditions.

Effect of Known and Unknown Future Arrivals

To understand the effect of known and unknown future arrivals, two sets of simulations were run using Paramics. The link length was kept small so that other stochastic effects were negligible. The seed value was kept the same in both runs. In the first run, arrivals were recorded on detectors using a separate API. These arrivals were stored in a text file and used in the next run (i.e., OPAC simulation). Hence, instead of using exponential inter-arrival times, arrivals were read directly from a separate text file. Because the seed value was kept the same, the pattern of arrivals did not change in two simulations. The unknown arrival scenario was also created in Macroscopic Intersection Simulation Program (MISP), and results were compared as shown in Table 40 and Table 41:

Table 40 OPAC-Like 3 simulations with known and unknown arrivals in Paramics

Main Street Demand	**1000 vphpl**
Cross Street Demand	**200 vphpl**

OPAC-like 3 Simulations with Randomness Using Paramics (Using single seed 2963)						
	With Known Arrivals (Short Link)			**With Unknown Arrivals (Short Link)**		
	Main Street	**Cross Street**	**Total Delay**	**Main Street**	**Cross Street**	**Total Delay**
Pre-timed	5.2	19.9	**7.7**	5.20	19.90	**7.65**
45/15	3.0	18.7	**5.6**	4.40	13.00	**5.83**
50/15	3.3	22.0	**6.4**	4.60	16.40	**6.57**
50/20	5.7	17.3	**7.6**	6.70	15.40	8.15
50/25	6.8	17.8	8.6	8.60	13.40	9.40
50/30	9.7	18.3	11.1	10.00	18.90	11.48
60/20	5.0	29.3	9.1	5.30	25.50	8.67
60/25	6.2	25.9	9.5	7.00	22.50	9.58
60/30	6.7	23.1	9.4	8.40	22.10	10.68
70/20	4.5	28.4	8.5	6.40	17.50	8.25
70/25	4.9	28.5	8.8	5.70	31.00	9.92
70/30	7.7	25.4	10.7	8.30	24.10	10.93
80/20	3.7	32.6	8.5	5.90	17.60	7.85
80/25	4.7	32.5	9.3	7.90	16.90	9.40
80/30	5.8	34.4	10.6	10.30	13.50	10.83
90/20	2.5	43.8	9.4	3.30	41.10	9.60
90/25	5.4	37.6	10.8	7.60	24.60	10.43
90/30	6.2	35.8	11.1	9.70	20.20	11.45
110/25	4.7	44.3	11.3	5.80	29.50	9.75
*Delays are in seconds/vehicle						

Table 41 OPAC-Like 3 simulations with known and unknown arrivals using MISP

OPAC Simulations with Randomness Using MISP						
	With Known Arrivals			**With Unknown Arrivals**		
	Main Street	**Cross Street**	**Total Delay**	**Main Street**	**Cross Street**	**Total Delay**
Pre-timed	12.029	22.028	**13.695**	12.029	22.028	**13.70**
45/15	5.848	8.550	**6.298**	6.15	9.25	**6.67**
50/15	6.014	7.971	**6.341**	6.37	8.77	**6.77**
50/20	6.135	8.140	**6.469**	6.73	8.44	**7.01**
50/25	7.487	6.884	**7.387**	7.98	7.36	**7.88**
50/30	7.729	6.473	**7.520**	8.00	7.02	**7.84**
60/20	6.400	7.681	**6.614**	6.90	8.10	**7.10**
60/25	7.198	6.884	**7.146**	7.79	7.36	**7.72**
60/30	7.439	6.473	**7.278**	7.86	6.90	**7.70**
70/20	5.893	8.212	**6.280**	6.49	8.49	**6.82**
70/25	7.246	6.908	**7.190**	7.79	7.33	**7.71**
70/30	7.439	6.497	**7.282**	7.86	6.92	**7.70**
80/20	5.893	8.236	**6.284**	6.49	8.51	**6.83**
80/25	7.246	6.908	**7.190**	7.79	7.33	**7.71**

80/30	7.439	6.497	**7.282**	7.86	6.93	**7.71**
90/20	5.893	8.236	**6.284**	6.49	8.51	**6.83**
90/25	7.246	6.908	**7.190**	7.79	7.33	**7.71**
90/30	7.439	6.497	**7.282**	7.86	6.93	**7.71**
110/25	5.893	8.236	**6.284**	7.79	7.33	**6.83**

*Delays are in seconds/vehicle

According to these experimental results, when departures and queues can be estimated accurately, knowing the arrivals gives lower delays in OPAC. However, in certain cases, delay with unknown arrivals in Paramics gives lower delays compared with cases in which arrivals are known.

According to the MISP simulation results, knowing arrivals gives lower delays compared with to the case when they were unknown. Similar delay values are also observed for different values of maximum green time. This supports the statement made earlier that increasing maximum green time beyond a certain limit does not necessarily produce better results.

Conclusions from Paramics Simulation Runs

1) The OPAC-Like 3 algorithm gives lower delays when the maximum green time is equals to the optimized green time mainly because of the 5 seconds time step being considered in the optimization procedure. Under real-time conditions, where the actual departure rates are not known, the queue may be dissipated earlier than expected. In such cases, it takes 5 seconds for the OPAC-Like algorithm to realize this error. Then it may decide to terminate the phase earlier than previously planned. Keeping the maximum green time equal to the optimized pre-timed case would terminate the phase at the “right” moment. Theoretically, it may be possible to achieve lower delay if the maximum green time is greater than the optimized green time; keeping the maximum green time equal to the optimized green time would reduce the number of wrong decisions and thus reduce delays.
2) Non-optimal detector location, calibration, and reliability might also lead to higher delays under OPAC. If detectors are not producing proper counts (this happens in Paramics where detectors sometimes miss vehicles), the signal switches based on some random numbers and not based on actual traffic demands. One productive way to overcome the detector miscount problem in Paramics is to reset every detector count to zero after a finite time interval so that the error is not propagated in time. However, Paramics does not allow the detector count to reset during the individual simulation runs.
3) Estimation of the queue lengths in Paramics was also not always very accurate. For higher demand levels, it is always difficult to estimate the queue length precisely. Subtracting two detector counts (sometimes erroneous) from the detector near the stop line and the upstream detector

may not always lead to a correct queue estimate, especially when the green time for a phase is large.

4) The effect of known and unknown arrivals is not clear based on the analysis of Paramics results. However, MISP results indicate that knowing arrivals in advance reduces delays. Regarding Paramics results, many factors such as departure rates, queue estimates, and detector errors that are not correctly estimated may lead to higher delays even though arrivals are known in advance. However, from Table 39 indicates that knowing arrivals in advance produces lower delays compared with optimized pre-timed arrivals for a larger number of cases with different green time selections. This result supports the fact that knowledge of arrivals is advantageous.
5) Additional simulation runs were conducted to see the effect of unknown arrivals and wrong decisions caused by unknown parameters in Paramics (Table 42). No difference is observed in the total delay for the known arrival and unknown arrival cases when the traffic demands in the main street and cross street were medium-high and medium-medium. However, when the traffic on the main street was very high compared to cross street, a wrong decision would cause more vehicles to wait on the main street, leading to a higher total delay. Hence, in cases where the main street traffic is considerably higher than cross street traffic, accurate prediction of arrivals is proven to be helpful.
6) Because departure rates cannot be determined in Paramics, simulation runs to study its effect were not conducted. To reduce the impact of this effect, departures rates used in horizon calculations were derived form the saturation flow rates. However, this approach may cause under-performance of OPAC. To overcome this problem, the time interval (which was considered as 5 seconds) can be reduced (e.g. to 1 second), however, this will in turn increase the computational effort because the number of time steps in the horizon will increase by a multiple of five.

Table 42 Paramics simulations with different demand levels to investigate the effect of known and unknown Arrivals

Main Street Demand	**1500 vphpl**
Cross Street Demand	**100 vphpl**

OPAC Simulations with Randomness Using Paramics (Using single seed 2963)						
	with Known Arrivals (Short Link)			**with Unknown Arrivals (Short Link)**		
	Main Street	**Cross Street**	**Total Delay**	**Main Street**	**Cross Street**	**Total Delay**
Pre-timed	4.3	55.4	**7.5**	4.3	55.4	**7.5**
125/15	2.1	58.3	**5.6**	3.40	56.60	**6.7**

Main Street Demand	**1500 vphpl**
Cross Street Demand	**500 vphpl**

OPAC Simulations with Randomness Using Paramics (Using single seed 2963)						
	With Known Arrivals (Short Link)			**With Unknown Arrivals (Short Link)**		
	Main Street	**Cross Street**	**Total Delay**	**Main Street**	**Cross Street**	**Total Delay**
Pre-timed	11.4	52.7	**14.0**	11.4	52.7	**14.0**
90/35	9.7	77.5	**13.9**	9.70	77.50	**13.9**

Main Street Demand	**500 vphpl**
Cross Street Demand	**500 vphpl**

OPAC Simulations with Randomness Using Paramics (Using single seed 2963)						
	With Known Arrivals (Short Link)			**With Unknown Arrivals (Short Link)**		
	Main Street	**Cross Street**	**Total Delay**	**Main Street**	**Cross Street**	**Total Delay**
Pre-timed	7.8	7.6	**7.8**	7.8	7.6	**7.8**
20/20	6.4	7.2	**6.5**	6.40	7.20	**6.5**
*Delays are in seconds/vehicle						

Simulation of SCOOT-Like and SCATS-Like Adaptive Control Strategies Using Paramics

Implementation of the APIs of SCOOT-like and SCATS-like strategies were developed using EZParamics. The flow charts for these APIs are the same as the ones presented in the previous section where macroscopic simulation implementations are described. The results from initial simulations of SCOOT and SCATS were found satisfactory. Unlike OPAC, these strategies do not require arrival and departure estimates. Hence experiments conducted for the OPAC-like strategy were not repeated for SCOOT-like and SCATS-like strategies. All three strategies were then tested on NJ Highways, generating a database for an expert system would help in acquiring the knowledge about the performance of these strategies under given network and demand conditions.

Required Input Data for SCOOT-like API

Three files are prepared for SCOOT-like API. One is the “priorities” file, provided by Paramics, to define the movement for all phases and initial phase timing. The “priorities” file could be modified and generated by the graphical display of priority panel in the Edit Junction window and also could be modified by typing text into the file. The other two files are “intersection_info“ file and “signal_control_parameters”, which are provided by the user and store intersection information and signal control parameters.

The “intersection_info” File

The “intersection_info” file contains all the related information about the intersection that is modeled. An example shown in Figure 58 illustrates the format of the file. Each record contains information for individual lanes in at the intersection. The example intersection shown in the figure 13 helps to explain this file. Node name in this intersection is 6. Movements on lanes 1 and 2 are controlled by phase 1. The name of the detector on the eastbound is EAST. The saturation flow on the eastbound approach is 1615 vehicles per hour. The calculation of saturation flow is discussed in the following section. The loss time for this lane is assumed to be zero. This file should be put in the same directory as that of the road network files.

```
# Intersection    Phase   Lane    DetectorName      Saturated Flow    lose time
# node id
      6             1       1        EAST               1615              0
      6             1       1        WEST               1900              3

      6             1       2        EAST               1900              3
      6             1       2        WEST               1900              3

      6             2       1        WhippNOR           1615              3
      6             2       1        WhippSOU           1900              3

      6             2       2        WhippNOR           1900              3
      6             2       2        WhippSOU           1638              3
```

Figure 58 The “intersection_info” file

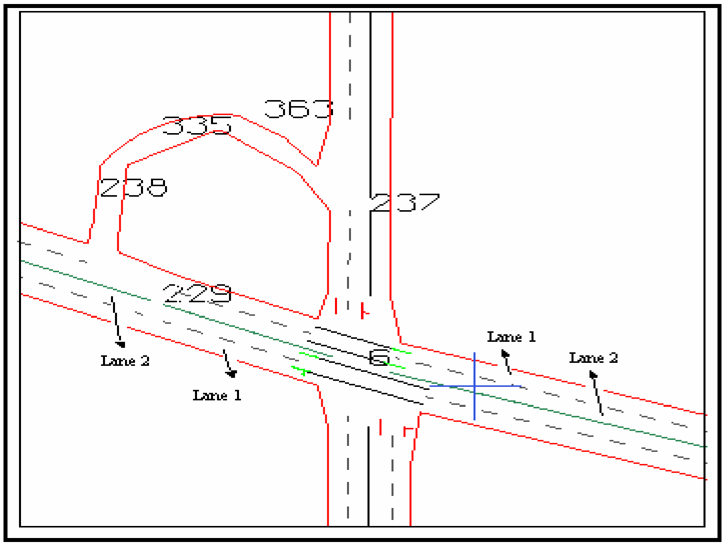


Figure 59 Graphical representation of the sample intersection frame

The "signal_control_parameters" File

The "signal_control_parameters" file includes all signal control parameters of the SCOOT-like algorithm.

```
# Defines various parameters that controls the behavior of the signal control module
#
DECISION_PHASE 5

CURRENT_GREEN_INCREMENT 4
NEXT_GREEN_INCREMENT 1

CURRENT_GREEN_DECREMENT 4
NEXT_GREEN_DECREMENT 1

MAX_DS_THRESHOLD 0.9
MIN_DS_THRESHOLD 0.8

# define the CYCLE_ADJUSTMENT_TIME in seconds. This is approximately the time
# by which cycle length should be adjusted
CYCLE_ADJUSTMENT_TIME 300

# define the maximum duration (secs) by which the cycle length should change
MAX_CYCLE_LENGTH_CHANGE 4

MAX_QUEUE_LENGTH_THRESHOLD 20
MIN_QUEUE_LENGTH_THRESHOLD 15
MODE_OF_OPERATION       SCOOTS
```

Figure 60 The "signal_control_parameters" file

These parameters are as follows:
"DECISION_PHASE" defines the number of phases in the intersection.

"CURRENT_GREEN_INCREMENT" is the increment of green time in optimizing the split.
"NEXT_GREEN_INCREMENT" is the decrement of green time in optimizing the split.
"MAX_DS_THRESHOLD" is the threshold of the maximum degree of saturation for increasing the green time or cycle length.
"MIN_DS_THRESHOLD" is the threshold of the minimum degree of saturation for decreasing the green or the cycle length.
"CYCLE_ADJUSTMENT_TIME" is a period after the cycle length is adjusted.
"MAX_CYCLE_LENGTH_CHANGE" is the amount of the cycle length change.
"MAX_QUEUE_LENGTH_THRESHOLD" is the threshold of the maximum queue length for increasing the green time.
"MIN_QUEUE_LENGTH_THRESHOLD" is the threshold of the maximum queue length for decreasing the green time.
"MODE_OF_OPERATION" is a switch between the SCOOT-like algorithm and SCATS-like algorithm.

The "priorities" file

A hierarchy of priorities exists in the order of MAJOR, MEDIUM, MINOR, and BARRED. MAJOR priority movements are free flow and not restricted by other streams of traffic. A MEDIUM priority yields to MAJOR streams of traffic but has priority over MINOR traffic movements. MINOR priority yields to both MAJOR and MEDIUM traffic flow and BARRED indicates the turn is banned to all vehicle movements.

```
actions 4                                  //node name
phase offset 0.00 sec
phase 1
16.00                          //default stored green time
min 10.00                     //minimal green time
max 126.00                    //maximal green time
red phase 4.00                        //yellow + all red
all barred except                     // the following defines priority of
movements
from 170 to 263 major
from 170 to 262 major
from 262 to 170 major
from 263 to 262 minor
phase 2
16.00
min 10.00
max 20.00
red phase 4.00
all barred except
from 170 to 263 medium
from 261 to 263 major
from 261 to 262 major
from 263 to 170 major
from 263 to 262 major
```

Figure 61 The "priorities" file

Required Input Data for SCATS API

Five files were prepared for SCATS API and stored in the same folder as network file. “priorities” and “intersection_info” files were the same as in the SCOOT-like algorithm API. The major difference between “signal_control_parameters” in the SCATS-like algorithm and SCOOT-like algorithm is the input item called MODE_OF_OPERATION. MODE_OF_OPERATION is an input to SCATS-like algorithm instead of the SCOOT-like algorithm. Two additional files, the “intersection_parameters” and “timing_plan” files are necessary for the SCATS-like algorithm API. These files are designed by the user, and define the intersection parameters and timing plan.

The “intersection_parameters” file

The following is an example of “intersection_parameters” file:

```
#NodeId  MAX_CYCLE_LENGTH_DECREASE    MAX_CYCLE_LENGTH_INCREASE
         MAX_FLOW_RATE_THRESH
4        -6                           6                             600
CMIN     CS       CX       CMAX     NOMINATED_PHASE
30       40       50       150      1
```

Figure 62 The “intersection_parameters” file

Where:
“NodeID” is the nodes ID in the network.
“MAX_CYCLE_LENGTH_DECREASE” is the lower bound of the cycle length decrement.
“MAX_CYCLE_LENGTH_INCREASE” is the upper bound of the cycle length increment.
“MAX_FLOW_RATE_THRESH” is the threshold of the maximum traffic flow below which the cycle length operates at a minimum cycle length.
“CMIN” is the minimum cycle length of this intersection.
“CS” is medium cycle length.
“CX” is threshold cycle length.
“CMAX” is the maximum cycle length of this intersection.
“NOMINATED_PHASE” defines a phase as a stretch phase.

The “timing_plan” file

The following is an example of “timing_plan” file:

```
#Nodeid  PLAN            Phase   TIME
4        1               1       50
4        1               2       50

4        2               1       65
4        2               2       35

4        3               1       77
4        3               2       23

4        4               1       88
4        4               2       12
```

Figure 63 The “timing_plan” file

Where:
“Nodeid” is the node ID in the road network.
“PLAN” is the plan number.
“Phase” is the phase number.
“Time” indicates the seconds allocated to the phase.

Calculation of the Saturation Flow

With the supply of input data, saturation flow can be calculated manually according to the Highway Capacity Manual or automatically by using Highway Capacity Software (HCS). The version of this HCS implementation procedures are defined in the 2000 Highway Capacity Manual (33).

The saturation flow rate is defined as a lane group, that can accommodate the maximum number of passenger cars in an hour, assuming that the green phase was always available to the lane group (i.e., that the green ratio, g/C, was 1.00). (24) Usually ideal saturation flow rate is 1900 passenger cars per hour of green time per lane (pcphgpl). The saturation flow rate is influenced by many factors such as lane width, heavy vehicles, grade of the road, right turn and left turn and so on. Therefore, the actual saturation flow rate is lower than the ideal saturation flow rate. A variety of adjustment factors are needed for the computation of the saturation flow rate. The following formula is given for the calculation of the saturation flow:

$$S=S_0 N f_w f_{HW} f_g f_p f_{bb} f_a f_{RT} f_{LT} \quad (18)$$ (33)

Where:
S= saturation flow rate for the subject lane group, expressed as a total for all lanes in the lane group under prevailing conditions (vphg)
S_0=ideal saturation flow rate per lane, usually 1900 (pc/hr green/ln)
f_w=adjustment factor for lane width
N=number of lanes in the lane group
f_{HV}= adjustment factor for heavy vehicles in the traffic stream.
f_g= adjustment factor for the approach grade
f_p= adjustment factor for the existence of a parking lane adjacent to the lane group and the parking activity in that lane
f_{bb}= adjustment factor for the blocking effect of local buses that stop within the intersection area
f_a= adjustment factor for the area type
f_{RT}= adjustment factor for right turns in the lane group
f_{LT}= adjustment factor for left turns in the lane group.

Assumptions for the network studied in this research:

- Average of lane width: 12 ft
- Approach grade: level
- No parking on the road
- No bus blocking
- No heavy vehicles
- No Central Business District (CBD) area

TESTING OF ADAPTIVE CONTROL ALGORITHMS ON NJ HIGHWAYS

After testing and debugging the adaptive control strategies implemented using three different API's written with EZParamics, these control algorithms were tested on various NJ highway intersections described in the previous sections of this report. A total of seven intersections from Routes 10, 18, and 23 were selected for testing. These intersections were selected based on different network features, traffic demand and number of phases in the traffic signal plan. The volume to capacity ratio on the main street was varied from 0.1 to 1.0 to test the behavior of these algorithms under various traffic conditions. Finally, plots showing percentage improvements for a predefined performance index were generated, comparing the results of all three adaptive control algorithms.

The results from the testing of these algorithms were used to generate a knowledge base for the final rule base system, which is explained in the next chapter. According to the simulation results of selected intersections, a knowledge base was prepared showing how much reduction in performance measures such as travel time, total delay, and stop time one can expect for a given network condition. The expert system first identifies the type of intersection based on its network parameters (e.g., saturation level and cross street demand), then looks into the knowledge base for a similar network, and finally gives an overview of how a selected adaptive control strategy would work for that intersection.

To compare the Paramics results for SCOOT, SCATS, and OPAC, it was necessary to define a performance index. Paramics generates an output file, which has a summary of statistics for travel time, stop time, etc. The file is named as "general" and is generated for every simulation run in the log folder for that run. The performance index (PI) used is as follows:

$$PI = 0.6 * travel\_time + 0.4 * stopped\_time \qquad (19)$$

The PI for each adaptive control algorithm was compared with the PI of the optimized pre-timed control traffic. (67)

The following notation is used in the simulation results:

Travel Time = mean travel time (seconds) in transit per vehicle for the network
Stop Time = mean stop time (seconds) per vehicle in the network
Performance Index (PI) = 0.6 * (Travel Time) + 0.4 * (Stop Time)

Intersection of Route 10 and Mt Pleasant Road

The intersection of Route 10 and Mt Pleasant has a jughandle turn from eastbound route 10 for vehicles to turn onto Mt. Pleasant road. Cross street traffic from Mt. Pleasant road and the jughandle is very low. The intersection signal runs on two phases.

Table 43 Simulation results of the intersection of Route 10 and Mt Pleasant Road

V/C Ratio	Pre-timed			SCOOT			SCATS			OPAC			% Improvement		
	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	SCOOT	SCATS	OPAC
0.1	80.6	0	48.36	77.9	0.1	46.78	77.4	1	46.84	79.1	0.1	47.5	3.3	3.1	1.8
0.2	85.4	2.1	52.08	82.5	0.5	49.7	81.1	0.8	48.98	85.3	0.5	51.38	4.6	6	1.3
0.3	87.3	5.1	54.42	84.2	2.1	51.36	82.2	0.6	49.56	86.4	12.6	56.88	5.6	8.9	-4.5
0.4	89.3	10.2	57.66	85.9	1.8	52.26	83.9	1	50.74	86.5	1.4	52.46	9.4	12	9.0
0.5	92.4	20.5	63.64	87	1.5	52.8	85.8	2	52.28	92.7	8.7	59.1	17	17.9	7.1
0.6	109.6	82.8	98.88	89.9	2.7	55.02	92	1.9	55.96	93.6	3.7	57.64	44.4	43.4	41.7
0.7	137.6	333.3	215.88	91.9	5	57.14	101.2	19.8	68.64	99	9.4	63.16	73.5	68.2	70.7
0.8	187.8	642.6	369.72	103	23.6	71.24	135.5	193.4	158.66	141.5	353	226.1	80.7	57.1	38.8
0.9	272.9	879.3	515.46	144.2	300.5	206.72	302.7	954.3	563.34	182.7	469.9	297.58	59.9	-9.3	42.3
1	335.6	1018	608.56	213.4	576.3	358.56	258.7	689.3	430.94	253.3	603.4	393.34	41.1	29.2	35.4

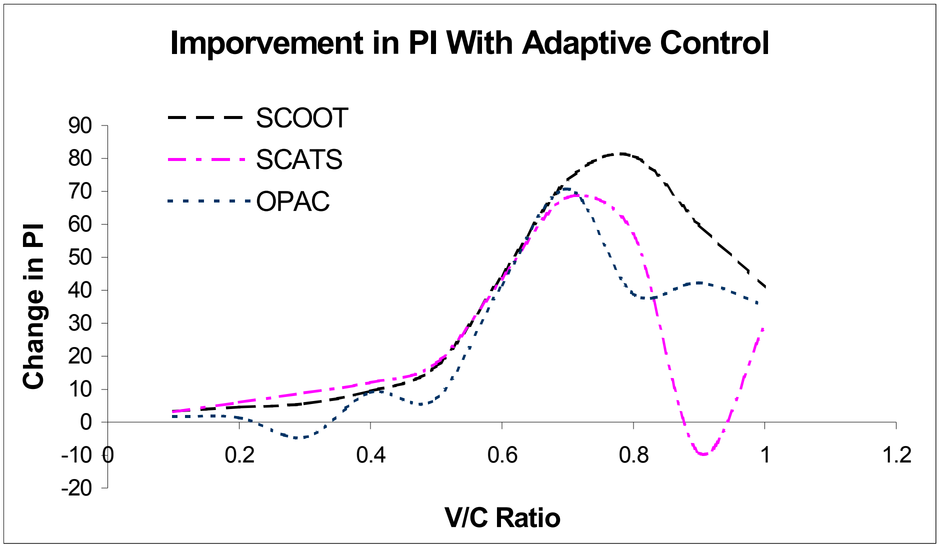


Figure 64 Percentage improvements in PI for adaptive control for the intersection of Route 10 and Mt Pleasant Road

Intersection of Route 10 and Whippany Road

The intersection of Whippany and Route 10 also has a jughandle turn from eastbound route 10 for vehicles to turn onto Whippany Road. Cross-street traffic from Mt Pleasant Road and the jughandle is high. The intersection signal runs on two phases.

Table 44 Simulation results for the intersection of Route 10 and Whippany Road

	Pre-timed			SCOOT			SCATS			OPAC			% Improvement		
V/C Ratio	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	SCOOT	SCATS	OPAC
0.1	63.2	4.1	39.56	61.8	2.8	38.2	62	2	38	63.3	3.7	39.46	3.4	3.9	0.3
0.2	64.8	5.1	40.92	62.6	2.5	38.56	63.4	4.3	39.76	65.7	8.3	42.74	5.8	2.8	-4.4
0.3	66.8	9.9	44.04	64.2	4.7	40.4	66.1	17.6	46.7	66.9	11.5	44.74	8.3	-6	-1.6
0.4	85.1	156	113.46	66.8	11.4	44.64	79.4	151.7	108.32	69.5	30.3	53.82	60.7	4.5	52.6
0.5	126.6	296.7	194.64	96.4	209.8	141.76	99.3	285	173.58	88	204.3	134.52	27.2	10.8	30.9
0.6	178.8	533.5	320.68	147.7	481	281.02	134.8	500.6	281.12	130.8	437.7	253.56	12.4	12.3	20.9
0.7	235	754.4	442.76	239.1	870.1	491.5	215.2	816.5	455.72	204.2	995	520.52	-11	-2.9	-17.6
0.8	295.4	991.4	573.8	313.5	1117.6	635.14	245.6	1219.4	635.12	266.2	1201	640.12	-10.7	-10.7	-11.6
0.9	351.9	1062.3	636.06	371.4	1070.8	651.16	303.1	991.8	578.58	355.9	1035.4	627.7	-2.4	9	1.3
1	398.6	1124.4	688.92	402.3	1092.4	678.34	346.8	1165.2	674.16	375.5	1110.4	669.46	1.5	2.1	2.8

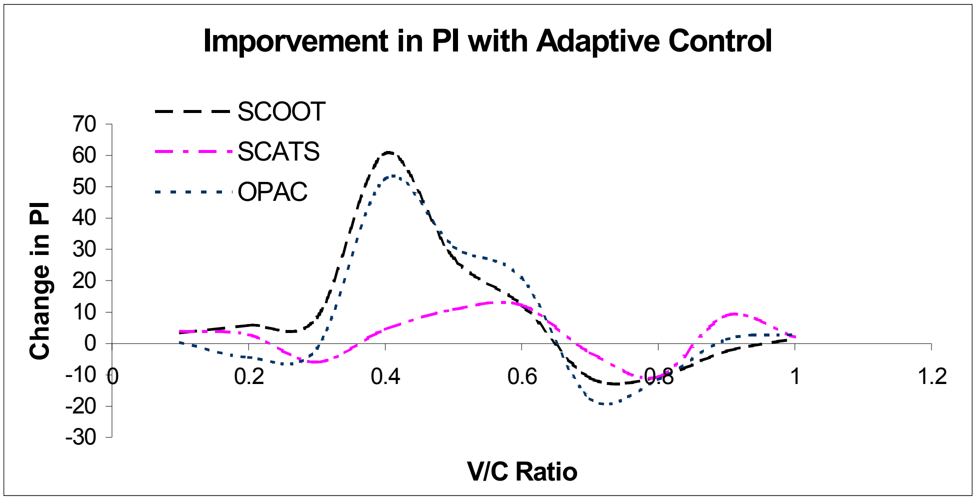


Figure 65 Percentage improvements in PI from adaptive control for the Intersection of Route 10 and Whippany Road

Intersection of Route 10 and Troy Hills Road

The intersection of Route 10 and Troy Hills Road has jughandle turns from both directions on Route 10. The cross-street demand is low compared with the main street. The signal on the intersection runs on three phases.

Table 45 Simulation results for the intersection of Route 10 and Troy Hills Road

Pre-timed			**SCOOT**			**SCATS**			**OPAC**			**% Improvement**		
Travel Time	**Stop Time**	**PI**	**Travel Time**	**Stop Time**	**PI**	**Travel Time**	**Stop Time**	**PI**	**Travel Time**	**Stop Time**	**PI**	**SCOOT**	**SCATS**	**OPAC**
56.5	14.8	39.82	50.9	2.4	31.5	53.5	10.1	36.14	53.1	9	7.2	20.9	9.2	76.90
58.8	35.2	49.36	52.5	18.5	38.9	53.6	17.4	39.12	53.5	13.2	8.88	21.2	20.7	58.06
118.8	636.4	325.84	52.2	20.8	39.64	58.8	54.3	57	60.4	61.4	28.16	87.8	82.5	74.68
139.4	776.1	394.08	59.1	71.9	64.22	62.8	96.4	76.24	76.4	219.3	91.32	83.7	80.7	79.52
243.3	1001.2	546.46	87.6	336.2	187.04	106.4	481.2	256.32	104.1	422.9	172.76	65.8	53.1	90.28
251.4	813.9	476.4	213.1	794.1	445.5	213.6	725.6	418.4	176.8	664.6	269.44	6.5	12.2	97.44
286.2	808.4	495.08	267.9	790.1	476.78	250.6	742.6	447.4	272	810.3	327.72	3.7	9.6	98.06
339.7	888.6	559.26	335.5	939	576.9	294.6	811.9	501.52	336.7	870.4	351.76	-3.2	10.3	98.16
415.3	972.9	638.34	398.2	943.8	616.44	334.7	823.2	530.1	395.3	872.7	352.68	3.4	17	97.34
458.4	1013.4	680.4	425.9	945.1	633.58	418.8	948.5	630.68	439.4	951	384	6.9	7.3	98.93

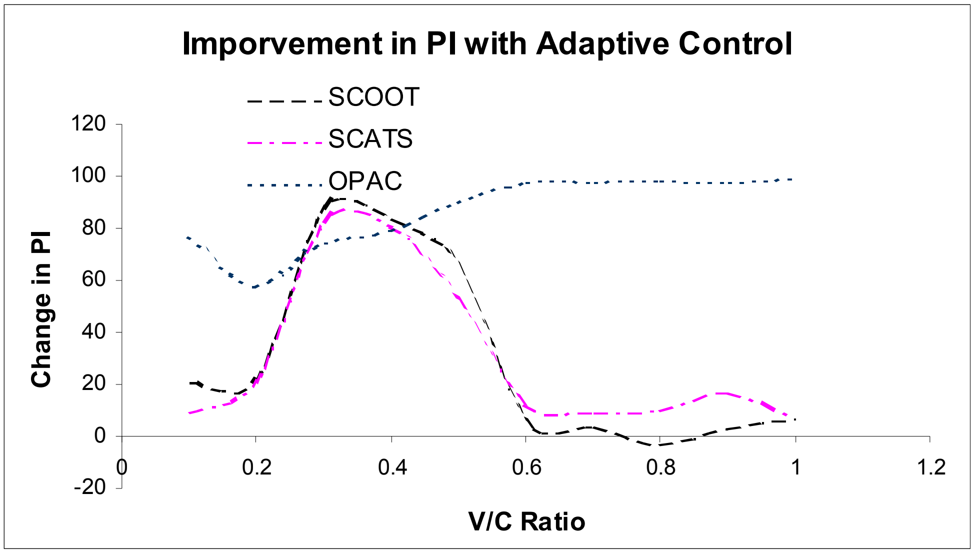


Figure 66 Percentage improvements in PI from adaptive control for the intersection of Route 10 and Troy Hills Road

Intersection of Route 18 and Eggers/S. Woodland Street

The intersection of Route 18 and Eggers/S. Woodland Street has a low cross street demand. The signal on this intersection runs on two phases.

Table 46 Simulation results for the intersection of Route 18 and Eggers/S. Woodland Street

V/C Ratio	Pre-timed			SCOOT			SCATS			OPAC			% Improvement		
	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	SCOOT	SCATS	OPAC
0.1	59.4	1	36.04	55	1.7	33.68	55	0.5	33.2	30.3	0.9	18.54	6.55	7.88	94.39
0.2	58.2	2.8	36.04	53.6	2.5	33.16	54.3	5.6	34.82	40.3	1	24.58	7.99	3.39	46.62
0.3	59.1	10.5	39.66	53	2.1	32.64	54.2	8.5	35.92	45.9	2.3	28.46	17.70	9.43	39.35
0.4	57.4	11.3	38.96	52.2	1.3	31.84	53.5	10.1	36.14	44.1	4.2	28.14	18.28	7.24	38.45
0.5	57.2	16.5	40.92	52.2	3.7	32.8	53.3	14.4	37.74	49.9	9.6	33.78	19.84	7.77	21.14
0.6	56.4	20.7	42.12	52.2	6	33.72	53.4	15.1	38.08	43.3	13	31.18	19.94	9.59	35.09
0.7	55	20.2	41.08	52.6	12.2	36.44	52.8	12.2	36.56	45	11.8	31.72	11.30	11.00	29.51
0.8	55.3	26.1	43.62	54.8	30.7	45.16	52.6	11.5	36.16	53.1	12.6	36.9	-3.53	17.10	18.21
0.9	56.2	31.6	46.36	57.9	60	58.74	55.8	36.4	48.04	54.1	18.6	39.9	-26.70	-3.62	16.19
1	62.7	96.7	76.3	76.1	191.9	122.42	68.2	80.8	73.24	66.5	100.5	80.1	-60.45	4.01	-4.74

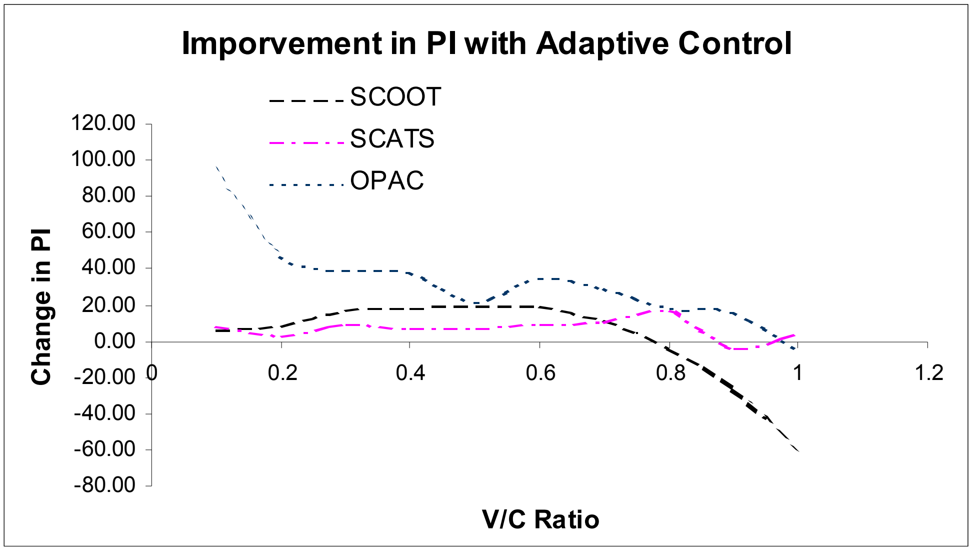


Figure 67 Percentage improvements in PI from adaptive control for the intersection of Route 18 and Eggers/S. Woodland Street

Intersection of Route 18 and Tices Lane

The intersection of Route 18 and Tices Lane has a jughandle turn from Route 18 northbound sides to vehicles to go onto westbound Tices lane. The signal at the intersection runs on two phases. The cross street demand on this intersection is high.

Table 47 Simulation results for the intersection of Route 18 and Tices Lane

V/C Ratio	Pretimed			SCOOT			SCATS			OPAC			% Improvement		
	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	SCOOT	SCATS	OPAC
0.1	54.6	3.4	34.12	51.6	1.8	31.68	53.3	3.9	33.54	51.6	1.8	31.68	7.2	1.7	4.7
0.2	52.1	6.2	33.74	48.4	1.5	29.64	50.9	8.1	33.78	48.8	2.4	30.24	12.2	-0.1	7.2
0.3	51.1	11.1	35.1	46.8	3.8	29.6	50.4	16.1	36.68	51.4	27.1	41.68	15.7	-4.5	-14.1
0.4	68.4	198.9	120.6	49.9	37.3	44.86	59	133.3	88.72	47.6	7.4	31.52	62.8	26.4	178.5
0.5	97.8	420.2	226.76	66.1	254	141.26	59.9	155.7	98.22	60.9	125.4	86.7	37.7	56.7	211.9
0.6	121.4	468.9	260.4	84.8	415.4	217.04	99.6	531.8	272.48	83.5	362.2	194.98	16.7	-4.6	77.1
0.7	167.8	708.9	384.24	123.1	653.7	335.34	143.9	566.6	312.98	142.5	689.2	361.18	12.7	18.5	18.7
0.8	185.9	777	422.34	189.4	758.9	417.2	217.5	678.8	402.02	174.5	772.9	413.86	1.2	4.8	4.5
0.9	289.2	940.9	549.88	247.4	844.9	486.4	289.9	783.4	487.3	240.6	799.2	464.04	11.5	11.4	34.7
1	324.6	981.7	587.44	301.6	899.8	540.88	317.7	888.4	545.98	311.4	908.7	550.32	7.9	7.1	12.3

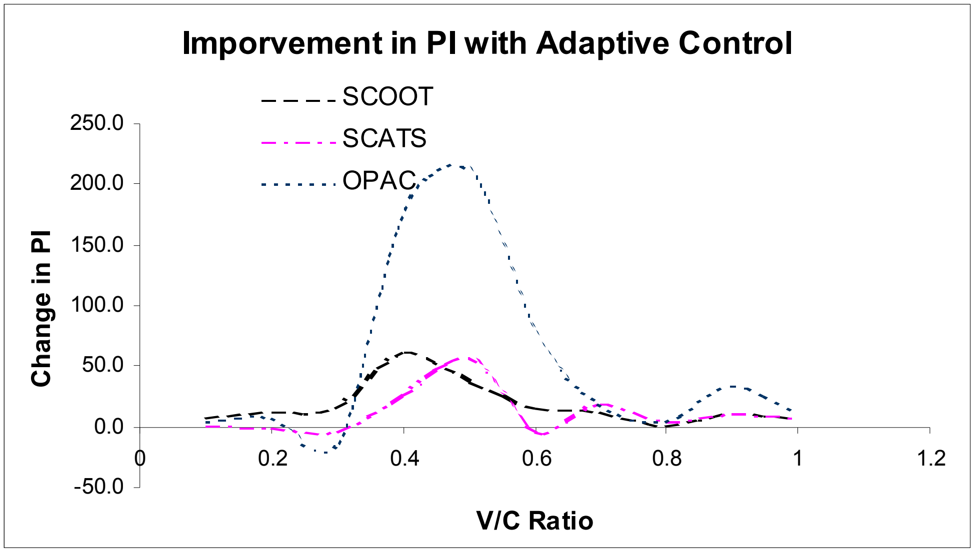


Figure 68 Percentage improvements in PI from adaptive control for the intersection of Route 18 and Tices Lane

Intersection of Route 23 and Oak Ridge Road

The intersection of Route 23 and Oak Ridge road has a jughandle turn from northbound Route 23. The cross street demand moving from Oak Ridge when making a right turn on Route 23 southbound is high. The signal runs on two phases.

Table 48 Simulation results for the intersection of Route 23 and Oak Ridge Road

V/C	Pre-timed			SCOOT			SCATS			OPAC			% Improvement		
Ratio	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	SCOOT	SCATS	OPAC
0.1	40.5	0.7	24.58	39.5	0.6	23.94	39.8	0.7	24.16	40.1	0.6	24.3	2.6	1.7	1.1
0.2	42.7	4.3	27.34	40.4	1	24.64	41.5	2.6	25.94	41.3	1.2	25.26	9.9	5.1	7.6
0.3	43.3	7	28.78	41.2	2.3	25.64	42	5.5	27.4	42.7	4.3	27.34	10.9	4.8	5.0
0.4	45.1	19.3	34.78	43	7.3	28.72	44.3	18	33.78	44.8	13.6	32.32	17.4	2.9	7.1
0.5	49.2	51.1	49.96	48	48	48	47.6	48	47.76	50.3	68.9	57.74	3.9	4.4	-15.6
0.6	61	160.1	100.64	56.9	152.6	95.18	48.5	55.4	51.26	57.1	134.2	87.94	5.4	49.1	12.6
0.7	80.3	325.6	178.42	65.2	237.5	134.12	70.1	306.1	164.5	75	265.4	151.16	24.8	7.8	15.3
0.8	103.9	338.1	197.58	97.3	420.4	226.54	85.2	342.4	188.08	89.7	297.8	172.94	-14.7	4.8	12.5
0.9	151.6	395.3	249.08	142.4	503.3	286.76	141.9	499.5	284.94	143	425.7	256.08	-15.1	-14.4	-2.8
1	205.2	432.5	296.12	216.5	536.9	344.66	168.3	704.1	382.62	199.3	431.4	292.14	-16.4	-29.2	1.3

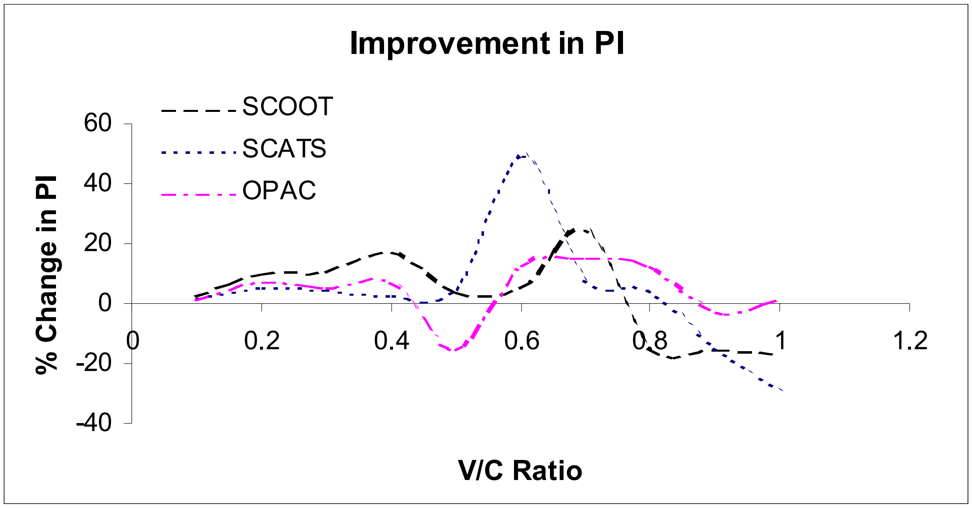


Figure 69 Percentage improvements in PI from adaptive control for the intersection of Route 23 and Oak Ridge Road

Intersection of Route 23 and LaRue Road

The intersection of Route 23 and LaRue road is separated by a median, and is there is a group control signal. The cross-street demand is low at this intersection.

Table 49 Simulation results for the intersection of Route 23 and LaRue Road

V/C	Pretimed			SCOOT			SCATS			OPAC			% Improvement		
Ratio	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	Travel Time	Stop Time	PI	SCOOT	SCATS	OPAC
0.1	42.2	0.5	25.52	37	0.1	22.24	42	0.7	25.48	39.3	0.9	23.94	12.9	0.2	6.6
0.2	45.6	2.2	28.24	40.8	0.8	24.8	43.9	2.4	27.3	42.6	2.7	26.64	12.2	3.3	6.0
0.3	49.1	6.9	32.22	43.4	1.9	26.8	46.3	8.3	31.1	44.9	4.9	28.9	16.8	3.5	11.5
0.4	50.1	11.6	34.7	44.4	3.1	27.88	47.4	11.8	33.16	44	4.4	28.16	19.7	4.4	23.2
0.5	52.6	13.5	36.96	44.5	4.2	28.38	48.7	15.7	35.5	50.1	18.3	37.38	23.2	4	-1.1
0.6	58.9	37.6	50.38	47.5	15.6	34.74	52.8	28.7	43.16	51.9	30.4	43.3	31	14.3	16.4
0.7	66.6	106.1	82.4	55.6	58.9	56.92	59.3	59.6	59.42	57.4	83.6	67.88	30.9	27.9	21.4
0.8	83.9	222.1	139.18	64.2	94.7	76.4	81.5	225.2	138.98	83.9	145.8	108.66	45.1	0.1	28.1
0.9	139.4	352.2	224.52	112.6	165.7	133.84	126.8	319.4	203.84	131.2	209.1	162.36	40.4	9.2	38.3
1	185.5	408.8	274.82	171.2	207.3	185.64	177.4	391.9	263.2	182.6	212.3	194.48	32.5	4.2	41.3

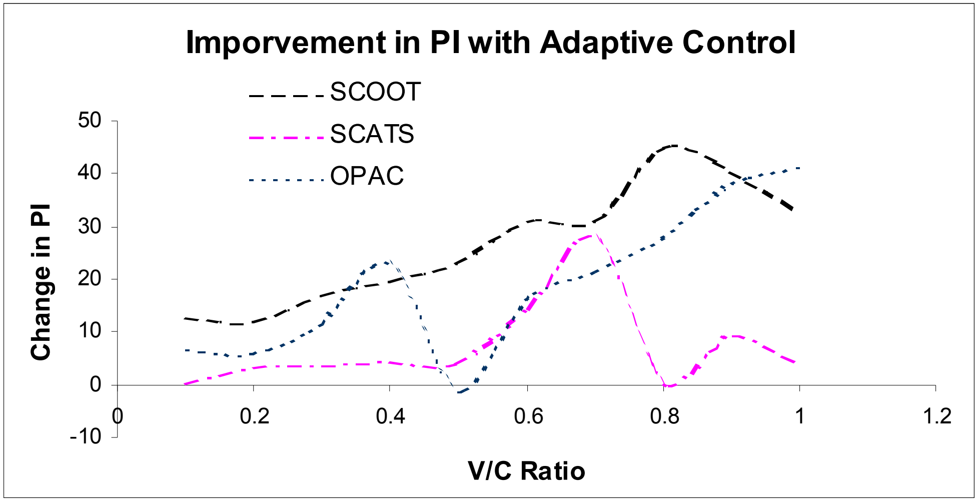


Figure 70 Percentage improvements in PI from adaptive control for the intersection for Route 23 and LaRue Road

Analysis of Paramics Simulation Results for NJ State Highway Intersections

It is difficult to see a consistent trend in the performance of prototype adaptive signal algorithms for different intersections. However as mentioned earlier, these intersections were selected such that they cover a broad range of intersection and network types. The results have effects of each parameter such as cross-street demand, network geometry, level of saturation etc. The results were also compared with the performance of adaptive control strategies reported in literature. It should be noted that for comparison the baseline control strategies, volume-to-capacity ratio, cross street demand, and network characteristics should be taken into account. For example, if there is a jug-handle turn at an intersection, it is difficult to install more detectors along the jug-handle and predict future arrivals during the head period. For (predictive) OPAC-Like algorithm that uses future arrivals at an intersection, the shorter link length on jug handles does not allow more detector placements. In such a case, it is possible only to predict the queue on the intersection, and the future arrivals are estimated even for the head period of the horizon. Hence, there is less information needed for dynamic programming model and it can be expected that OPAC-Like prototype will act differently for this type of intersections compared with intersections where there is enough space on the approaches to accommodate more detectors. For the reactive SCOOT-Like algorithm, the detector would be placed at the entrance of the jug-handle turn. For cases where cross street demand is less, the detector at the entrance of the jug-handle does not remain occupied for a longer duration. Hence reactive algorithm would give higher green time for the main street resulting in more stopped time for vehicles on the jug handle. This results in higher delay for SCOOT-Like prototype. For reactive/case-based SCATS-like algorithm, the detector at the stop line would mostly remain occupied, however the algorithm would not identify if the queue on jug-handle spills on to the main street. This affects the travel time on the main street and there will be reduction in improvement in the performance index. However proactive algorithm (OPAC-like), uses a good queue estimate algorithm, would get the best estimate of number of vehicles on the jug handle and can act to reduce the queue.

The percentage change in the performance index defined above varied for networks with jug-handle turns from 2.87% to 33.95% for SCOOT-like (reactive) algorithm, 2.58% to 30.26% for SCATS-like (case-based) algorithm, and 4.47% to 60.31% for OPAC-Like (proactive) algorithm. The average reduction the in performance index for higher volume to capacity ratio (0.6 to 1.0) was 17.36%, 7.09%, and 16.11% for SCOOT-Like, SCATS-Like and OPAC-Like algorithms, respectively.

Similarly, the number of phases also has an impact on performance of adaptive signal control prototypes. For Route 10 and Troy Hills Road intersection, SCOOT-like and SCATS-like prototypes fail to generate benefits at higher volume-to-capacity ratio. However, OPAC-like prototype gives a consistent improvement in PI, because it can respond to varying traffic demand, by changing the cycle length quickly. SCOOT-like and SCATS-like prototypes may be slower to react since change in cycle length is done in small increments at regular intervals of about 300 seconds. The percentage change in the performance index for networks with higher number of phases was 29.67% for SCOOT-like (reactive) algorithm, 30.26% for SCATS-like (case-based) algorithm, and 60.32% for OPAC-like (proactive) algorithm. However, the average reduction the in performance index for higher volume to capacity ratio (0.6 to 1.0) was 3.46%, 11.28%, and 40.53% for SCOOT-like, SCATS-like and OPAC-like algorithms, respectively.

From the results of Route 18/Eggers, S. Woodland street intersection and Route 18/Tices lane, we see that with higher cross street demand, SCOOT-like, SCATS-like and OPAC- like prototypes fail to generate higher benefits at higher volume to capacity ratio. The main street demand in all the cases is similar and both intersections run on two phases. The percentage change in the performance index defined above for intersections with higher cross street demand were 9.82% for SCOOT-like (reactive) algorithm, 9.55% for SCATS-like (case-based) algorithm, and 26.18% for OPAC-like (proactive) algorithm. However, the average reduction the in performance index for higher volume to capacity ratio (0.6 to 1.0) was –0.944%, 7.58%, and 14.93% for SCOOT-like, SCATS-like and OPAC-like algorithm. Compared to this, the overall reduction in performance index for intersections with lower cross street demand were 20.49%, 13.46%, and 28.44% for SCOOT-like, SCATS-like and OPAC-like prototypes respectively. For higher volume to capacity ratio the reduction was 18.82%, 13.14% and 35.96% for these three strategies, respectively. Hence adaptive signal prototypes work well on networks with lower cross street demand.

DEVELOPMENT OF DECISION SUPPORT MODEL

Introduction

With recent advances in computing and communication technologies, many adaptive traffic control strategies are now available to traffic engineers. Each of these strategies control signal timings in a different way, and hence are expected to give different results under given network and traffic conditions. Therefore, it is necessary to know how much improvement (if at all) can be expected from a particular traffic control strategy. The improvement can then be compared with the cost of implementing this specific strategy. These steps would then help in making a sound decision regarding implementation of adaptive control strategies.

A decision support tool is developed that helps in making such decisions. This tool has the following features:

- The input-output module allows for easy visual selection of intersections with the help of GIS software.
- A simulation tool for the macroscopic simulation of selected intersections that is available in the DSS database. The macroscopic simulation results can be compared with the optimized pre-timed traffic control strategy.
- It gives an overview of the performance of these control strategies based on previous implementation results.
- The expert system module decides whether a selected adaptive control strategy works, based on the rule base developed using previous implementation results and macroscopic simulation, MISP.
- A benefit-cost analysis capability allows for the evaluation of adaptive control strategy implementation for a selected intersection.

There are four modules in this tool:

- GIS-based input-output module
- Macroscopic simulation module, MISP
- Expert system module
- Benefit-Cost analysis module.

The macroscopic simulation module runs simulations as described in the previous sections. The module has an interactive GUI that displays simulation results and compares them with pre-timed simulation. The other modules, (i.e., input-output module, expert system module, and benefit-cost analysis module) are described in the next sections.

Geographical Information System (GIS) based Input-Output Module

The input-output module of the expert system selects an intersection or arterial for adaptive signal control simulations. This module also helps in the easy input and output of new traffic demand; new timing plans, and network geometry changes that might occur in the database in the future.

Software packages such as Synchro, Corsim, and Paramics store traffic demand, network geometry, timing plans, etc., of the network in their own format. To run simulations with adaptive control, it is necessary to pass this information to prototypes developed using C and Visual Basic programming languages. The best way is to store intersection details in a GIS database and use this information for simulation. Networks can be prepared and calibrated in micro-simulation packages such as Synchro, Corsim, and Paramics. By decoding their file formats such a database can be generated. In this project, GMPro was used to create the GIS-based application, while Synchro files of calibrated networks were used to generate the New Jersey intersection database.

GMPro is a platform employed to develop GIS applications. To develop a customized GMPro application, a customized script is written in Visual Basic using GeoMedia commands. In this application, a GIS map is developed so that a Visual Basic form is loaded when the user clicks on any intersection. This form enables the user to view data in a similar format as Synchro, so the application of Synchro in the actual GIS environment becomes as user friendly as an operating traffic network in Synchro itself. This is an attractive feature for traffic engineers who are familiar with Synchro. For importing and exporting data from Synchro to GMPro and from GMPro to Synchro, a customized Visual Basic program is developed.

GMPro can read traffic data from Universal Traffic Data Format (UTDF) files with .csv extensions (comma-delimited (.csv) text file format) and export it to an MS Access Database. The Universal Traffic Data Format (UTDF) is a standard format used by traffic engineering software packages and a readable format used to exchange traffic data for Synchro. The MS Access database can be accessed for viewing or editing by writing simple Visual Basic forms. The edited files can then be converted into UTDF files, which can be easily imported back into Synchro. Figure 71 gives a representation of the system diagram.

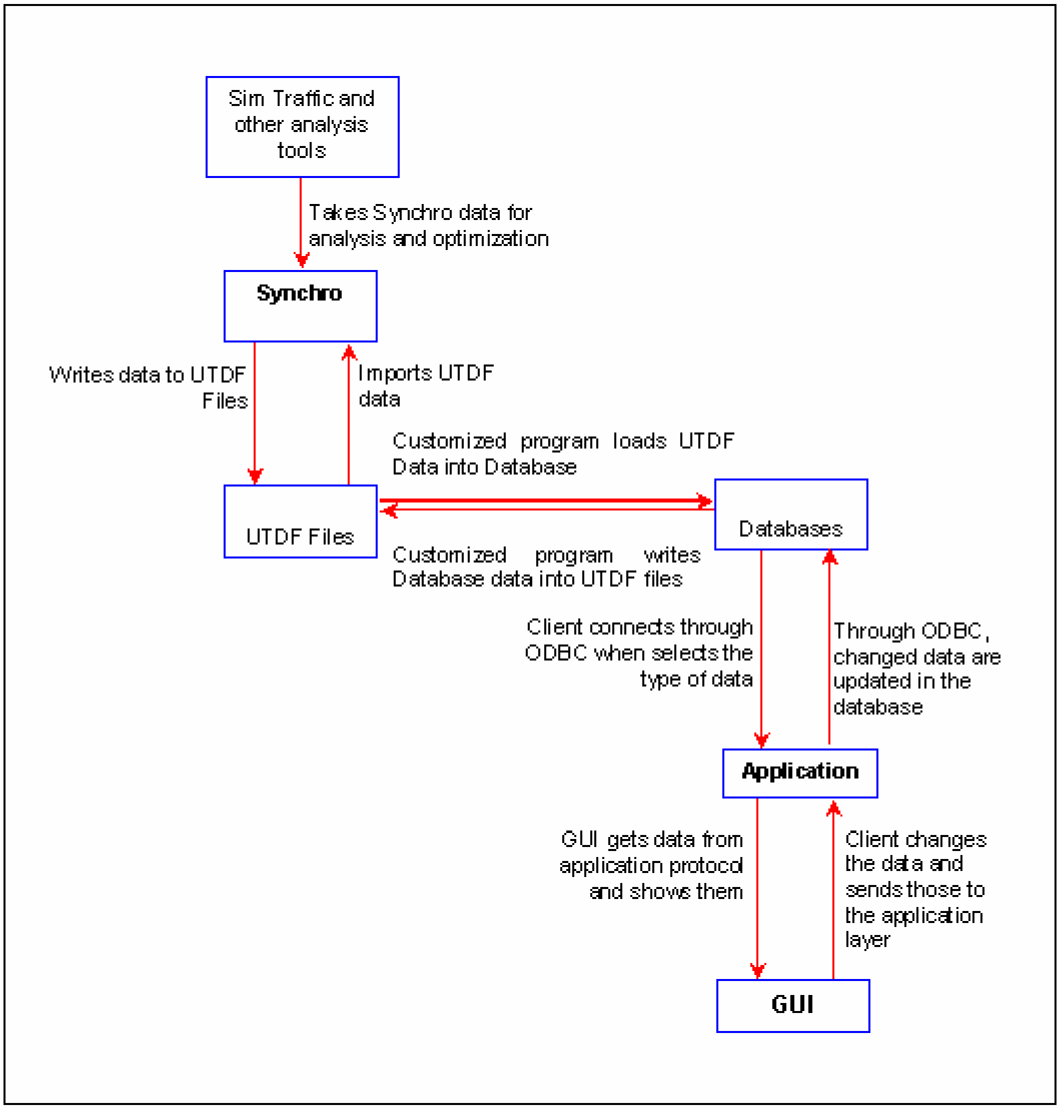


Figure 71 Block diagram for input-output module

Components of the system are:

- Synchro
- UTDF files
- Databases
- GMPro application
- User interface

Synchro supports UTDF format data. There are five types of data files in Synchro:

Lane Data (65)

Lane data contains information about lanes such as:

- Number of shared lanes
- Ideal saturated flow
- Lane width
- Grade
- Area type
- Storage length
- Storage lanes
- Total lost time
- Leading detector
- Trailing detector
- Turning speed
- Lane utilization factor
- Right-turn factors
- Left-turns factors
- Saturated flow rates
- Right-pad bike factor
- Left-pad factor
- Right-turn on red
- Saturated flow rate
- Headway factor

Layout Data (65)

Layout data shows information such as:

- Intersection ID
- X coordinate
- Y coordinate

Volume Data (65)

Volume data contains information such as:

- Peak hour factor
- Growth factor
- Heavy vehicles
- Bus blockages

Timing Data (65)

Timing data contains information such as:

- Traffic volumes
- Protected and permitted phases
- Detector phases
- Current cycle length

- Split information
- Lock timings
- Offset settings
- Sign control
- Yellow time
- All red time
- Phase lagging
- Allow lead/lag optimize
- Intersection capacity utilization

Phasing Data (65)

Phasing Window contains information such as:

- Current cycle length
- Actuated cycles
- Split information
- Minimum gap time before reduction
- Walk time
- Flashing don’t walk time

When the user activates the “Import Data” button, on the customized interface for GMPro, the Visual Basic program runs in the background. Then the application accesses UTDF data files exported from Synchro, reads data record by record, and inserts the records into the MS Access data files. There is a separate database for each type of data:

- Lane Data
- Layout Data
- Volume Data
- Timing Data
- Phasing Data

When the user activates the “Show Data” button, the customized VB application is connected to the related database for the specific data type through the Open DataBase Connectivity (ODBC) feature. As the user interface applet appears, data can be viewed in the exact similar fashion as in Synchro. The applet gets the data from the application protocol, which is one layer below the applet in the protocol stack.

When the user activates the “Save Data” button, the updated data are sent to the lower protocol, which is the application layer. The application is connected to the database through ODBC to enable data access.

When the user activates the “Export Data” button, the application loads and updates data into the UTDF files. Synchro not only writes data to the UTDF files, but also reads the data. The client can then run simulations in Synchro / Simtraffic or any other software to see the effects of the updates after importing the UTDF files through an interface in Synchro.

Development of the Application

Development of the application consists of the following steps:

Developing a Map

A map was developed in GMPro using Visual Basic 5.0 programming language.

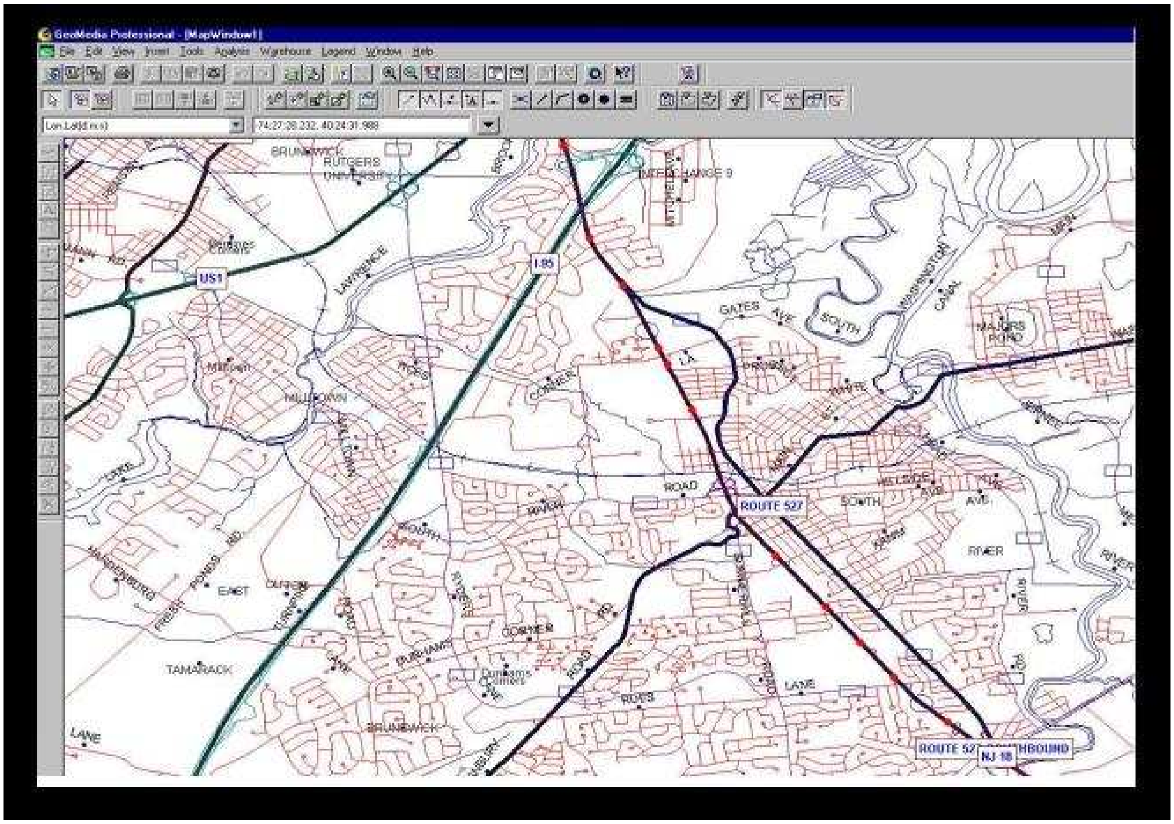


Figure 72 Customized map developed in GeoMedia professional

Dynamic Segmentation for Intersections

To perform dynamic segmentation, a new database table was developed called “points” in MGE_GIS_Export.mdb file for the intersections of Routes 10, 18, and 23. The primary key is defined as “NodeNo” field. From the control points and structures on the highways that are predefined, distances are found for the intersections as offsets from straight-line diagrams. “Points” and control_points tables are related with a common attribute, namely “sri”. In ODBC, a new connection is made for those overlaying intersections namely “projInter,” pointing to the MGE_GIS_Export.mdb database file. Distributed attributes are defined in the parameter file to enable the visual representation of the intersections on the GIS map. A known marker is used as the referencing system in the modular GIS environment, and the output data to be stored in the parameter file are selected as the point features. The necessary parameters are shown in Table 50.

Table 50 Parameter definition

Parameter	**Associated Column**
Linear Feature ID	Sri
Marker ID	Begin_Marker
Offset	Begin_Offset

Importing Data

To locate Synchro data in the GMPro application, the UTDF data must be first imported into the MS Access database. Compared with UTDF files saved in .csv format, it is easier to maintain and update data in MS Access database using GMPro. Data and the number of attributes are different for each intersection for the routes in the highway network. However, the GMPro application developed for this project can read such inconsistent files and create a meaningful database using the following procedure:

- Creating a separate database for each type of file (e.g., layout, lane, and timing etc.) for each intersection
- Creating a table within each database where each line read from the .csv files is stored
- Reading data from the UTDF files line by line, which contain the names of the attributes. The comma (,) character is considered as a field delimiter (the user-defined “split” function is used to separate the attributes from the single line and to put those names into an array).
- Creating separate fields in a table for each line read from UTDF files
- Appending the table into the database
- Saving the database.

Developing Customized User Interface in Visual Basic

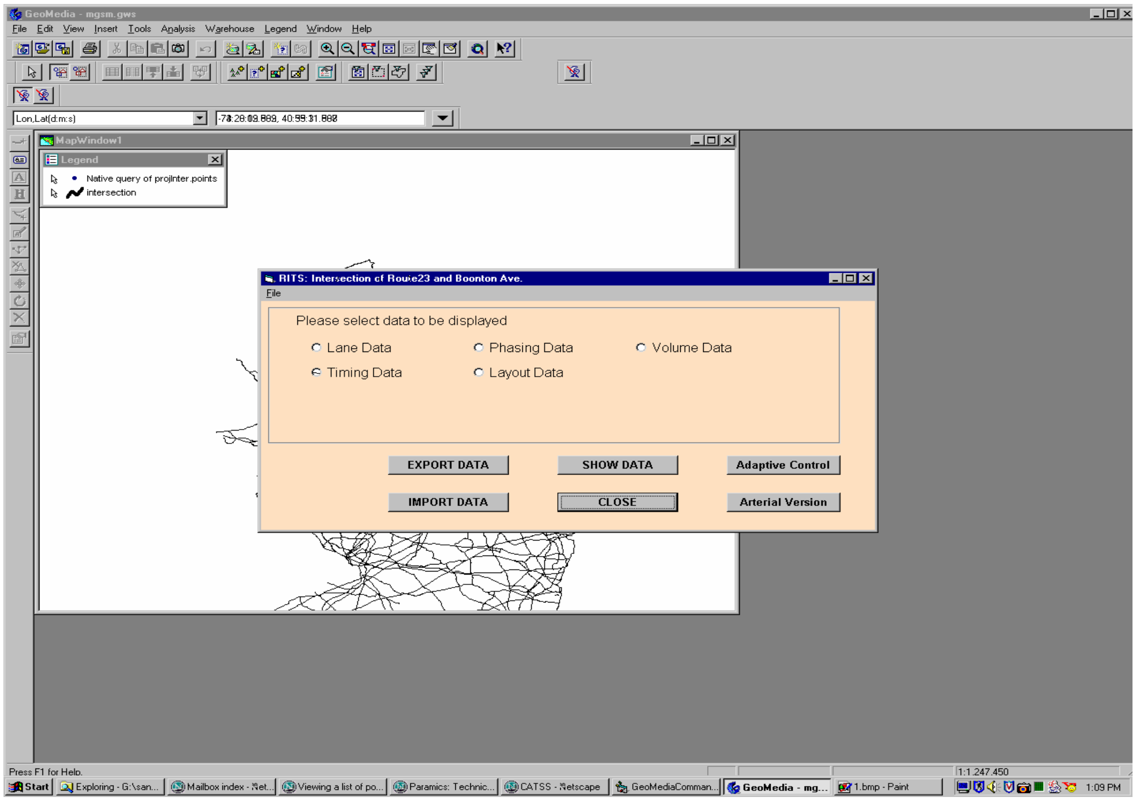
A customized user interface in Visual Basic shows the intersection routes on the heading of the form. There are several option buttons for the different types of data:

- Lane data
- Layout data
- Volume data
- Phasing data

- Timing data

There are four command buttons:

- Show data
- Import data
- Export data
- Close


RITS: Intersection of Route23 and Boonton Ave.

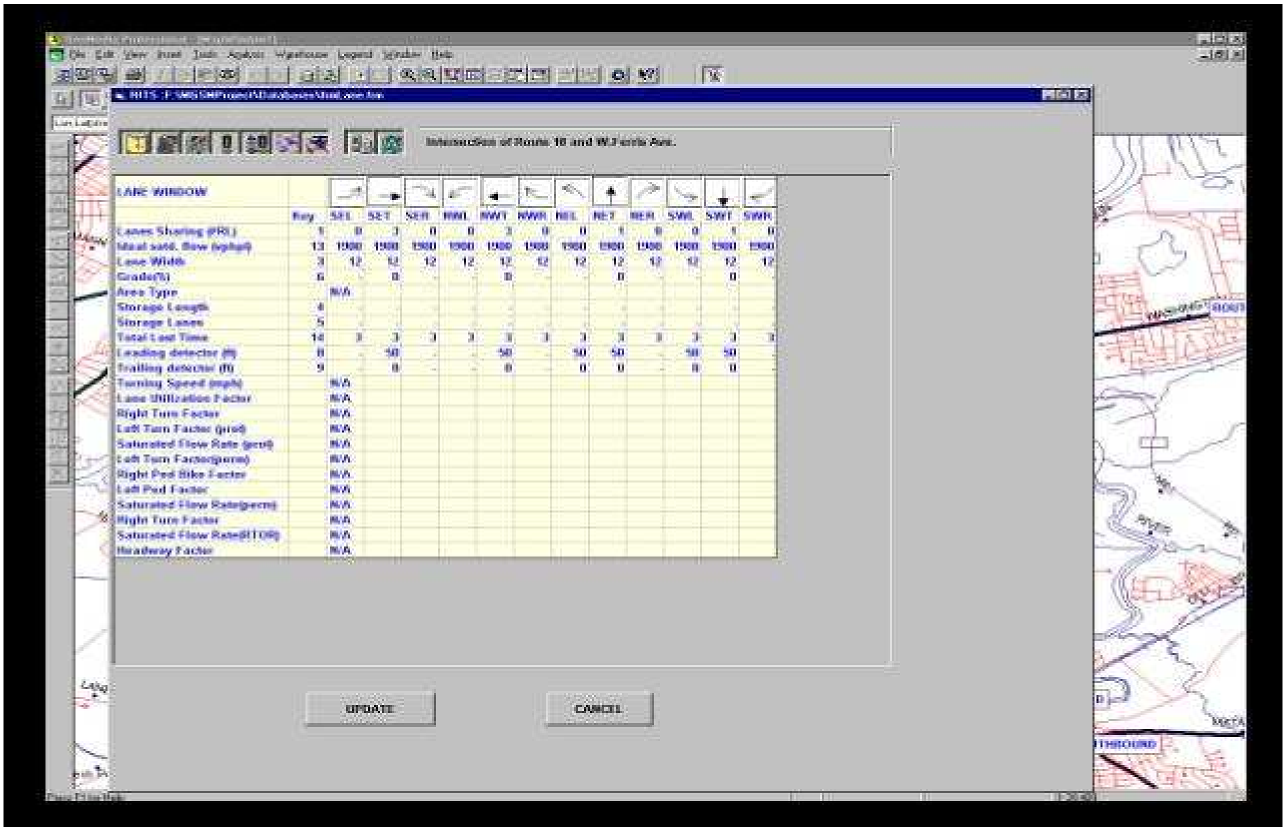
Please select data to be displayed

○ Lane Data
○ Phasing Data
○ Volume Data
● Timing Data
○ Layout Data

Figure 73 User interface in GeoMedia professional application

Sub-forms for Data Display

All the forms that enable the user to access and view data use Microsoft's Flex Grid control. MS Flex Grid is flexible in installing the display rules and is able to overwrite the particular cells, thus updating data directly in the database. It also has a user interface very similar to Synchro, and the user can be given authorization to change data and see the effects of such updates, which is shown in Figure 74.



Intersection of Route 10 and W.Ferris Ave.

LANE WINDOW	Key	SEL	SET	SER	NWL	NWT	NWR	NEL	NET	NER	SWL	SWT	SWR
Lanes Sharing (#RL)	1	0	1	0	0	1	0	0	1	0	0	1	0
Ideal satd. flow (vphpl)	13	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Lane Width	3	12	12	12	12	12	12	12	12	12	12	12	12
Grade(%)	6	-	0	-	-	0	-	-	0	-	-	0	-
Area Type		N/A											
Storage Length	4	-	-	-	-	-	-	-	-	-	-	-	-
Storage Lanes	5	-	-	-	-	-	-	-	-	-	-	-	-
Total Lost Time	14	3	3	3	3	3	3	3	3	3	3	3	3
Leading detector (ft)	8	-	50	-	-	50	-	50	50	-	50	50	-
Trailing detector (ft)	9	-	0	-	-	0	-	0	0	-	0	0	-
Turning Speed (mph)		N/A											
Lane Utilization Factor		N/A											
Right Turn Factor		N/A											
Left Turn Factor (prot)		N/A											
Saturated Flow Rate (prot)		N/A											
Left Turn Factor(perm)		N/A											
Right Ped Bike Factor		N/A											
Left Ped Factor		N/A											
Saturated Flow Rate(perm)		N/A											
Right Turn Factor		N/A											
Saturated Flow Rate(RTOR)		N/A											
Headway Factor		N/A											

Figure 74 A User interface showing lane data in GeoMedia professional application

Updating data

The user may update the data for any intersection on the GIS map via the "Update data" button in the customized user interface by. The user simply selects the cell in Flex Grid, enters the data, and activates the "Update Data" button. Because ODBC is created for each database, performing an "update query" completes updates. The user also views and analyzes the effects of the updates in Synchro. To maintain permanent data consistency, the update process takes place only in the database and not in Synchro. The user always exports data back to Synchro to view the updates in Synchro.

Exporting Data

The user may export data for any intersection on the GIS map through the customized user interface by the "Export data" button. The user exports data back to Synchro by clicking the command button on the main VB form for each intersection. A flat file is then created, and data is written in this file along with the field name, using the comma (,) as a delimiter with ODBC. After that Synchro is able to read data from the UTDF files (refer to Figure 71).

Expert System Development

The expert system module was developed to evaluate the performance of various adaptive traffic signal control strategies for a selected intersection operating under various network conditions. To make such a system it was essential to gather information regarding performance of adaptive control systems for specific networks and traffic demand conditions. Along with this information it was also necessary to know how to differentiate between different networks and traffic demand levels.

The expert system module is based on the method of classification commonly used as a learning method in the fields such as statistics and machine learning etc. The classification method is a two-step process. In the first step, a model is built analyze available sample space based on a set of attributes, while in the second step; the same model is used to classify future sample space. Of the available methods to build a classification model (classifier), the decision tree classifier is used to develop the decision support model for adaptive signal strategies. The decision tree classifier was selected because it can generate understandable rules and perform classification without much computational need as soon as the attributes important for prediction/classification are identified (18).

The method of classification was implemented using the following procedure:

1) Identification of attributes important in the classification procedure
2) Knowledge Acquisition
3) Knowledge Representation
4) Rule Base Development

Identification of attributes important in the classification procedure

A thorough literature review was conducted to identify the attributes that would be useful in the classification process. Finally the following parameters describing a network were selected as set of attributes:

1. Level of Saturation
    1. Undersaturated (v/c ratio < 0.7)
    2. Saturated (0.7 < v/c ratio < 1)
    3. Oversaturated (v/c ratio >= 1)

2. Intersection Spacing
    1. Close (within 300 feet)
    2. Distant (greater than 300 feet)
    3. Grid network

3. Cross Street Demand
    1. Low (< 400vphpl)
    2. High (> 400vphpl)

4. Traffic Volume
    1. Constant – Number of vehicles per cycle (or specified time interval) is almost constant
    2. Varying – Large variation in number of vehicles within one cycle

5. Number of Phases in Each Cycle
    1. Two (simple phasing)
    2. More than two (complex phasing)

Knowledge Acquisition

As a part of the knowledge acquisition process, necessary data regarding the performance of adaptive signal strategies was collected from a thorough literature study. It was observed that field implementation of adaptive signal strategies was conducted in few locations. As a result, the sample space created based on the literature review was very limited, and it covered only a few combinations of the above described network attributes.

To expand the sample space for the decision tree classifier, an attempt was made to derive results from the microscopic simulation results of adaptive signal prototypes (similar to the ones described in is there a section 5). OPAC, SCOOT and SCATS prototypes developed by the research team were tested using Paramics microscopic simulation tool. The intersections selected for microscopic simulations had different characteristics and they covered all possible sample space for the identified attributes of isolated intersections.

Based on the above network attributes, one can have a number of combinations for specifying network conditions. For example, one condition can be an oversaturated isolated intersection with a high cross street demand, varying volume and complex phase pattern, whereas another example could be an undersaturated arterial, with low cross-street demand along its intersections, with varying volumes and simple phase pattern.

Knowledge Acquired from Previous Studies

Network conditions during the implementation of adaptive traffic control strategies that are available in the literature were carefully studied. Often it was difficult to determine a specific network condition for implementation results found in the literature. However, results were obtained for twelve network conditions for four adaptive traffic signal control strategies from the literature. These are listed as Rules 1 through Rule 12.

Rule 1:

Conditions:

Level of Saturation	**Saturated**
Network Type	**Grid**
Intersection Spacing	**Close**
Cross-Street Demand	**Low**
Traffic Volume	**Varying**
Phase Pattern	**Simple**

Performance:

	SCOOT*[1]	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control	Optimized fixed time plan	**-**	**-**	**-**
MOE's Compared with Baseline Signal				
Average Travel Time	-27% to 30%	Not Available **Comments:** This strategy has the ability to adapt to the traffic and network conditions described	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections	**Not Available** **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections
Total Delay	-23% to 30%			
Stopped Delay	Not available			
Control Delay	Not available			
Total Number of Stops	-25% to 31%			
Traffic Progression	Not available			
Basis of Results	Simulation	Expert View	Expert View	Expert View

* Level of saturation for peak period is assumed to be saturated

***Reference**: 1. Blake G. Hansen, Peter T Martin and H. Joseph Perrin Jr.," SCOOT Real-Time Adaptive Control in a CORSIM Simulation Environment," Transportation Research Record 1727, Paper No. 00-1520.*

Rule 2:

Conditions:

Level of Saturation	**Undersaturated**
Network Type	**Grid**
Intersection Spacing	**Close**
Cross-Street Demand	**Low**
Traffic Volume	**Varying**
Phase Pattern	**Simple**

Performance:

	SCOOT*[1]	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control	Optimized fixed time plan	-	-	-
MOE's Compared with Baseline Signal				
Average Travel Time	-22% to -28%	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections
Total Delay	-21% to -27%			
Stopped Delay	Not available			
Control Delay	Not available			
Total Number of Stops	-21 to -23%			
Traffic Progression	Not available			
Basis of Results	Simulation	Expert View	Expert View	Expert View

***** Level of saturation for off peak period is assumed to be undersaturated

***Reference**: 1. Blake G. Hansen, Peter T Martin and H. Joseph Perrin Jr.," SCOOT Real-Time Adaptive Control in a CORSIM Simulation Environment," Transportation Research Record 1727, Paper No. 00-1520.*

Rule 3:

Conditions:

Level of Saturation	**Saturated**
Network Type	**Grid**
Intersection Spacing	**Wide**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT*[1]	**SCATS[2]**	**OPAC**	**RHODES**
Baseline Signal Control	Optimized fixed time plan		**-**	**-**
MOE's Compared with Baseline Signal				
Average Travel Time	3.31% to 6.43%	Not available	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections
Total Delay	3.31% to 6.43%	Not available		
Stopped Delay	Not available	Not available		
Approach Delay	Not available	"Significant" increase (by 1.34 sec/veh)		
Total Number of Stops	Not available	Not available		
Traffic Progression	Not available	Not available		
Basis of Results	Field Results	Simulation	Expert View	Expert View

* Level of saturation for peak period is assumed to be saturated

***Reference**:*

1. *James E. Moore, II, R.Jayakrishnan, M.G. McNally, C. Arthur MacCarley ,"Evaluation of the Anaheim Advanced Traffic Control System Field Operational Test: Introduction and Task A: Evaluation of SCOOT Performance,"UCB-ITS-PRR-99-26, California PATH Research Report*
2. *Brian Wolshon ,William C. Taylor, "Impact of Adaptive Signal Control on Major and Minor Approach Delay,", Journal of Transportation Engineering/January/February 1999, Pg. 30-38.*

Rule 4:

Conditions:

Level of Saturation	**Undersaturated**
Network Type	**Grid**
Intersection Spacing	**Wide**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT*[1]	**SCATS[2]**	**OPAC**	**RHODES**
Baseline Signal Control	Optimized fixed time plan		**-**	**-**
MOE's Compared to Baseline Signal				
Average Travel Time	-35.53% to 11.4%	Not available	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections (Under saturated Conditions only)
Total Delay	-35.53% to 11.4%	Not available		
Stopped Delay	Not available	Not available		
Approach Delay	Not available	Reported decrease but "not significant"		
Total Number of Stops	Not available	Not available		
Traffic Progression	Not available	Not available		
Basis of Results	Field Results	Simulation	Expert View	Expert View

***** Level of saturation for off peak period is assumed to be under saturated

***Reference**:*

1. *James E. Moore, II, R.Jayakrishnan, M.G. McNally, C. Arthur MacCarley ,"Evaluation of the Anaheim Advanced Traffic Control System Field Operational Test: Introduction and Task A: Evaluation of SCOOT Performance,"UCB-ITS-PRR-99-26, California PATH Research Report*
2. *Brian Wolshon ,William C. Taylor, "Impact of Adaptive Signal Control on Major and Minor Approach Delay,", Journal of Transportation Engineering/January/February 1999, Pg. 30-38.*

Rule 5:

Conditions:

Level of Saturation	**Undersaturated/Saturated**
Network Type	**Arterial**
Intersection Spacing	**Close**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control	Optimized fixed time plan		**-**	**-**
MOE's Compared with Baseline Signal				
Average Travel Time	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	**Not** Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections
Total Delay				
Stopped Delay				
Approach Delay				
Total Number of Stops				
Traffic Progression				
Basis of Results	Expert View	Expert View	Expert View	Expert View

(Arterials are one of the main places where adaptive control strategies are implemented. However, such strategies are rarely conducted in an undersaturated network; hence, data for this rule is not available in the literature).

Rule 6:

Conditions:

Level of Saturation	**Undersaturated**
Network Type	**Arterial**
Intersection Spacing	**Wide**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT[1]	**SCATS**[2]	**OPAC**[3]	**RHODES**[3]
Baseline Signal Control	Optimized fixed time plan	Optimized fixed time plan	Optimized Semi-Actuated	Optimized Semi-Actuated
MOE's Compared with Baseline Signal				
Average Travel Time	-17%	-15 to -31.8%	Not available	Not available
Throughput	Not available	Not available	-1.2%	0.75%
Total Delay	-8%	-7.6%	3.6%	-37.7%
Stopped Delay	Not available	-14.7%	Not available	Not available
Approach Delay	Not available	Not available	Not available	Not available
Total Number of Stops	-22%	Not available	7.89%	-14.08%
Traffic Progression	Not available	5.6% increase in bandwidth	Not available	Not available
Basis of Results	Simulation	Field Implementation	Simulation	Simulation

Reference*:*

1. *Chintan S. Jhaveri, Joseph Perrin Jr., Peter T. Martin, "SCOOT Adaptive Signal Control: An Evaluation of its Effectiveness over a Range of Congestion Intensities", Transportation Research Board, January2003 Annual Meeting.*
2. *William C. Taylor, Ahmed S Abdel-Rahim, "Analysis of Corridor Delay Under SCATS Control",* www.itsdocs.fhwa.dot.gov/%5CJPODOCS%5CREPTS_TE/ 75V01!.PDF
3. *"Laboratory Evaluation of RT-TRACS Prototype Strategy", Prepared by ITT Industries Inc., Systems Division, PO 15012, Colorado Springs, CO 80935, March 31, 1999.*

Rule 7:

Conditions:

Level of Saturation	**Saturated**
Network Type	**Arterial**
Intersection Spacing	**Wide**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT[1]	**SCATS**[2]	**OPAC**[3]	**RHODES**[4]
Baseline Signal Control	Optimized fixed time plan	Optimized fixed time plan	Optimized Semi-Actuated	Optimized Semi-Actuated
MOE's Compared to Baseline Signal				
Average Travel Time	9%-13%	-6.6% to 20.3%	-4.4% to-6.9%	1% to 4%
Total Delay	Not available	-2.6%	Not available	Not available
Stopped Delay	Not available	-4.6%	Not available	"Decreased"
Approach Delay	Not available	Not available	Not available	Not available
Total Number of Stops	Not available	Not available	Not significant	Not available
Traffic Progression	Not available	9.2% to 23.2% increase in bandwidth	Not available	Not available
Basis of Results	Simulation	Field Implementation	Field Implementation	Simulation/Field Implementation

***Reference**:*

1. *Chintan S. Jhaveri, Joseph Perrin Jr., Peter T. Martin, "SCOOT Adaptive Signal Control: An Evaluation of its Effectiveness over a Range of Congestion Intensities", Transportation Research Board, January2003 Annual Meeting.*
2. *William C. Taylor, Ahmed S Abdel-Rahim, "Analysis of Corridor Delay Under SCATS Control",* www.itsdocs.fhwa.dot.gov/%5CJPODOCS%5CREPTS_TE/ 75V01!.PDF
3. *Christina M. Andrews, S. Manzur Elahi, and James E Clark*, *Evaluation of New Jersey Route 18 OPAC/MIST Traffic-Control System, Transportation Research Record 1603, Paper no. 971253.*
4. "Evaluation of the SR-522 Seattle, Washington ACS field test," Prepared by ITT Industries Inc., Systems Division, PO 15012, Colorado Springs, CO 80935, March 31, 1999.

For SCOOT the benefits in travel time along an arterial were reported to be around 1% for v/c ratio of 1.0 and 0% for v/c ratio of 1.1.

Rule 8:

Conditions:

Level of Saturation	**Saturated**
Network Type	**Arterial**
Intersection Spacing	**Wide**
Traffic Volume	**Constant**
Phase Pattern	**Complex**

Performance:

	SCOOT	**SCATS**	**OPAC[1]**	**RHODES**
Baseline Signal Control	Optimized fixed time plan	Optimized fixed time plan	Optimized Semi-Actuated	Optimized Semi-Actuated
MOE's Compared with Baseline Signal				
Average Travel Time	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not significant	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections (Under saturated Conditions only)
Total Delay			Not available	
Stopped Delay			Not available	
Approach Delay			Not available	
Total Number of Stops			Not significant	
Traffic Progression			Not Available	
Basis of Results	Expert View	Expert View	Field Implementation	

***Reference**:*

1. *Christina M. Andrews, S. Manzur Elahi, and James E Clark, Evaluation of New Jersey Route 18 OPAC/MIST Traffic-Control System, Transportation Research Record 1603, Paper no. 971253.*

Rule 9:

Conditions:

Level of Saturation	**Oversaturated**
Network Type	**Arterial**
Intersection Spacing	**Wide**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT	**SCATS**	**OPAC**[1]	**RHODES**[2]
Baseline Signal Control	Optimized fixed time plan	Optimized fixed time plan	Optimized Semi-Actuated	Optimized Semi-Actuated
MOE's Compared with Baseline Signal				
Average Travel Time	Not available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not available	2% to -14%
Total Delay			12.5%	Not available
Through put			-4.2%	Not available
Stopped Delay			Not available	-32% to 18%
Approach Delay			Not available	Not available
Total Number of Stops			8.13%	Not available
Traffic Progression			Not available	"Better"
Basis of Results	Expert View	Expert View	Simulation/Field Implementation	Field Implementation

Reference:

1. *"Evaluation of the Reston Parkway RT-TRACS Field Test," Prepared by ITT Industries Inc., Systems Division, PO 15012, Colorado Springs, CO 80935, March 31, 1999.*
2. *"Evaluation of the SR-522 Seattle, Washington ACS field test," Prepared by ITT Industries Inc., Systems Division, PO 15012, Colorado Springs, CO 80935, March 31, 1999.*

Rule 10:

Conditions:

Level of Saturation	**Under Saturated**
Network Type	**Isolated Intersection**
Intersection Spacing	**Not applicable**
Traffic Volume	**Varying**
Phase Pattern	**Simple**

Performance:

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control		Optimized fixed time plan	Time of Day Operation	
MOE's Compared with Baseline Signal				
Average Travel Time	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not available	Not available	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described
Average Intersection Delay		9.0%	Not available	
Stopped Delay		Not available	-39.1% "Significant"	
Approach Delay		Not available	Not available	
Total Number of Stops		Not available	"Not significant"	
Basis of Results	Expert Comment	Simulation	Field Implementation	Expert Comment

Reference:

1. *Rahmi Akcelik, Mark Besley, Edward Chung, "An Evaluation of SCATS Master Isolated Control,"Akcelik & Akcelik Associates Ptv. Ltd.*
2. *Christina M. Andrews, S. Manzur Elahi and James E Clark, Evaluation of New Jersey Route 18 OPAC/MIST Traffic-Control System, Transportation Research Record 1603, Paper no. 971253.*

Rule 11:

Conditions:

Level of Saturation	**Saturated**
Network Type	**Isolated Intersection**
Intersection Spacing	**Not applicable**
Traffic Volume	**Varying**
Phase Pattern	

Performance:

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control		Optimized fixed time plan		
MOE's Compared with Baseline Signal				
Average Travel Time	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not Available	Not Available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not Available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections (Under saturated Conditions only)
Average Intersection Delay		-5.4% to –42.1%		
Stopped Delay		Not available		
Approach Delay		Not available		
Total Number of Stops		Not available		
Basis of Results	Expert Comment	Simulation	Expert Comment*	Expert Comment

Reference:

1. *Rahmi Akcelik, Mark Besley, Edward Chung, "An Evaluation of SCATS Master Isolated Control,"Akcelik & Akcelik Associates Ptv. Ltd.*

Rule 12:

Conditions:

Level of Saturation	**Oversaturated**
Network Type	**Isolated Intersection**
Intersection Spacing	**Not applicable**
Traffic Volume	**Varying**
Phase Pattern	**Complex**

Performance:

	SCOOT	SCATS	OPAC	RHODES
Baseline Signal Control		Optimized fixed time plan		
MOE's Compared to Baseline Signal				
Average Travel Time	Not available **Comments:** The strategy has the ability to adapt to the traffic and network conditions described	Not Available	Not available **Comments:** This strategy is applicable to undersaturated arterials with possible extension to saturated one.	Not available **Comments:** This strategy is applicable mainly to arterials with widely spaced intersections (Undersaturated Conditions only)
Average Intersection Delay		1.6% to 2.8%		
Stopped Delay		Not available		
Approach Delay		Not available		
Total Number of Stops		Not available		
Basis of Results	Expert Comment	Simulation	Expert Comment*	Expert Comment

Reference:

1. *Rahmi Akcelik, Mark Besley, Edward Chung, " An Evaluation of SCATS Master Isolated Control,"Akcelik & Akcelik Associates Ptv. Ltd.*

Microscopic Simulation Results for OPAC, SCOOT, and SCATS

Because results from the literature review were available for only 12 cases, additional simulations using prototypes developed were conducted using Paramics, SCOOT, SCATS, and OPAC prototypes, which were tested on selected isolated intersections on NJ highways. Intersections were selected based on number of phases, traffic demand, etc. Traffic demand was varied so that the level of saturation covered the three levels (undersaturated, oversaturated and saturated). Based on these results, new sets of rules were prepared, which are listed below (from Rules 13 to 20).

Rule 13:

Level of Saturation	**Saturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**Low**
Phase Pattern	**Simple**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control	**Optimized Pre-timed**	**Optimized Pre-timed**	**Optimized Pre-timed**	
MOE’s Compared with Baseline Signal				
Average Travel Time	-32%	-29.0%	-14.59%	
Average Intersection Delay		-5.4% to -42.1%[2]		
Total Delay				
Stopped Delay	-69.1%	-52.4%	-46.6%	
Control Delay				
Total Number of Stops				

Rule 14:

Level of Saturation	**Undersaturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**Low**
Phase Pattern	**Simple**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control			**Optimized Pre-timed**	
MOE’s Compared with Baseline Signal				
Average Travel Time	-10.2%	-8.2%	-12.911%	
Average Intersection Delay		**9.0%[2]**		
Total Delay				
Stopped Delay	-62.0%	-26.6%	-24.16%[3] to -39.1%[1]	
Control Delay				
Total Number of Stops			“Not Significant”[1]	

References

1. *Christina M. Andrews, S. Manzur Elahi and James E Clark, “Evaluation of New Jersey Route 18 OPAC/MIST Traffic-Control System”, Transportation Research Record 1603, Paper no. 971253*
2. *Rahmi Akcelik, Mark Besley, Edward Chung, “ An Evaluation of SCATS Master Isolated Control,”Akcelik & Akcelik Associates Ptv. Ltd.*
3. *Paramics Simulation of OPAC like prototype.*

Rule 15:

Level of Saturation	**Oversaturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**Low**
Phase Pattern	**Simple**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control	**Optimized Pre-timed**	**Optimized Pre-timed**	**Optimized Pre-timed**	
MOE's Compared with Baseline Signal				
Average Travel Time	-25.8%	-19.1%	-6.67%	
Total Delay				
Stopped Delay	-20.9%	-28.9%	-28.28%	
Control Delay				
Total Number of Stops				

Rule 16:

Level of Saturation	**Saturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**High**
Phase Pattern	**Simple**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control			**Optimized Pre-timed**	
MOE's Compared with Baseline Signal				
Average Travel Time	-3.1%	-5.9%	-9.53%	
Total Delay				
Stopped Delay	9.5%	14.9%	1.04%	
Control Delay				
Total Number of Stops				

Rule 17:

Level of Saturation	**Oversaturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**High**
Phase Pattern	**Simple**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control				
MOE's Compared with Baseline Signal				
Average Travel Time	-6.8%	-5.7%	-4.24%	
Total Delay				
Stopped Delay	1.4%	1.7%	-2.97%	
Control Delay				
Total Number of Stops				

Rule 18:

Level of Saturation	**Undersaturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**Low**
Phase Pattern	**Complex**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control				
MOE's Compared with Baseline Signal				
Average Travel Time	-25.6%	-22.1%	-32.11%	
Total Delay				
Stopped Delay	-76.0%	-63.3 %	-56.64%	
Control Delay				
Total Number of Stops				

Rule 19:

Level of Saturation	**Oversaturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**Low**
Phase Pattern	**Complex**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control				
MOE's Compared with Baseline Signal				
Average Travel Time	-4.1%	-11.7%	-4.14%	
Average Intersection Delay		1.6% to 2.8%[1]		
Total Delay				
Stopped Delay	-1.3%	-6.7%	-6.51%	
Control Delay				
Total Number of Stops				

Rule 20:

Level of Saturation	**Saturated**
Intersection Spacing	**Isolated**
Cross Street Demand	**High**
Phase Pattern	**Complex**

	SCOOT	**SCATS**	**OPAC**	**RHODES**
Baseline Signal Control				
MOE's Compared with Baseline Signal				
Average Travel Time	-3.7%	-11.3%	-3.55%	
Total Delay				
Stopped Delay	-5.8%	-15.6%	-4.03%	
Control Delay				
Total Number of Stops				

Reference:

1. *Rahmi Akcelik, Mark Besley, Edward Chung, " An Evaluation of SCATS Master Isolated Control,"Akcelik & Akcelik Associates Ptv. Ltd.*

Knowledge Representation

As discussed in the previous section, implementation results for SCOOT, SCATS, OPAC, and RHODES were found in the literature. The results were from actual field implementation, expert simulations by developers of adaptive strategies and prototype simulations made to increase the knowledge database. For prototype simulations carried out at the RITS laboratory, the final results in the above rules were obtained by taking the arithmetic average of the results of all considered intersections.

It was observed that results from field implementations, simulation results, and prototype simulations vary for each adaptive control strategy. Hence, there was a need to develop a valuation module that would use a value function to estimate the "utility" of deploying adaptive control strategies at different types of network. The process of estimation value function for decision making for public agencies was first proposed by Saaty (70,71). He described this process as "Analytical Hierarchy Process" and proposed the estimation of the parameters or weights of a value function that ranks the alternatives using input from real-world decision makers. The proposed function is in the following form:

$$w_1 x_1 + w_2 x_2 + \ldots\ldots\ldots w_n x_n = \sum_{i=1}^{n} w_i x_i \tag{20}$$

where,
w_i = weights for each variable
x_i = considered performance measures

In this application, weights can be interpreted as how reliable each output is considered to be and as a performance measure for the reliability of field implementation, expert simulation and prototype simulation results. It is logical to assign more weight to field implementation results because these results are obtained under real-world conditions and less weight to simulation results, because there are certain assumptions involved while obtaining them.

Possible combinations of available performance measures are shown below:

1) Field Results + Expert Simulation Results + Prototype Simulation Results
2) Expert Simulation Results + Prototype Simulation Results
3) Field Results + Prototype Simulation Results

Assigning Weights to Performance Measures Values Obtained from Different Types of Implementation

Field Implementation Results: More weight should be assigned to field implementation results because these results are obtained under real-world conditions. Lower or higher weights can be applied for particular cases wherein field results are obtained under certain conditions of traffic demand (e.g., results from traffic control on a network serving traffic to a public event in the vicinity)

Expert Simulation Results: Simulations performed by commercial providers of the control strategies are considered as expert simulations. These simulations do not accurately represent real-world conditions because accuracy and reliability of the simulations are limited by the type of package used and type of calibration and modeling efforts undertaken by the modelers (e.g., different packages uses different distributions for arrivals, different traffic models). In this case, weight assigned is slightly lower compared with field implementation results.

RITS Prototype Simulation Results using Paramics: The prototype of a control strategy is usually not the same as the actual control strategy. Prototypes are developed from the available information about the control strategy. They may or may not represent correct version of the strategies currently being used. Hence, less weight is assigned to these results.

Table 51 Weight assignments to available results for adaptive control strategies

Case	Weights
Field Implementation / Expert Simulation / Prototype Simulation	0.40 / 0.35 / 0.25
Expert Simulation / Prototype Simulation	0.60 / 0.40
Field Implementation / Prototype Simulation	0.65 / 0.35

Ex: If there is a 2% decrease in delay from field implementation and an 8% decrease from prototype simulation, then we can say that the expected decrease would be 0.65*0.02 + 0.35*0.08, which amounts to 4.1%

Development of the Rule Base

After developing the knowledge base, the next step is the development of the rule base, which allows for the use of this acquired knowledge to make decisions. A hierarchical decision structure was employed to obtain meaningful results from the knowledge database. After selecting a network, it is classified based on its travel demand, network geometry, signaling method, etc., into one of the above-discussed rules. The rule base then combines the available results with the weighted average approach (as discussed in knowledge representation) and gives the performance of the adaptive systems for a list of measures of effectiveness. If a rule is not available for a given network, the rule base gives expert opinion about the performance of adaptive systems, based on the information found in the literature.

To make the decision process simple, the rules were converted into individual text files. Each file was named based on the network parameters discussed before. For example, a file named “opacIOSCH.txt” file would have rules for the OPAC algorithm for an isolated intersection with an oversaturated travel demand, a complex phase pattern and a high cross-street demand. The rule base extracts this information from the selected network using the flow chart given in Figure 75. It then opens text files associated with the prefixes that it gets through the decision process. This is when the software uses the input-output module. After an intersection is selected, the software gathers important information regarding the intersection such as traffic demand, number of phases, and cross street demand from the Synchro interface. It might not be possible to get all the information such as level of saturation. In these cases, the software uses HCM formulas to calculate the required parameter from the available data.

Because the input-output module is not capable of selecting multiple intersections at this point, a separate form is made to allow the user to select other network types, such as Grid and Arterial. A user-friendly interface then allows selection of other parameters, and the rule base opens the appropriate text file for the selected network. The process of selecting the intersection or arterial/grid is explained in detail in the user manual of the developed software.

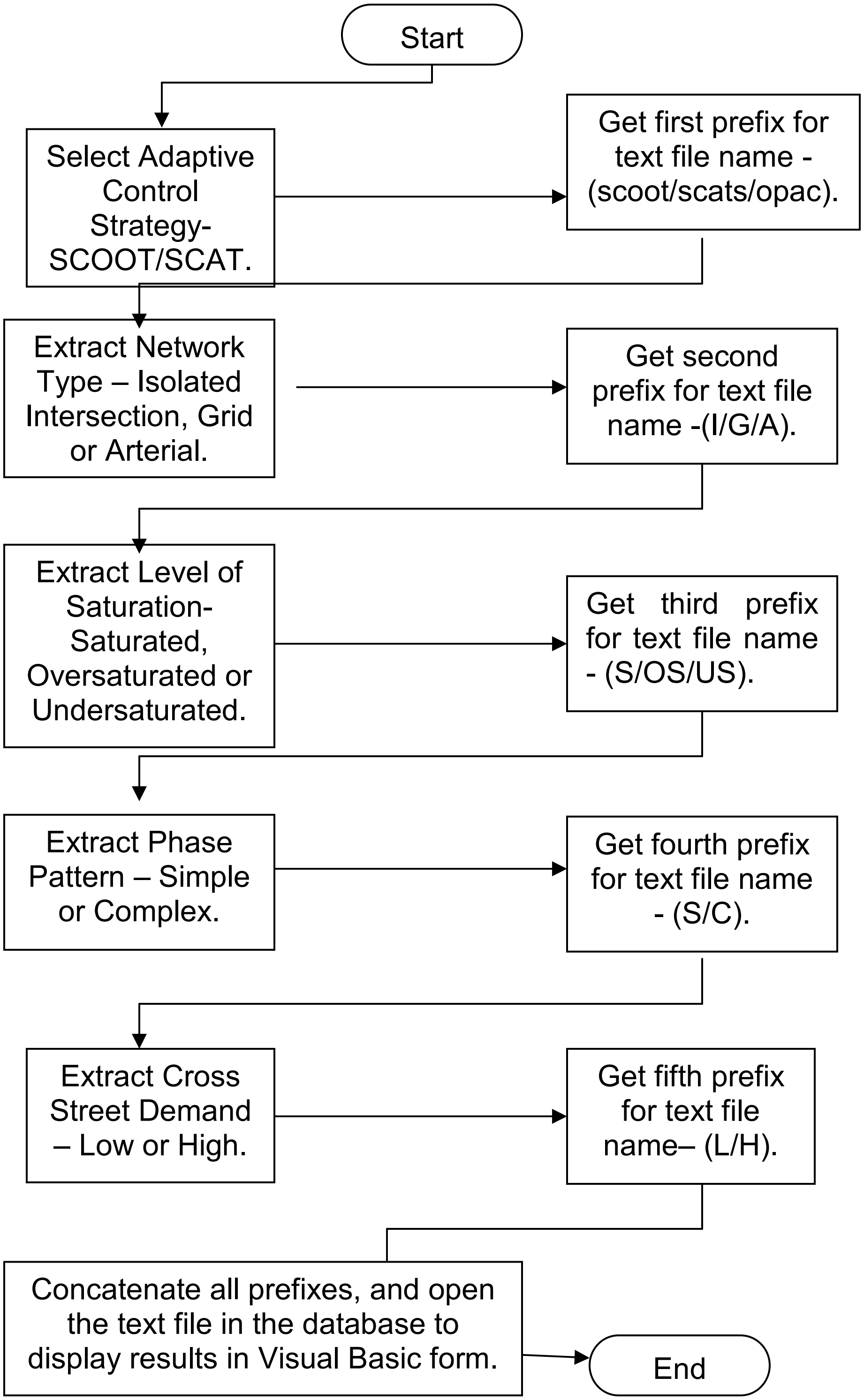


Figure 75 Flowchart used to implement decision tree classifier for expert system

Benefit-Cost Analysis Module

To determine the cost effectiveness of adaptive control algorithms, a cost-benefit analysis module was developed and integrated into the DSS software. This module is capable of running benefit-cost analysis for a period of 20 years. The benefit-cost analysis module calculates the benefit-cost ratio, net present value of the project, and rate of return for the control strategy selected.

Of these three economic indicators, the net present value (NPV) and benefit-cost ratio are important. The NPV is appropriate for comparing the differential economic worth of projects while evaluating project alternatives that result in equal categorical benefits but unequal costs. All benefits and costs over an alternatives' life cycle are discounted to the present, and the costs are subtracted from the benefits to yield an NPV. If benefits exceed costs, the NPV is positive and the project is worth pursuing. NPV is given by the following equation:

$$NPV = -\left[\sum_{t=1}^{T_c} \frac{\dfrac{C}{T_c}}{(1+r)^t}\right] + \sum_{t=Tc}^{T_c+T} \frac{B_t\left[(1+g(t)^{(t-T_c)}\right] + R - M}{(1+r)^t} \tag{21}$$

Where,

C = Total project cost ($)
T_c = Construction period of the project (assumed 4 years for all projects)
T = Expected life time of the project (assumed to be 20 years for all projects)
B_t = Monetary value of benefit (travel time savings, reduction in emissions etc.) ($)
g(t) = growth rate in year *t*
R = fair box revenue
M = annual operating cost ($)
r = interest rate (%)

The NPV formula above has two important parts: The present value of cost and the present value of benefits. The maintenance cost is considered as a negative benefit.

The benefit-cost ratio method is generally used when project funding is restricted. The benefit-cost ratio is obtained by placing the present value of benefits in the numerator and present value of cost in the denominator. The cost includes only the construction cost as in the NPV. The benefit-cost ratio is given by the following equation:

Benefit-Cost Ratio, $$BCR = \frac{\sum_{t=1}^{T_c} \frac{\frac{C}{T_c}}{(1+r)^t}}{\sum_{t=Tc}^{T_c+T} \frac{B_t\left[(1+g(t)^{(t-T_c)}\right]+R-M}{(1+r)^t}} \tag{22}$$

Rate of Return is given by the following equation:

Rate of Return, $$RoR = \frac{NPV}{\sum_{t=1}^{T_c} \frac{\frac{C}{T_c}}{(1+r)^t}} \tag{23}$$

Implementation of Benefit-Cost Analysis Methodology

The first step in implementing the benefit-cost analysis is to make a list of costs and benefits that are expected from the project. A detailed literature study was conducted to find out infrastructure costs associated with adaptive traffic signal control. Infrastructure cost includes:

1) Cost for a dedicated communication line
2) Central software cost
3) Operator hardware cost
4) Cost of local controllers
5) Detector cost
6) Labor cost for maintaining signal timings etc.

A communication line may be a typical leased line, a fiber optic line, or a copper cable line. Each of these lines has a different initial, as well as maintenance costs. Central software generally depends on adaptive control system providers. It may be different for SCOOT, SCATS, and OPAC. Operator hardware generally includes a workstation, a platform for the workstation, and a LAN system etc. This hardware may be installed centrally or at multiple locations depending on the type of adaptive traffic control system being considered. The cost of the local controller may vary depending on whether current controllers need an upgrade or new controllers should replace them. The type of detectors to be used is generally specified by the system providers; however, inductive loops are widely used. Labor cost includes cost of a transportation engineers and signal technicians who are required for signal-timing tuning at different times as well as other labor work required at the signalized junction. The maintenance cost for the infrastructure system is considered as a negative benefit while calculating the NPV of benefits.

SCOOT

1. COMMUNICATION REQUIREMENT								
TYPE			CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
			LOW	HIGH	LOW	HIGH		
			*1000 $	*1000 $	*1000 $	*1000 $		
LEASED LINE	56 Kbps		0.5	1	0.6	1.2	20	Minimum Distance 8-15 miles. Cost is insensitive to distance.
	1.544 Mbps	(T1 line)	0.5	1	4.8	8.4	20	
	44.736 Mbps	(T3 line)	3	5	24	72	20	
COPPER CABLE	Twisted Pair			12		0.02	20	Cost per mile
FIBER OPTIC				20		0.02	20	Cost per mile

2. CENTRAL SOFTWARE						
TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	HIGH	LOW	HIGH	LOW		
	*1000 $	*1000 $	*1000 $	*1000 $		
DEC ALPHA WORKSTATIONS RUNNING Open VMS	30		N.A.		N.A.	Cost as available in reference[2]

3. OPERATOR WORKSTATIONS						
TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	HIGH	LOW	HIGH	LOW		
	*1000 $	*1000 $	*1000 $	*1000 $		
PC RUNNING Windows 95/98/NT/2000	0.5*	1.5*	N.A.	N.A.	N.A.	General cost of PCs
LAN CONNECTED VIA X-Windows EMULATION	40	70	0.4	0.8	20	
REMOTE DIAL-INS VIA TERMINAL SERVER	N.A.					

4. LOCAL CONTROLLERS

TYPE		CAPITAL COST HIGH **1000 $*	CAPITAL COST LOW **1000 $*	O & M COST HIGH **1000 $*	O & M COST LOW **1000 $*	EXPECTED LIFE TIME	COMMENTS
NEW (w / cabinet)	EPAC, 2070	11	17.5	0.2	0.9	N.A.	Cost is per intersection
UPGRADE	Needs additional communication unit & upgrade of controller firmware	2.5	10	11	12	N.A.	Cost is per intersection

5. DETECTORS

TYPE		CAPITAL COST HIGH **1000 $*	CAPITAL COST LOW **1000 $*	O & M COST HIGH **1000 $*	O & M COST LOW **1000 $*	EXPECTED LIFE TIME	COMMENTS
INDUCTIVE LOOP AT INTERSECTION	a	7	5	N.A.		N.A.	As availabe in reference [2]
	b	9	16	1	1.6	N.A.	Four Legs, 2 lanes/approach

References: -

1. Equipement Cost Worksheet from IDAS software

2. Steven Venglar and Thomas Urbanik II, Evolving to Real-Time Adaptive Traffic Signal Control, Texas Transportation Institute, Texas A&M University System

SCATS

1. COMMUNICATION REQUIREMENT

TYPE			CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
			LOW	HIGH	LOW	HIGH		
			*1000 $	*1000 $	*1000 $	*1000 $		
LAN-TCP/IP CONNECTION	56 Kbps		0.5	1	0.6	1.2	20	Minimum Distance 8-15 miles Cost is insensitive to distance
	1.544 Mbps	(T1 line)	0.5	1	4.8	8.4	20	
	44.736 Mbps	(T3 line)	3	5	24	72	20	
COPPER CABLE	Twisted Pair	(300-1200 Bps)		12		0.02	20	Cost per mile

2. CENTRAL HARDWARE

TYPE		CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
		LOW	HIGH	LOW	HIGH		
		*1000 $	*1000 $	*1000 $	*1000 $		
DEC ALPHA WORKSTATIONS RUNNING OPEN VMS	SCATS 1	70	40	N.A.		N.A.	Cost as available in reference [2]
NETWORKED PCs (LAN)*	SCATS 2	40	70	0.4	0.8		

3. CENTRAL SOFTWARE

TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	LOW	HIGH	LOW	HIGH		
	*1000 $	*1000 $	*1000 $	*1000 $		
DATABASE and MANAGEMENT SYSTEMS + CENTRAL S/W	40	70	N.A.		N.A.	Cost as available in reference [2]

4. OPERATOR WORKSTATIONS						
TYPE	**CAPITAL COST**		**O & M COST**		**EXPECTED LIFE TIME**	**COMMENTS**
	LOW	**HIGH**	**LOW**	**HIGH**		
	**1000 $*	**1000 $*	**1000 $*	**1000 $*		
PC RUNNING Windows 95/98/NT/2000	0.5*	1.5*	N.A.	N.A.	N.A.	General PC cost

5. LOCAL CONTROLLERS							
TYPE		**CAPITAL COST**		**O & M COST**		**EXPECTED LIFE TIME**	**COMMENTS**
		LOW	**HIGH**	**LOW**	**HIGH**		
		**1000 $*	**1000 $*	**1000 $*	**1000 $*		
NEW	SCATS 2070, SCATS 2070N	N.A.	N.A.	N.A.	N.A.	N.A.	
UPGRADE	AWA DELTA 170 replaces 170 AWA DELTA 3N replaces NEMA	N.A.	N.A.	N.A.	N.A.	N.A.	
TACTICAL SCATS CONTROLLER	(w / cabinet)	11	17.5	0.2	0.9		Cost per intersection

6. DETECTORS							
TYPE		**CAPITAL COST**		**O & M COST**		**EXPECTED LIFE TIME**	**COMMENTS**
		LOW	**HIGH**	**LOW**	**HIGH**		
		**1000 $*	**1000 $*	**1000 $*	**1000 $*		
INDUCTIVE LOOP AT INTERSECTION	a	7	5	N.A.		N.A.	As available in reference [2]
	b	9	16	1	1.6	N.A.	Four Legs, 2 lanes/approach

References: -

1. Equipement Cost Worksheet from IDAS software

2. Steven Venglar and Thomas Urbanik II, Evolving to Real-Time Adaptive Traffic Signal Control, Texas Transportation Institute, Texas A&M University System

OPAC

1. COMMUNICATION REQUIREMENT

TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	LOW	HIGH	LOW	HIGH		
	**1000 $*	**1000 $*	**1000 $*	**1000 $*		
N.A.						

2. CENTRAL SOFTWARE

TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	LOW	HIGH	LOW	HIGH		
	**1000 $*	**1000 $*	**1000 $*	**1000 $*		
MIST COMUNICATION SERVER, DATABASE SERVER, OI & SERVER	100	200	N.A.		N.A.	Cost as in reference [2]

3. CENTRAL HARDWARE

TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	LOW	HIGH	LOW	HIGH		
	**1000 $*	**1000 $*	**1000 $*	**1000 $*		
100Mb ETHERNET, LOCAL AND DIAL-IN WORKSTATIONS + OTHER	20	50	N.A.	N.A.	N.A.	Cost as in reference [2]

4. LOCAL CONTROLLERS

TYPE	CAPITAL COST		O & M COST		EXPECTED LIFE TIME	COMMENTS
	LOW	HIGH	LOW	HIGH		
	**1000 $*	**1000 $*	**1000 $*	**1000 $*		
ATC CONTROLLERS (2070,2070 lite, NEW 170), NEMA TS2	4	6	N.A.	N.A.	N.A.	

5. DETECTORS						
TYPE	**CAPITAL COST**		**O & M COST**		**EXPECTED**	**COMMENTS**
	LOW	**HIGH**	**LOW**	**HIGH**	**LIFE TIME**	
	**1000 $*	**1000 $*	**1000 $*	**1000 $*		
INDUCTIVE LOOPS AT INTERSECTIONS	9	16	1	1.6	N.A.	Four Legs, 2 lanes/approach

References: -
1. Equipement Cost Worksheet from IDAS software
2. Steven Venglar and Thomas Urbanik II, Evolving to Real-Time Adaptive Traffic Signal Control, Texas Transportation Institute, Texas A&M University System

Considering the above cost elements, benefit-cost analysis can be conducted based on maximum or minimum costs, and hence, results may vary. However, cost for central hardware, software, and communication system may not be easily applicable for isolated intersections. The cost elements should be properly scaled down for performing benefit-cost analysis for isolated intersections.

The next step is to consider benefits that are expected to result from the implementation of the adaptive traffic control systems. Main benefits are reduction in travel time and in emissions. For benefit-cost analysis, monetary value must be assigned to these benefits. The monetary value for reduction in travel time depends on various factors. Emission reductions include reduction in CO_2, CO, and NO_X etc. Monetary values corresponding to reduction in these emissions are available in the IDAS benefits module. (63)

Table 52 Monetary values assigned to ITS benefits (32)

Benefit	Monetary Value
Travel Time Saving	7.6 $/hour
Hydrocarbon Emissions	1,774 $/ton
Nitrous Oxide Emissions	3,731 $/ton
Carbon Monoxide Emissions	3,839 $/ton
Particulates (PM_{10}) Emissions	11,066 $ ton
Carbon Dioxide Emissions	3.56 $/ton
Accident Cost – Fatality (Internal + External Cost)	2,726,350 $
Accident Cost – Injury (Internal + External Cost)	59,718 $
Accident Cost – Property Damage (Internal + External Cost)	3,322$

From macroscopic simulations, MISP, it is relatively easy to get the total time saved at the intersection while comparing it with optimized pre-timed control because they are computed for each intersection internally by the program. However to obtain benefits from reduction in emissions and fatality rates, it is necessary to consider other parameters as well.

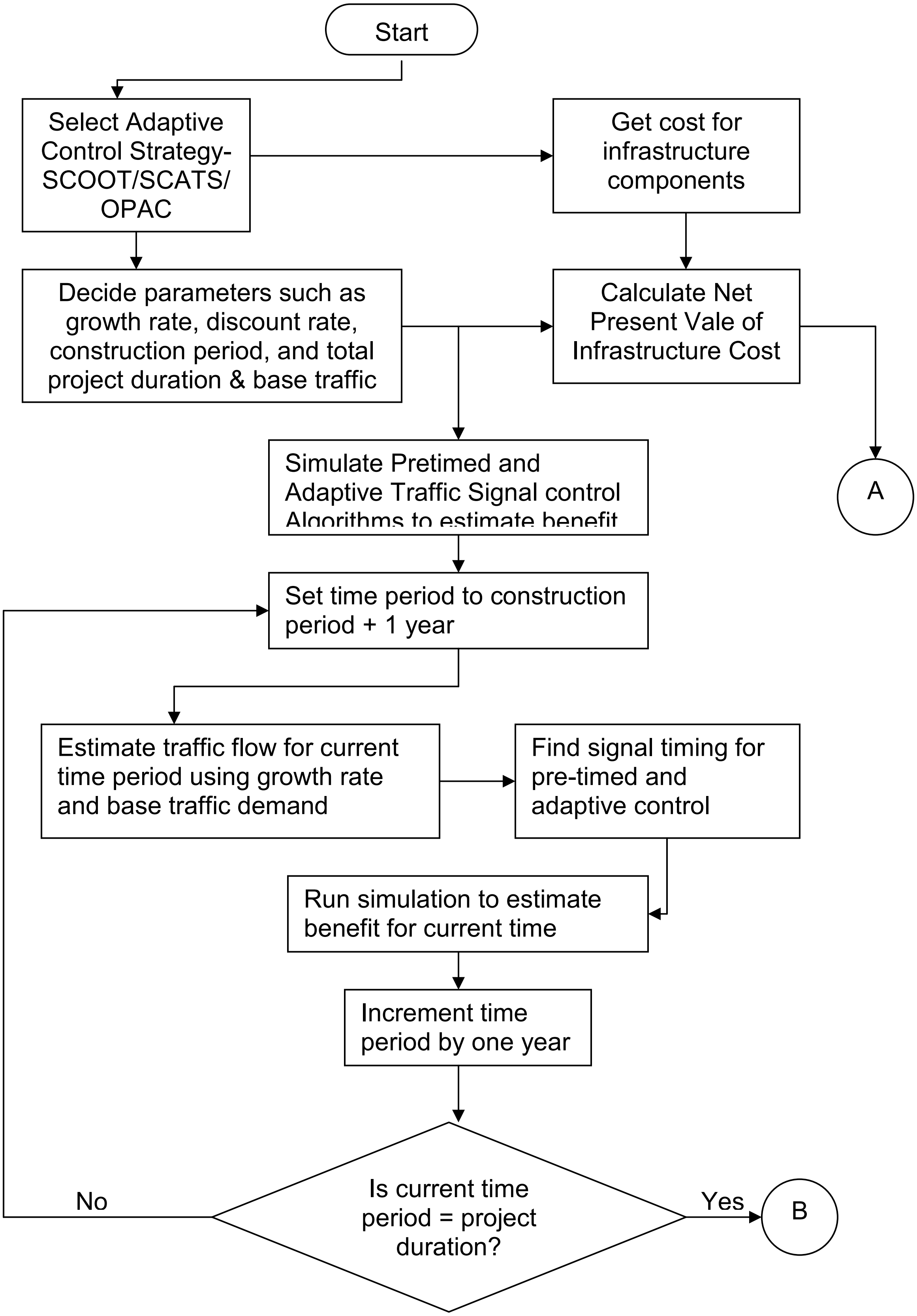
Emissions rates are obtained for each vehicle miles traveled. In macroscopic simulations, it is not possible to get vehicle miles traveled directly. However by multiplying total travel time reduction by average velocity on the route, a measure of reduction in vehicle miles traveled is obtained. This can be used as a measure of benefit to calculate reduction in benefits.

Similarly, accident costs are based on volume-capacity ratio and vehicle miles traveled. IDAS has a calibrated database of accident rate per million vehicle miles as a function of volume-capacity ratio. It is easy to calculate volume to capacity ratio, since signal timing plan and flow are known. Vehicle miles traveled can be obtained from a similar way as discussed above.

One important consideration while evaluating benefits is that during the lifetime of the project duration the traffic demand may vary according to a certain growth rate. Hence, while performing simulations to find reduction in travel time etc. new signal timings should be used for each year. Moreover, the travel demand is not constant throughout the day. Hence, peak and off-peak traffic conditions should be considered, a simple way is to perform a separate set of simulations for peak and off-peak demand for each year. Knowing the peak demand duration during the day, one can estimate the total travel time saving for 365 days for both peak and off-peak hours. Signal timings for each year can be calculated using HCM guidelines.

Other parameters such as growth rate, interest rate, construction time period and total project duration are to be determined before performing the analysis. If standard values are not available, sensitivity analysis can be performed for these parameters.

As soon as all parameters are known, the NPV, benefit-cost ratio and rate of return can be calculated using previously presented formulas. The flow chart in Figure 76 shows the implementation of benefit-cost estimation procedure. This procedure was implemented using C programming language. A GUI was designed using Visual Basic, which for allows easy input of parameters. The program first performs simulation for a specified time interval, updating the signal plan for each step in the time interval for two cases: pre-timed and adaptive control. Reduction in travel time owing to adaptive control is taken as benefit. At this point the program does not consider emission reductions. Results of benefit-cost analysis are discussed in case studies in the next section.


Start
Select Adaptive Control Strategy-SCOOT/SCATS/OPAC
Get cost for infrastructure components
Decide parameters such as growth rate, discount rate, construction period, and total project duration & base traffic
Calculate Net Present Vale of Infrastructure Cost
Simulate Pretimed and Adaptive Traffic Signal control Algorithms to estimate benefit
A
Set time period to construction period + 1 year
Estimate traffic flow for current time period using growth rate and base traffic demand
Find signal timing for pre-timed and adaptive control
Run simulation to estimate benefit for current time
Increment time period by one year
Is current time period = project duration?
No
Yes
B

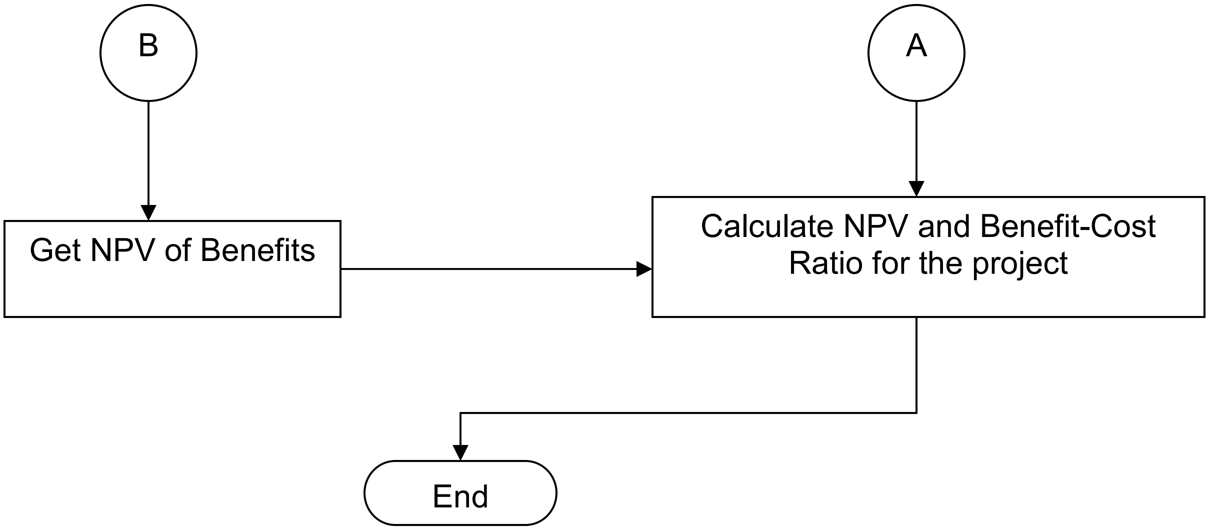


Figure 76 Flowchart to Implement Benefit-Cost Analysis

IMPLEMENTATION OF EXPERT SYSTEM FOR DEVELOPMENT OF SELECTION STRATEGIES

The expert system developed was tested on certain New Jersey Highways. The calibrated networks of Route 10, 18 and 23 that were earlier developed in Synchro were used by the expert system program to test three adaptive control strategies. Isolated intersections on these arterials were selected and adaptive control strategies such as SCOOT-like, and OPAC-like algorithms were simulated. The main aim of these case studies is to know how SCOOT- like, and OPAC-like work under similar network conditions. SCATS-like prototype is not yet available yet, since it needs more validation.

The output of the case studies is a decision, which would be based on two criteria:

- Macroscopic Simulation Results for Adaptive Control Strategy
- Rules of the Knowledge-Based Expert System.

The results will suggest whether adaptive strategies work on given intersection based on prototypes and whether previous implementation results (from knowledge base) also suggest good performance under given network conditions. The case study also compares simulation output for adaptive control with pre-timed simulation. Finally, a benefit-cost analysis was performed. Hence, the “decision” along with simulation output and benefit-cost analysis helps to simplify the selection strategy for adaptive control systems for a given network condition.

Notation:
Total Delay: Total delay for a phase is given in vehicle-hours. Total delay for the intersection is the addition of the total delay of all phases.
Average Delay: Mean Delay is given in seconds-vehicle for each phase.
Signal Timing Plans: Times in signal plans are all in seconds.

Case Study 1

Name of Intersection: Route 23 and Boonton Avenue

Rule Base: Refer to Rule 14 in the rules database on page 151 for the network type and expected system performance.

Traffic Demand:

	Left	Through	Right
Eastbound	0	3695	146
Westbound	0	1016	40
Northbound	303	111	18
Southbound	35	82	137

Signal Timing Plan:

Pretimed Control

	Phase 1	Phase 2
Movements	EB, WB	NB, SB
Green Time	68	24
Amber + All Red	4	4

OPAC-Like Control

	Phase 1	Phase 2
Min. Green Time	40	10
Max. Green Time	105	35
Amber + All Red	5	5

SCOOT-Like Control

	Phase 1	Phase 2
Green Time	68	24
Min. Green Time	40	10
Max. Green Time	105	35
Amber + All Red	5	5

Macroscopic Simulation Result

MOE	Pre-timed	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
Total Intersection Delay	46.61	17.36	20.36	69.54
Total Delay (phase 1)	40.61	7.49	11.58	5.94
Average Delay (phase 1)	15.22	5.62	8.88	31.53
Total Delay (phase 2)	6.0	9.87	8.78	63.60
Average Delay (phase2)	16.67	54.86	47.83	41.67

Benefit-Cost Analysis Result

	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
NPV	114836.00	60208424	-19490498.00
Benefit-Cost Ratio	1.23	79.37	-24.37
Rate of Return	0.23	78.37	-25.37

Analysis

Net-Present Value (NPV) depends on the savings in delays with respect to pre-timed control strategy and other infrastructure (hardware) and operational costs. For example, for this case study, SCATS like strategy increases intersection delays compared to pre-timed control strategy. This, in turn, causes negative NPV for the SCATS-like strategy. According to the macroscopic simulation results we see the OPAC-like prototype gives lower delays compared with pre-timed signal control. However the benefit-cost ratio for OPAC is very low compared with SCOOT-like strategy. The primary reason for this is that the amount of benefit that the OPAC-like prototype gives is not sufficient to cover the infrastructure and maintenance cost for this control strategy. Detailed analysis of the simulation results showed that the operations and maintenance cost of OPAC-like strategy are much higher compared with SCOOT-like. This intersection has about 12 lanes and OPAC-like strategy needs four detectors to estimate arrivals and queue per lane. Hence the number of detectors needed in OPAC increases four times compared to SCOOT-like strategy. This is a primary reason for lower benefit-cost ratio for the OPAC-like strategy.

Case Study 2

Name of Intersection: Route 23 and Echo-lake Road
Rule Base: Refer to Rule 16 in the rules database on page 152 for the network type and expected system performance.

Traffic Demand:

	Left	Through	Right
Eastbound	0	2879	81
Westbound	0	503	98
Northbound	21	81	6
Southbound	416	25	44

Signal Timing Plan:

Pre-timed Control

	Phase 1	Phase 2
Movements	NB, SB	EB, WB
Green Time	28	64
Amber + All Red	4	4

OPAC-Like Control

	Phase 1	Phase 2
Min. Green Time	10	35
Max. Green Time	35	105
Amber + All Red	5	5

SCOOT-Like Control

	Phase 1	Phase 2
Green Time	28	64
Min. Green Time	10	35
Max. Green Time	35	105
Amber + All Red	5	5

Macroscopic Simulation Result

MOE	Pre-timed	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
Total Intersection Delay	211.30	94.64	165.00	254.69
Total Delay (phase 1)	5.34	74.11	10.47	5.68
Average Delay (phase 1)	16.75	464.80	66.06	35.20
Total Delay (phase 2)	205.96	20.53	154.53	249.01
Average Delay (phase2)	104.63	20.86	163.53	255.83

Benefit-Cost Analysis Result

	OPAC-LIKE	SCOOT-LIKE
NPV	15111710	56378941
Benefit-Cost Ratio	20.67	60.89
Rate of Return	19.67	59.89

Analysis

In this case study, the total number of lanes at the intersection is 8 compared to 12 at the intersection of NJ Route 23 and Boonton Avenue. However, the traffic demand per lane at this intersection is lower compared to NJ Route 23 and Boonton Avenue. The OPAC-like strategy gives a higher benefit-cost ratio at this intersection compared to Boonton Avenue mainly because there is less infrastructure as well as operations and maintenance costs associated with this intersection (due to reduced number of lanes). Benefit-cost ratio of SCOOT-like strategy is still higher when compared with the OPAC-like strategy.

Case Study 3

Name of Intersection: Route 10 and Ridgedale Avenue

Rule Base: Refer to Rule 16 in the rules database on page 152 for the network type and expected system performance.

Traffic Demand:

	Left	Through	Right
Eastbound	0	1198	413
Westbound	0	1049	17
Northbound	211	256	56
Southbound	308	325	94

Signal Timing Plan:

Pre-timed Control

	Phase 1	Phase 2
Movements	NB, SB	EB, WB
Green Time	29	38
Amber + All Red	4	4

OPAC-Like 3 Control

	Phase 1	Phase 2
Min. Green Time	15	25
Max. Green Time	45	75
Amber + All Red	5	5

SCOOT-Like Control

	Phase 1	Phase 2
Green Time	29	38
Min. Green Time	15	25
Max. Green Time	45	75
Amber + All Red	5	5

SCATS-Like Control

	Phase 1	Phase 2
Plan 1 (Green)	29	38
Plan 2 (Green)	25	32
Plan 3 (Green)	34	43
Plan 4 (Green)	38	49
Amber + All Red	4	4

Macroscopic Simulation Result

MOE	Pretimed	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
Total Intersection Delay	23.92	13.13	97.57	53.91
Total Delay (phase 1)	6.48	6.58	15.66	5.11
Average Delay (phase 1)	9.83	19.98	47.25	14.86
Total Delay (phase 2)	17.44	6.55	81.91	48.80
Average Delay (phase2)	11.55	8.67	114.78	67.06

Benefit-Cost Analysis Result

	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
NPV	2193906	30457896	-24179418
Benefit-Cost Ratio	5.52	65.21	-48.89
Rate of Return	4.52	64.21	-49.89

Analysis

This intersection has a higher cross street demand compared to the other two previously analyzed intersections. The number of lanes at this intersection is the same as compared to NJ State Highway 23 and Echo Lake Road. However, owing to higher cross-street demand, the benefit-cost ratio for the OPAC-like strategy has reduced. SCATS-like prototype produced negative benefit-cost ratio for this case study. For cases similar to case study-3 where OPAC-like strategy reduced delays compared to pre-timed control strategy but still low or negative B/C ratios are obtained, there was not sufficient reduction of delays to justify the use of this adaptive control strategy. In other words, the benefits due to the delay reduction could not offset the cost of hardware and maintenance required to deploy and operate this strategy. OPAC-like strategy performed better than SCOOT-like strategy for intersections with a higher cross street demand and hence its benefit-cost ratio is the highest.

Case Study 4

Name of Intersection: Route 18 and Eggers/S.Woodland Street

Rule Base: Refer to Rule 14 in the rules database on page 151 for the network type and expected system performance.

Traffic Demand:

	Left	Through	Right
Eastbound	0	60	62
Westbound	61	44	51
Northbound	0	3781	0
Southbound	0	1765	153

Signal Timing Plan:

Pre-timed Control

	Phase 1	Phase 2
Movements	NB, SB	EB, WB
Green Time	100	20
Amber + All Red	4	4

OPAC-Like 3 Control

	Phase 1	Phase 2
Min. Green Time	75	10
Max. Green Time	110	30
Amber + All Red	5	5

SCOOT-Like Control

	Phase 1	Phase 2
Green Time	100	20
Min. Green Time	75	10
Max. Green Time	110	30
Amber + All Red	5	5

SCATS-Like Control

	Phase 1	Phase 2
Plan 1 (Green)	62	12
Plan 2 (Green)	71	13
Plan 3 (Green)	79	15
Plan 4 (Green)	87	17
Amber + All Red	4	4

Macroscopic Simulation Result

MOE	Pre-timed	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
Total Intersection Delay	10.47	7.21	83.71	12.2
Total Delay (phase 1)	6.41	2.46	15.23	4.44

Average Delay (phase 1)	2.86	2.20	14.05	4.00
Total Delay (phase 2)	4.06	4.75	68.48	7.76
Average Delay (phase2)	23.43	54.85	856.02	6.50

Benefit-Cost Analysis Result

	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
NPV	-1097466	-1568961	-2273187.50
Benefit-Cost Ratio	-1.02	-8.96	-3.19
Rate of Return	-2.02	-9.96	-4.19

Analysis

The main reason for the negative benefit-cost ratio for the OPAC-like prototype is the fact that there are 10 lanes at this intersection. The cross-street demand at this intersection is very low. Travel time saving with the OPAC-like prototype for such intersections is not high enough to cover its maintenance costs. At a low cross-street demand, SCOOT-like strategy performance also deteriorates and the benefit-cost ratio for this case is even less compared to OPAC-like strategy. Hence, it can be concluded that SCOOT-like strategies are not as cost effective compared to the OPAC-like strategy when the cross-street demand is very low. Similar behavior is observed for the SCATS-like strategy.

Case Study 5

Name of Intersection: Doremus Avenue and Route 23

Rule Base: Refer to Rule 14 in the rules database on page 151 for the network type and expected system performance.

Traffic Demand:

	Left	Through	Right
Eastbound	0	1728	0
Westbound	0	432	5
Northbound	6	2	2
Southbound	37	0	9

Signal Timing Plan:

Pre-timed Control

	Phase 1	Phase 2
Movements	NB, SB	EB, WB
Green Time	15	35
Amber + All Red	5	5

OPAC-Like 3 Control

	Phase 1	Phase 2
Min. Green Time	5	20
Max. Green Time	30	70
Amber + All Red	5	5

SCOOT-LIKE Control

	Phase 1	Phase 2
Green Time	15	35
Min. Green Time	5	30
Max. Green Time	20	70
Amber + All Red	5	5

SCATS-Like Control

	Phase 1	Phase 2
Plan 1 (Green)	12	30
Plan 2 (Green)	16	36
Plan 3 (Green)	19	43
Plan 4 (Green)	22	50
Amber + All Red	4	4

Macroscopic Simulation Result

MOE	Pre-timed	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
Total Intersection Delay	4.81	1.47	3.31	5.64
Total Delay (phase 1)	0.3	0.4	0.33	0.12
Average Delay (phase 1)	10.0	28.15	28.05	8.50
Total Delay (phase 2)	4.51	1.07	2.98	5.52
Average Delay (phase2)	3.83	1.82	5.17	9.30

Benefit-Cost Analysis Result

	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
NPV ($)	-108131	201563	-1472945
Benefit-Cost Ratio	-1.23	0.36	-2.03
Rate of Return	-2.23	-1.36	-3.03

Analysis

OPAC-like strategy reduced delays compared to pre-timed control strategy but still low or negative B/C ratios are obtained because there was not sufficient reduction of delays to justify the use of this adaptive control strategy. In other words, the benefits due to the delay reduction could not offset the cost of hardware and maintenance required to deploy and operate this strategy. In brief, this intersection has a very low cross-street demand, but the number of lanes at this intersection is 8. In this case, prototypes of all both control strategies are not cost-effective.

Case Study 6

Name of Intersection: Walnut Street and Route 10

Rule Base: Refer to Rule 14 in the rules database on page 151 for the network type and expected system performance.

Traffic Demand:

	Left	Through	Right
Eastbound	0	770	256
Westbound	0	1651	4
Northbound	371	16	72
Southbound	90	53	1

Signal Timing Plan:

Pre-timed Control

	Phase 1	Phase 2
Movements	NB, SB	EB, WB
Green Time	15	35
Amber + All Red	5	5

OPAC-Like 3 Control

	Phase 1	Phase 2
Min. Green Time	5	15
Max. Green Time	20	70
Amber + All Red	5	5

SCOOT-Like Control

	Phase 1	Phase 2
Green Time	15	35
Min. Green Time	5	15
Max. Green Time	20	70
Amber + All Red	5	5

SCATS-Like Control

	Phase 1	Phase 2
Plan 1 (Green)	17	35
Plan 2 (Green)	14	28
Plan 3 (Green)	20	42
Plan 4 (Green)	24	48
Amber + All Red	4	4

Macroscopic Simulation Result

MOE	Pre-timed	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
Total Intersection Delay	9.58	7.80	12.40	9.47
Total Delay (phase 1)	4.49	4.73	4.37	2.69
Average Delay (phase 1)	14.05	29.59	27.00	16.36
Total Delay (phase 2)	5.09	3.07	8.13	6.78
Average Delay (phase2)	3.57	4.31	11.38	9.23

Benefit-Cost Analysis Result

	OPAC-LIKE	SCOOT-LIKE	SCATS-LIKE
NPV ($)	114836	-112341	-740981
Benefit-Cost Ratio	1.23	-6.57	-0.53
Rate of Return	0.23	-7.57	-1.53

Analysis

This intersection has medium cross-street demand compared with the main street. The number of lanes at this intersection is 8. In such a case, we see that OPAC-like strategy is the most cost-effective adaptive control strategy. The SCOOT-like and SCATS-like strategies both have strategy has a negative benefit-cost ratio. Also, it is observed that the benefit-cost ratio for OPAC-like strategy is very close to 1, which suggests that this strategy is not very cost-effective.

CONCLUSIONS

The development of the decision support tool helps with easy analysis of the adaptive signal strategies when applied to various types of intersections. The tool allows testing OPAC-Like and SCOOT-Like prototypes on NJ highways through a GIS-based interface. A working prototype of the decision support system software was developed and tested. Expert system module of this prototype DSS tool helps the user to utilize past implementation results of adaptive control strategies to evaluate intersection that is being studied. This offers a good indication of what to expect from adaptive control strategies for a given transportation network.

The knowledge-based expert system coupled with a macroscopic simulation tool that can simulate all three control strategies proved to be a useful evaluation tool. On the other hand, multiple runs of the microscopic simulation can be quite time consuming, and in general a single run for one hour of simulation requires more than 15 minutes. An important feature of the prototype DSS is its ability to communicate with the standard traffic signal optimization toll namely Synchro. This enables the user to make changes to intersection characteristics using Synchro and then export the updated intersection to the prototype DSS. Similar modifications can also be made using the DSS tool and exported to SYNHCRO. Thus, Synchro and the developed prototype are efficiently connected using a custom software application developed as part of this project.

This new capability that connects Synchro to GIS is unique and very useful when dealing with multiple routes that are geographically dispersed over the entire State of New Jersey. The user can simply use the GIS map to locate the route section and the intersection on it to conduct the analysis. The benefit-cost analysis module of the developed DSS helps to evaluate the cost-effectiveness of these systems.

Finally, a detailed analysis of adaptive control strategies on selected NJ highways is conducted. This analysis highlights the effectiveness of adaptive signal strategies under different types of traffic networks. The benefit-cost analysis gives a detailed performance review of these systems in terms of benefit-cost ratio, net present value and rate-of-return.

Major Findings: Adaptive Control features

1. For adaptive control strategies using Optimization techniques, such as OPAC, it is necessary to control the cycle length efficiently. Optimization tends to terminate the cycle length abruptly or may lengthen it to an unreasonable duration, which may not be desired for real-world

conditions. For reactive control strategies, such as SCOOT, the cycle length transition is in fixed steps, and at times it can be too slow. The SCATS-like strategy has a selective algorithm that chooses the best cycle length for the current traffic demand, but for the SCOOT-like strategy, the transition to a new cycle length can sometimes be slow when traffic conditions are changing too rapidly. The SCATS-like strategy has a selective algorithm that chooses the best cycle length for the current traffic demand, but for the SCOOT-like strategy, the transition to a new cycle length can sometimes be slow when traffic conditions are changing too rapidly. Performance of the SCATS-like strategy highly depends on the pre-determined plans and the results presented on this report are a function of these plans. For all the case studies presented in this report, SCATS like strategy produced "total intersection delays" that are higher than the delays due to the pre-timed control strategy. This, in turn, generated negative Net-Present Values and negative B/C ratios for all case studies of SCATS-like strategy. However, it is possible to improve the performance of the SCATS-like strategy by generating more effective pre-determined plans. This is left as a future research objective. For cases where OPAC-like strategy reduced delays compared to pre-timed control strategy butt still low or negative B/C ratios are obtained, there was not sufficient in the reduction of delays to justify the use of this adaptive control strategy. In other words, the benefits due to the delay reduction could not offset the cost of hardware and maintenance required to deploy and operate this strategy.

2. It is difficult to see a consistent trend in the performance of prototype adaptive signal algorithms for different intersections. The intersections were selected to cover a broad range of intersection and network types. The results have effects of each parameter such as cross-street demand, network geometry, and level of saturation etc. For example, if there is a jug-handle turn at an intersection, it is difficult to install more detectors along the jug-handle and predict future arrivals during the head period. For the (predictive) OPAC-like algorithm that uses future arrivals at an intersection, the shorter link length on jug-handles does not permit more detector placements. In such a case, it is possible only to predict the queue on the intersection, and the future arrivals are estimated even for the head period of the horizon. Hence, there is less information needed for dynamic programming model and it can be expected that the OPAC-like prototype will act differently for this type of intersections compared with intersections where there is enough space on the approaches to accommodate more detectors. For the reactive SCOOT-like algorithm, the detector would be placed at the entrance of the jug-handle turn. For cases where cross street demand is less, the detector at the entrance of the jug-handle does not remain occupied for a longer duration. Hence the reactive algorithm would give higher green time for the main street, resulting in more stopped time for vehicles on the jug-handle. This results in higher delay for the SCOOT-like prototype. For the reactive/case-based SCATS-

like algorithm, the detector at the stop line would mostly remain occupied; however, the algorithm would not identify if the queue on jug-handle spills on to the main street. This would affect the travel time on the main street and there will be reduction in improvement in the performance index. Nevertheless, the proactive algorithm (OPAC-like), uses a good queue estimate algorithm, would get the best estimate of number of vehicles on the jug-handle and could act to reduce the queue.

3. Similarly, the number of phases also has an impact on the performance of adaptive signal control prototypes. For such intersections, SCOOT-like and SCATS-like prototypes fail to generate benefits at higher volume-to-capacity ratio. However, the OPAC-like prototype gives a consistent improvement in PI, because it can respond to varying traffic demand, by changing the cycle length quickly. SCOOT-like and SCATS-like prototypes may be slower to react because change in cycle length is accomplished in small increments at regular intervals of about 300 seconds.
4. With higher cross street demand, SCOOT-like, SCATS-like and OPAC-like prototypes fail to generate higher benefits at higher volume to capacity ratio.
5. In infrastructure facilities, systems using the optimization procedure need more detectors. This, in turn, would lead to higher maintenance and operations costs compared with other systems. The computational needs of adaptive systems using the optimization procedure also increase. But when in place, such systems would be easy to maintain and would need less updates and tuning during their lifetime. On the other hand, systems such as SCATS, that use off-line stored signal plans, which might need periodic updating and frequent tuning. SCOOT-like strategies on the other hand have modest infrastructure and computational needs. SCOOT-like and SCATS like strategies use the volume-occupancy ratio, which is calculated from detectors, whereas OPAC-like strategies use vehicle counts. Hence, OPAC like systems are more prone to detector errors than SCOOT-like and SCATS-like systems.

At this stage, it is difficult to comment on cost-effectiveness of adaptive control systems studied in this report. These systems are difficult to implement and they would not always flawlessly perform in saturated or over-saturated networks (where they are really needed).

In short, the developed tool provides the traffic engineers and decision makers with a user-friendly suite of tools to guide NJDOT engineers in identifying the most suitable intersections for adaptive control and accurately assessing their potential benefits over the existing control.

REFERENCES

1. ITS Decision Website URL: http://www.path.berkeley.edu/~leap/TTM/Traffic_control/traffic_signal_report.htm accessed on 03/03
2. URL:http://www.path.berkeley.edu/itsdecision/serv_and_tech/Traffic_signal_control/trafficsig_report.html accessed on 03/03
3. N. Gartner, S. Chronis, and T. Phillip , "Development of Advanced Traffic Signal Control Strategies for ITS," Transportation Research Record 1494, 1996.
4. L. E. Owen, C. Stallard, and Y. Zhang, "Laboratory Evaluation of RT-TRACS Prototype Strategies," FHWA report prepared by ITT Industries Inc., Systems Division, PO 15012, Colorado Springs, CO 80935, March 31, 1999.
5. P.B. Hunt, D. I. Robertson, and R. D. Brethorton, and R. I. Winton, "SCOOT-A Traffic Responsive Method of Coordinating Signals", TRRL Report LR1014, 1981.
6. URL: http://signalsystems.tamu.edu/documents/TRBWorkshop2000/SCOOT_trb2000part2.ppt accessed on 03/03
7. P. T. Martin, and R. Disengi, Adaptive Traffic Signals Control Systems Survey, TRB Signal Systems Committee, Report UTL – (07/02). URL: http://signalsystems.tamu.edu/documents/AdaptiveSignalsSurvye_Martin_Disengi.pdf accessed on 03/03
8. URL: http://www.scoot-utc.com/whatis.htm accessed on 03/03
9. P. R. Lowrie, "The Sydney Coordinated Adaptive Traffic system Principles, Methodology, Algorithms," IEE Conference, 207, 1982, 67-70.
10. URL: http://signalsystems.tamu.edu/documents/TRBWorkshop2000/SCATS_trb2000pres2.ppt accessed on 03/03
11. N. Gartner, "OPAC: A Demand-Responsive Strategy for Traffic Signal Control," Transportation Research Record, 906,1983, pp 75-81.
12. URL: http://signalsystems.tamu.edu/documents/TRBWorkshop2000/OPAC_Trb2000.ppt accessed on 04/03
13. P. Mirchandani, and L. Head, "RHODES : A Real Time Traffic Signal Control System, Architecture Algorithm and Analysis," URL: http://www.sie.arizona.edu/ATLAS/docs/TRISTANIII.pdf accessed on 04/03
14. URL: www.signalsystems.tamu.edu/documents/Jan2001AnnualMeeting/PituMirchandaniRHODES_TRB2001.pdf accessed on 04/03
15. URL: http://signalsystems.tamu.edu/documents/Jan2001AnnualMeeting/HangHu-LAATCS_TRB2001.pdf accessed on 04/03

16. S. Venglar, and T. Urbanik, "Evolving to Real-Time Adaptive Traffic Signal Control," Steps Forward: Proceedings of the Second World Congress on ITS: '95 Yokohama, SHRP-H-358, VERTIS, Japan, November 1995.
17. URL: http://ai-depot.com/Tutorial/RuleBased.html, Accessed on 03/05.
18. I. Day, "SCOOT-Split, Cycle and Offset Optimization Technique" Transportation Research Board,TRB Committee A3A18 Traffic Signal Systems TRB Mid-year Meeting and Adaptive Traffic Signal Control Workshop, July 12-14,1998.
19. R.D. Bretherton, K. Wood and G.T. Bowen, " SCOOT Version 4 " 1998. 9th International Conference on (Conf. Publ. No. 454), 21-23 Apr 1998 pp.104 –108.
20. F. Middelham and H. Taale, J. Dibbits, W. Fransen, "The Assessment of the SCOOT System in Nijmegen" Road traffic monitoring and control, 23-25 April 1996, conference publication No. 422, IEE ,1996
21. R.D. Bretherton, "SCOOT Information", UK Developments in Road Traffic Signaling, IEE Colloquium, 5 May 1988, pp 11/1 -11/3.
22. Dennis I. Robertson and R. D. Bretherton, "Optimizing Network of Traffic Signals in Real Time-The SCOOT Method" IEEE transaction on vehicular Technology, vol. 40, No. 1, February 1991.
23. M. C. Bell and P. T. Martin, "Use of Detector Data for Traffic Control Strategies" Road Traffic Control, 1990., Third International Conference on , 1-3 May 1990, pp.86 –89.
24. J.D. Hoffman, F.S., Turner "Data and Route Guidance", Third International Conference on Road Traffic Control, pp. 16 -22, May 1-3 1990
25. F. Middleham, J. Dibbits and W. Fransen, "The Implementation of SCOOT in the City of Nijmegen in the Netherlands", 1994.
26. J. Perrin, P. T. Martin and B. G. Hansen, "Connecting SCOOT to CORSIM: Real-Time Signal Optimization Simulation" Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE, Vol.3 pp. 1898 -1903, 2001.
27. H. Taale, W. C. M. Fransen J. Dibbits, "The Second Assessment of the SCOOT System in Nijmegen", Road Transport Information and Control, 1998. 9th International Conference on (Conf. Publ. No. 454), pp. 109 – 113, April 21-23 1998.
28. A. Ash, "Incident Detection in Urban Areas Controlled by SCOOT, Incident Detection and Management", (Digest No: 1997/123), IEE Colloquium on, 2 Jun 1997, Page(s): 8/1 -8/5.
29. F. Boillot, J.M. Blosseville, J.B. Lesort, V. Motyka, M. Papageorgiou, S. Sellam, "Optimal Signal Control of Urban Traffic Networks"**.** Road Traffic Monitoring, (IEE Conf. Pub. 355), 1992.
30. D. Bretherton, "Current Developments in SCOOT: Version 3", Transportation Research Record 1554, 2000.
31. B. G. Hansen, P. T. Martin, and H. J. Perrin, Jr., "SCOOT Real-time Adaptive Control in a CORSIM Simulation Environment", Transportation Research Record 1727, 2000.

32. N. Rouphail, A. Tarko and J. Li, "Signalized Intersections," Draft Chapter 9 of *Traffic Flow Theory Monograph*, January 1996.
33. Highway Capacity Manual, 2000.
34. G. F. Newell. “The Oversaturated Isolated Intersection”, Proceedings of the 15th International Symposium on Transportation and Traffic Theory *International Symposium on Transportation and Traffic Theory (ISTTT),* (Editor: Michael A.P. Taylor), 2002, Pergamon Press.
35. T. Altiok and B. Melamed, “Simulation Modeling and Analysis With Arena”, Cyber Research, Inc. and Enterprise Technology Solutions, Inc., 2001.
36. B. Friedrich. “Adaptive Signal Control: An Overview”. http://www.iasi.rm.cnr.it/ewgt/13conference/102_friedrich.pdf. Accessed on 04/03.
37. P. T. Martin, Joseph Perrin, “Adaptive Signal Control II”, January 2003.
38. N. Gartner, “OPAC: A demand-Responsive Strategy for Traffic Signal Control”. Transportation Research Record 906, TRB, Washington, D.C, 1983.
39. A.G. Sims, A. B. Finlay, “SCATS - Splits and Offsets Simplified (S.O.S)”, ARRB Proceedings Vol. 12, Part 4, 1984
40. J. Y. K. Luk, A. G. Sims and P. R. Lowrie. “SCATS-Application and Field Comparison with a Transyt Optimised Fixed Time System”, International Conference on Road Traffic Signalling, London, 1982.
41. B. Wolshon, W. C.Taylor, “Analysis of Intersection Delay Under Real-time Adaptive Signal Control”. Transportation Research Part C, pp. 53-72, 1999.
42. R.D.Bretherton and G .I. Rai, “The Use of SCOOT in Low Flow Conditions”., Traffic Engineering and Control, December, 1982.
43. R.D. Bretherton, “ SCOOT Urban Traffic Control System-Philosophy and Evaluation” IFAC Symposium on Control, Communications in Transportation, September 1989, Published by Pergamon Press, Oxford.
44. PARAMICS manual, 2004.
45. “LADOT’s Adaptive Traffic Control System (ATCS)”. Presentation at the TRB Workshop on Adaptive Traffic Signal Control Systems, Sponsored by Signal Systems Committee. January 9, 2000.
46. P. Mirchandani, L. Head. “RHODES: A Real-Time Traffic Signal Control System: Architecture, Algorithms, And Analysis”. URL: www.sie.arizone.edu/ATLAS/Docs/TRISTANIII.pdf. Accessed on 04/03
47. P. T.Martin and R. Disegni, “Adaptive Traffic Signals Control Systems Survey”, TRB Signal Systems Committee, 2002.
48. P. T. Martin, “SCATS, an Overview,” TRB Annual Meeting, 2001.
49. B. Melamed, “An Overview of TES Processes and Modeling Methodology. Performance Evaluation of Computer and Communications System,” (L.Donatiello and R. Nelson, Eds.), pp.359--393, Lecture Notes in Computer Science, Springer-Verlag, 1993.
50. R. E. Allsop, “Delay at a Fixed Time Traffic Signal-I: Theoretical Analysis,” Transportation Science 6, pp.260-285, 1972.

51. T. Altiok, B. Melamed. Simulation Modeling and Analysis with Arena. Cyber Research, Inc. and Enterprise Technology Solutions, Inc., 2001.
52. J.R. Hill, and B. Melamed, “TEStool, A Visual Interactive Environment for Modeling Autocorrelated Time Series. Performance Evaluation,” Vol. 24, No. 1\&2, pp.3 - 22, 1995.
53. B. Melamed, “The Empirical TES Methodology: Modeling Empirical Time Series” Journal of applied Mathematics and Stochastic Analysis, Vol. 10 No. 4, pp.333-353, 1997.
54. C. F. Daganzo, Fundamentals Transportation and Traffic Operations, Pergamon Press Inc., Elmsford, 1997.
55. W.F. Adams, “Road Traffic Considered as Random Series,” Journal of the Institution of Civil Engineers, 1, pp.325-62, 1952.
56. G.F. Newell, “Statistical Analysis of the Flow of Highway Traffic through a Signalized Intersection,” Q. Applied Math, Vol.13, pp.353-69, 1956.
57. McNeil, “The Delay at a Traffic Signal,” Dept. Of Statistics Technical Paper No. 56, Her Majesty’s Stationery Office, London, 1966.
58. A.J. Miller, “Settings for Fixed Cycle Traffic Signals,” Operational Research Quarterly, No.14, pp.373-386, 1963.
59. M. Beckman, C.B. McGuiree, and C.B. Winsten, Studies in the Economics of Transportation, Yale University Press, New Haven, Ct., 1956.
60. K. Ozbay, S. Goel, and H. Doshi, “Development of Object Oriented-based Programming Approach for API coding in PARAMICS,” Working Paper, 2005.
61. A. M. Law, and W. D. Kelton, Simulation Modeling and Analysis, 2nd Edition, McGraw-Hill, Inc., 1991.
62. IDAS Software - Equipment Cost and Benefit Module Worksheets http://idas.camsys.com/userManual/Sect_4.pdf and http://idas.camsys.com/userManual/Sect_5.pdf accessed on 04/03.
63. Federal Highway Administration, “ITS Benefits” ITS Benefits and Operational Test Results.” FHWA, 1997, Washington, D.C.
64. N. Gartner, C. Stamatiadis, and P. Tarnoff “Development of Advanced Traffic Control Strategies of Intelligent transportation Systems – A Multi-Level Design. Transportation Research Board, 79th Annual Meeting, January 1995, Washington D.C.
65. R.A. Glasco, et al., “Studies of Potential Intelligent Transportation Systems Benefits Using Traffic Simulation Modeling.” Mitretek Systems, 1996, Washington, D.C.
66. B.G. Hansen, P.T. Martin, and H.J. Perrin, “SCOOT Real-Time Adaptive Control in a CORSIM Simulation Environment.” Transportation Research Board, 79th Annual Meeting, January 9-13, 2000, Washington D.C.
67. J.E. Moore, R. Jayakrishnan, and M.G. McNally, C.A. MacCarley, “SCOOT Performance in Anaheim Advanced Traffic Control System.” Intellimotion, Vol. 8 No.3, 1999.
68. NJDOT Traffic Operations,“ ITS Strategic Business Plan- Draft.” New Jersey Department of Transportation, Trenton, NJ, 1998.

69. K. Ozbay and P. Kachroo, “Wide-Area Incident Management System on the Internet” Proceedings of the 1998 IEEE Conference on Systems, Man, and Cybernetics, San Diego, October 1998.
70. T.L. Saaty “Analytical Hierarchy Process.” McGraw-Hill, 1980.
71. T.L. Saaty “How to Make a Decision: The Analytical Hierarchy Process.” *Interfaces*, 24(6), pp.19-34, 1994.
72. URL: mapquest.com, accessed on 04/03/03.
73. R. E. Levitt, Y. Jin, and C. L. Dym, “Knowledge-based support for management of concurrent, multidisciplinary design,” Artificial Intelligence Eng. Design, Analysis Manuf, vol. 5, no. 3, pp. 77-95, 1991.

APPENDIX 1 USER MANUAL FOR DECISION SUPPORT SYSTEM TOOL

USER MANUAL FOR EXPERT SYSTEM PROGRAM

This manual intends to make a user familiar with the adaptive traffic control software. The manual is divided in to three parts. The first part describes the installation procedure for the program. The second part helps the user to get familiar with all the modules in the program and how to use them. The third part describes changing of default values used in the program.

This program runs macroscopic simulation for adaptive traffic control system. The adaptive traffic control systems are based on prototypes developed at Rutgers University, and may differ from actual control strategies available in the field.

Installation of the Application

Run the setup program from the CD

Select the path to install MGSMProject in any client computer
Install all the necessary files in this directory
Setup program will create a folder named MGSMProject for the selected path

Setup an Environmental variable MGSM_Dir

- Goto Control panel → Systems → Environmental Variables
- Enter the following:
  Variable: MGSM_Dir
  Value: Path under which MGSMProject has been installed, i.e. C:\Program Files\MGSMProject

Setup ODBC Connection

- Goto Control panel → Data Sources (ODBC)
- Add new user data source
  Select Microsoft Access Driver (*.mdb)

  Data Source name: projInter

  Select the database as the whole path to MGE_Gis_Export.mdb database. This database is stored in Path to MGSMProject\MGSMProject\Databases folder. If MGSMProject folder is stored in C:\Program files, then select C:\Program Files\MGSMProject\Databases\MGE_Gis_Export.mdb database

  Data Source name : Rutgers

  Select the database as the whole path to MGE_Gis_Export.mdb database. This database is stored in Path to MGSMProject\MGSMProject\Databases folder. If MGSMProject folder is stored in C:\Program Files, then select C:\Program Files\MGSMProject\Databases\MGE_Gis_Export.mdb database

Restart the computer to activate the system variables

Register the GeoMedia Command

- Copy abcd.dll and abcd.ini from MGSMProject\Databases to the Program folder of GeoMedia Professional
- Open the command prompt
- Go to the directory of Program folder of GeoMedia Professional, i.e. C:\Program Files\GeoMedia Professional\Program
- Install the GeoMedia command by using the statement as the following:

```
Installusrcmd /prod "GeoMedia Professional" abcd.dll abcd.ini
```

Check to have the complete components of the software

- **Synchro File**
  rt18est.sy6

- **UTDF Files**

  To be saved in the folder MGSMProject/Trafficware, the data files are:
  - Lanes.DAT
  - Layout.DAT Phasing.DAT
  - Timing.DAT
  - Volume.DAT

Customized program to export the UTDF data into Access Database

The VB program enables the creation of the databases for 13 intersections of routes 18, 20 and 23. If the databases are already created, the application opens the UTDF files, reads the traffic data namely Lane Data, Phasing Data, Layout Data, Timing Data and Volume Data. These data are then exported to the related databases upon user's such request. All databases are automatically created in the folder MGSMProject\Databases\.

Databases
MGSMProject\Databases\MGE_Gis_Export.mdb
Lane Databases, i.e. LaneDB1.mdb for 1st intersection in
MGSMProject\Databases\LaneDB1.mdb
Phasing Databases, i.e. PhasingDB1.mdb for 1st intersection in
MGSMProject\Databases\PhasingDB1.mdb
Timing Databases, i.e. TimingDB1.mdb for 1st intersection in
MGSMProject\Databases\TimingDB1.mdb
Layout Database, i.e. LayoutDB.mdb for all the intersections in
MGSMProject\Databases\Layout.mdb
Volume Database, i.e. VolumeDB1.mdb for 1st intersection in
MGSMProject\Databases\VolumeDB1.mdb

Trouble shooting with installing the interface of GeoMedia Professional and Synchro

Required Softwares

- GeoMedia Professional 4.0
- Microsoft Access 97
- Microsoft Visual Basic 5.0
- Synchro
- MGE

Required Files

- Myproj1.vbp (Main VB program)
- Abcd.dll (Required for registering GeoMedia Command)
- Abcd.ini (Required for registering GeoMedia Command)
- Project5.vbp (Customized VB program for creating the databases for each intersection)
- Rt18est.sy6 (Synchro file for route 18)
- MGE_GIS_Export.mdb
- Rutgers.mge
- Nj18.gws (Geoworkspace for the application)
- njdot.prm (parameter file)
- intersections.crd (coordinate file)
- seed83.dgn (seed design file)

Write UTDF Files

Write Synchro data(i.e. Volume Data, Timing Data, Phasing Data, Lane Data, Layout Data) for all intersections at a time in UTDF (Universal Traffic Data Format) one by one.

Databases created for each intersection and exporting UTDF data to such databases

To create database for each intersection of the route and exporting UTDF data into related databases needs a customized Visual Basic program: project5.vbp (User must check the directory path of the input UTDF files and the creation of

ODBC

ODBC is required to connect GeoMedia Professional with Access Database

- Control Panel → select Data Sources
- For User DSN, add Microsoft Access Driver
- Select Data source as MGE_GIS_Export.mdb database

Locating Event Control Button on VB form

If locating Event Control command is not possible on the main VB program, then event.ocx command should be registered

- From Start → Run → Regsvr32 event.ocx
- Create .dll file for the VB project
- Name it as abcd.dll to create an initialization file as abcd.ini

Setting up Environment Variables

- Control panel → systems → Environment
- Set MGSM_Dir command name, and Path is the path of MGSMProject

Installing GeoMedia Command

- Copy abcd.dll and abcd.ini in GeoMedia Professional's project folder. On a command prompt, go to the path of \GeoMedia Profession\Programs
- Install GeoMedia's command
- Installusrcmd \prod "GeoMedia Professional" abcd.dll abcd.ini

Activating GeoMedia Command

- When nj18.gws is loaded, activate the registered GeoMedia command named "Displays a Link Map"
- When VB form is closed for the particular intersection and loaded the main form for another intersection, GeoMedia command "Displays a Link Map" has to be reactivated by the user

USING THE SOFTWARE

Introduction

There are four modules in this tool:

- Input-output module
- Macroscopic simulation module
- Expert system module
- Cost-benefit analysis module.

The macroscopic simulation module runs simulations as described in the previous chapters. The module has forms that display simulation results and compares them with pre-timed simulation. The other modules, (i.e., input-output module, expert system module, and cost-benefit analysis module) are described in the next sections.

Input-Output Module

The input-output module of the expert system selects an intersection/arterial for adaptive signal control simulations. This module also helps in easy input and output of new traffic demand; new timing plans, and network geometry changes that might happen in the future in the database.

Software packages such as Synchro, Corsim, and Paramics store traffic demand, network geometry, timing plans, etc., of the network in their own format. To run simulations with adaptive control, it is necessary to pass this information to prototypes developed using programming languages (C and Visual Basic). The best way is to store intersection details in the GIS database and use this information for simulation. Networks can be prepared and calibrated in micro-simulation packages such as Synchro, Corsim, and Paramcis. By decoding their file formats we can generate such a database. In this project, GMPro was used to generate the GIS application, while Synchro files of calibrated networks were used to generate the database.

GMPro is a platform to develop GIS applications. To develop a customized GMPro application, script is written in VB and generated with a customized GeoMedia command. In this application, a GIS map is developed so that a VB form is loaded when the user clicks on any intersection. This enables the user to view data in a similar format as Synchro, so the application of Synchro in the actual GIS environment becomes as user friendly as operating traffic network in Synchro itself. For importing and exporting data from Synchro to GMPro and from GMPro to Synchro, a customized VB program is developed. GMPro reads the imported data from MS Access files access database data. To maintain data consistency, the updated data must be exported to Synchro before running any

Synchro application. The Universal Traffic Data Format (UTDF) is a standard and readable format for exchanging traffic data for Synchro. Figure 58 gives a representation of the system diagram.

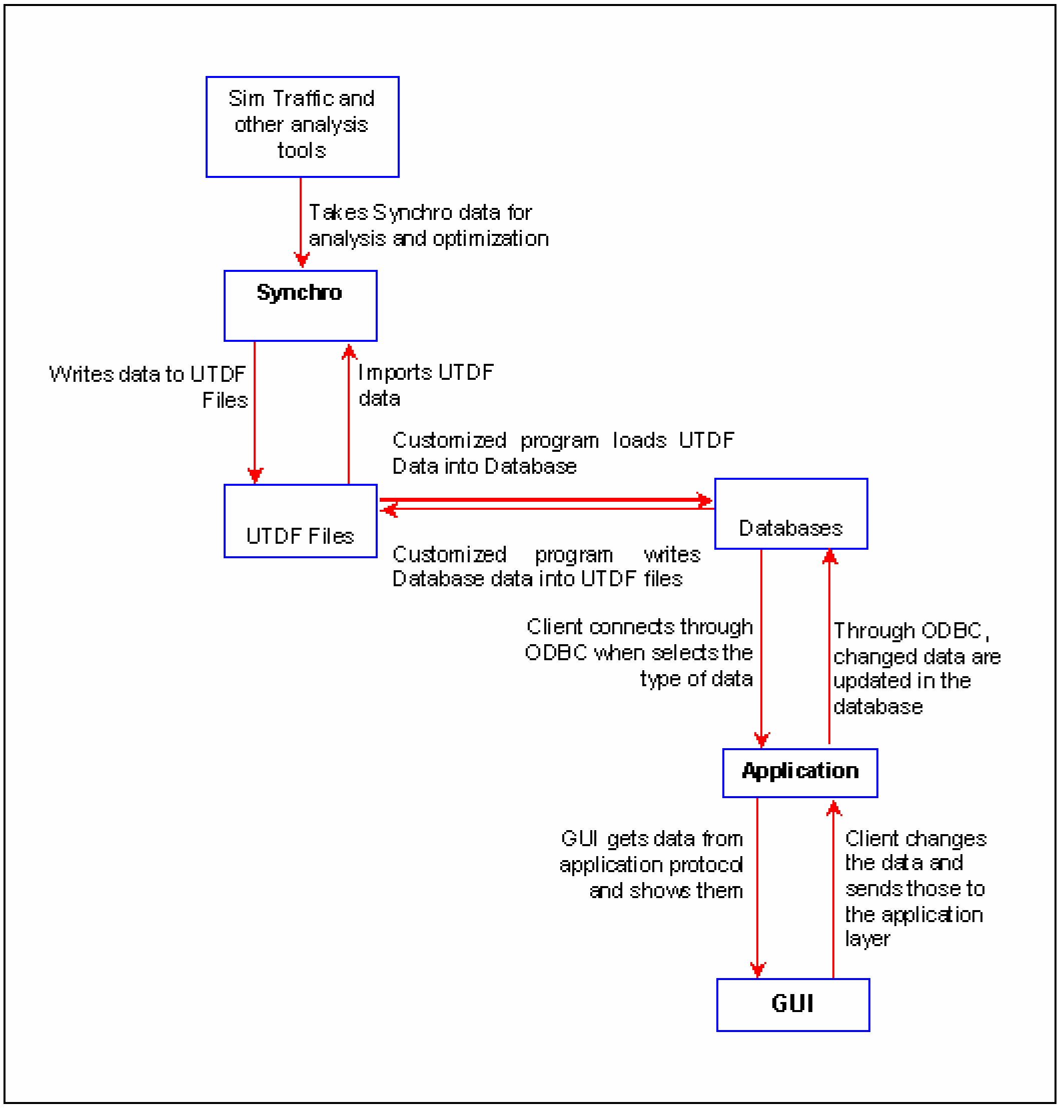


Figure 1 Block Diagram for Input Output Module

Components of the system are:

- Synchro
- UTDF files
- Databases
- GMPro application
- User interfaces

Synchro supports UTDF format data. There are five types of files in Synchro:

Lane Data

Lane data contains information about lanes such as:

- The number of shared lanes
- Ideal saturated flow
- Lane width
- Grade
- Area type
- Storage length
- Storage lanes
- Total lost time
- Leasding detector
- Trailing detector
- Turning speed
- Lane utilization factor
- Right turn factors
- Left turns factors
- Saturated flow rates
- Right pad bike factor
- Left pad factor
- Right turn on red
- Saturated flow rate
- Headway factor

Layout Data

Layout data shows the information such as:

- Intersection ID
- X coordinate
- Y coordinate

Volume Data

Volume data contains information such as:

- Peak hour factor
- Growth factor
- Heavy vehicles
- Bus blockages

Timing Data

Timing data contains information such as:

- Traffic volumes
- Protected and permitted phases
- Detector phases
- Current cycle length
- Split information
- Lock timings
- Offset settings
- Sign control
- Yellow time
- All red time
- Phase lagging
- Allow lead/lag optimize
- Intersection capacity utilization

Phasing Data

Phasing Window contains information such as:

- Current cycle length
- Actuated cycles
- Split information
- Minimum gap time before reduce
- Walk time
- Flashing don’t walk time

When the user activates the “Import Data” button, on the customized interface for GMPro, the VB program runs in the background. Then the application accesses UTDF data files, reads data record by record; and inserts the records into the MS Access data files. There is a separate database for each type of data:

- Lane Data database
- Layout Data database
- Volume Data database
- Timing Data database
- Phasing Data database

When the user activates the “Show Data” button, the application is connected to the related database for the specific data type through ODBC. As the user interface applet appears, data is viewed in the exact similar fashion as in Synchro. The applet gets the data from the application protocol, which is one layer below the applet in the protocol stack.

When the user activates the “Save data” button, the updated data are sent to the lower protocol, which is the application layer. The application is connected to the database through ODBC to enable data access.

When the user activates the “Export Data” button, the application loads and updates data into the UTDF files. Synchro not only writes data to the UTDF files, but also reads the data. The client can run simulation in Simtraffic or any other software to see the effects of the updates.

DEVELOPMENT OF THE APPLICATION

Development of the application consists of the steps described in the following.

Developing a Map

A map is developed in GMPro scripting in Visual Basic 5.0.

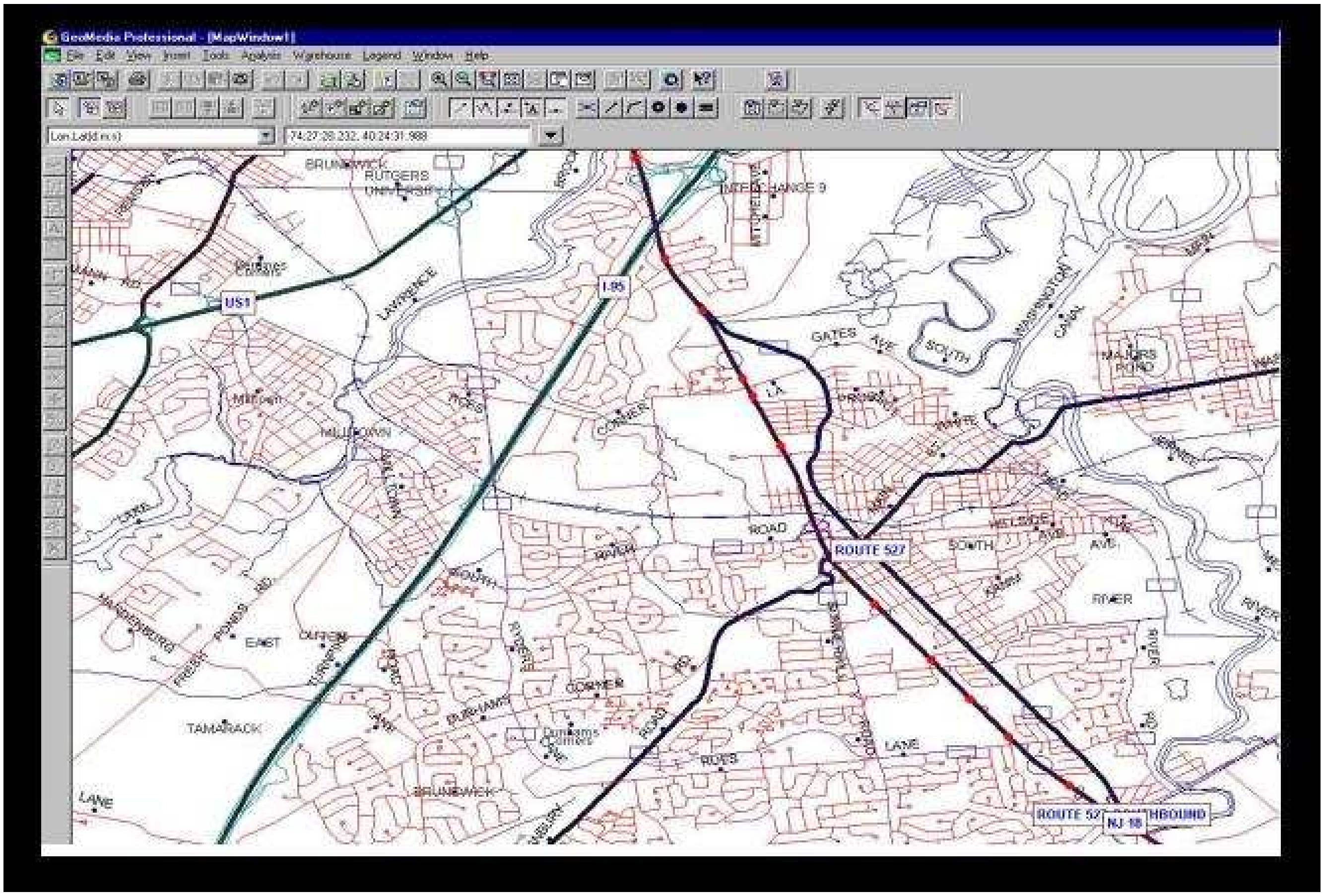


Figure 2 A map in GeoMedia Professional

Dynamic Segmentation for Intersections

To perform dynamic segmentation, a new database table is developed called "points" in MGE_GIS_Export.mdb file for the intersections of Routes 10, 18, and 23. The primary key is defined as "NodeNo" field. From the control points and structures on the highways those are predefined, distances are found for the intersections as offsets from straight-line diagrams. "Points" table and control_points table are related with a common attribute, namely "sri". In ODBC, a new connection is made for those overlaying intersections namely "projInter," pointing to the MGE_GIS_Export.mdb database file. Distributed attributes are defined in the parameter file to enable the visual representation of the intersections on the GIS map. A known marker is used as the referencing system in the modular GIS environment, and the output data to be stored in the parameter file are selected as the point features. The necessary parameters are shown in the following table.

Table 1 Parameter definition

PARAMETER	**ASSOCIATED COLUMN**
Linear Feature ID	Sri
Marker ID	Begin_Marker
Offset	Begin_Offset

Importing Data

To locate synchro data in the GMPro application, the UTDF data must import. Synchro data is saved in the Access database rather than UTDF files to enable the efficient maintenance and update of data through GMPro. Data and number of attributes are different for each intersection for the routes in the highway network. If the number of attributes are not consistent, the VB program is created and completes the following steps as:

- Creating the database
- Creating the table
- Reading the attributes from UTDF files
- Creating related fields
- Appending the table into the database
- Importing all data to the created database

Implementation of this program has the following logical procedure:

- Reading the lines from the UTDF file (.CSV) line by line
- Putting those lines into an array
- Reading the line, which contains the names of the attributes. The comma (,) character is considered as a field delimiter (the user-defined "split" function is

defined to separate the attributes from the single line and to put those names into an array).

- Creating the database
- Creating the table
- Creating the fields where field names are assigned from the array into which attribute names are stored after split (doing this automatically allows the consistency of the attributes in the same type of data for each intersection).
- Once the database is created, calling the function that imports the UTDF data into the database.

The same logic exists behind splitting the UTDF data. Data is read line by line from the UTDF data files. Because the comma character is used as a delimiter, related data are stored into an array for one record at a time. Then the data is inserted into the database tables and saved. The application reads the next line, splits the data, and stores them into database until the end of each data file for each type of data.

Briefly the VB program creates databases for all the intersections and for all types of data. It enables the creation of the databases separately, (e.g. for Lane Data and Volume Data, separate databases are created for each intersection).

Developing Customized User Interface in VB

Customized user interface in VB shows the intersection routes on the heading of the form. There are several option buttons for the different types of data:

- Lane data
- Layout data
- Volume data
- Phasing data
- Timing data

There are four command buttons:

- Show data
- Import data
- Export data
- Close

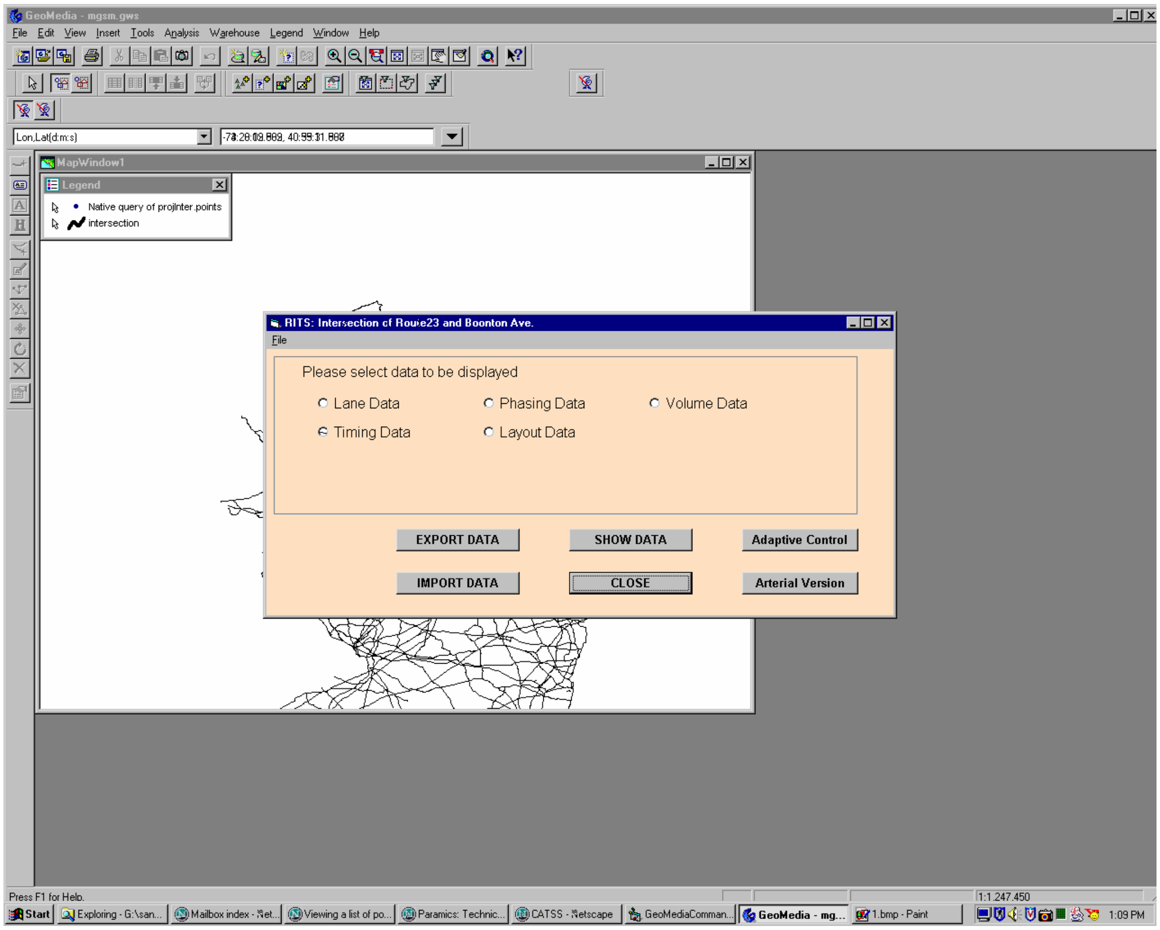
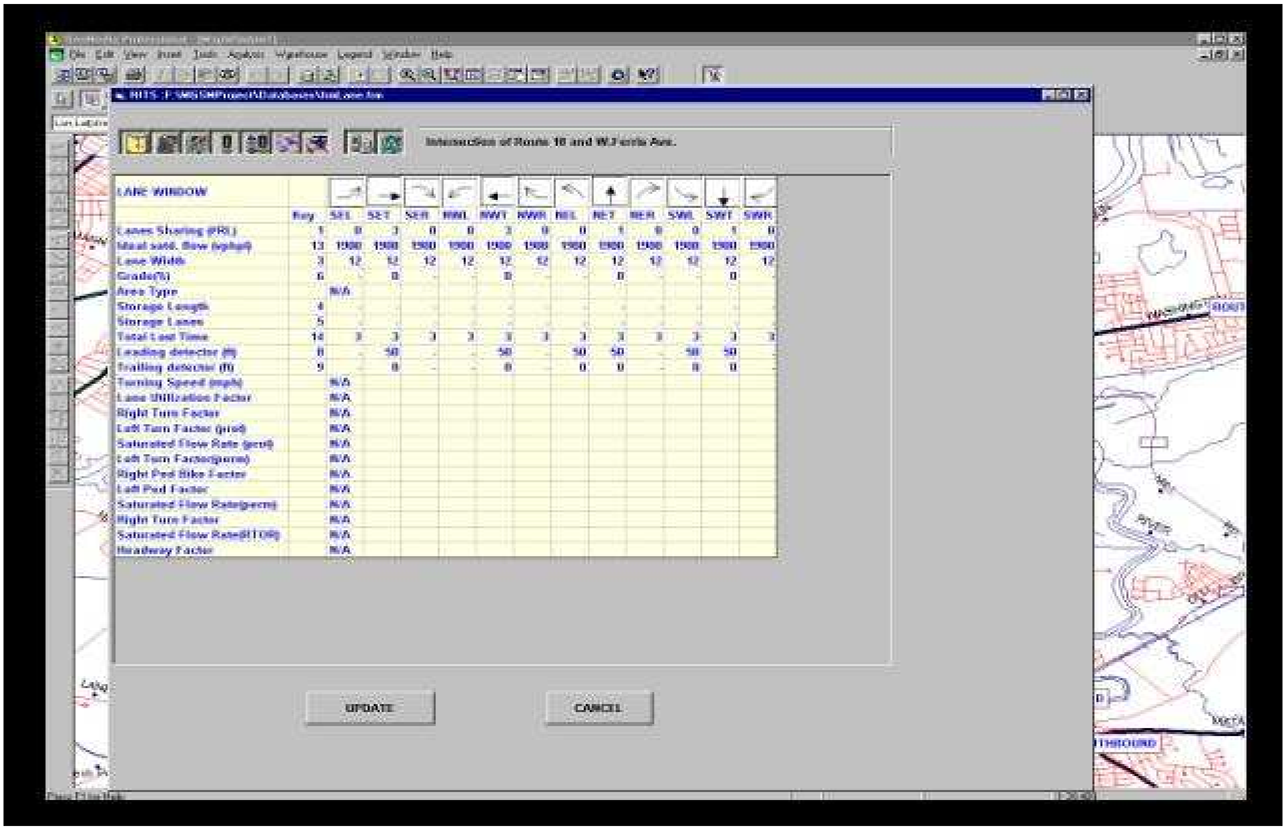


Figure 3 User Interface in GeoMedia Professional application

Sub-forms for Data Display

All the forms that enable the user to access and view data use Microsoft's Flex Grid control. MS Flex Grid is flexible in installing the display rules and the is able to overwrite the particular cells, thus updating data directly in the database. It also has a very similar user interface as Synchro, and the user can be given authorization to change data and see the effects of such updates.



Intersection of Route 18 and W.Ferris Ave.

LANE WINDOW	Key	SEL	SET	SER	NWL	NWT	NWR	NEL	NET	NER	SWL	SWT	SWR
Lanes Sharing (#RL)	1	0	3	0	0	3	0	0	1	0	0	1	0
Ideal satd. flow (vphpl)	13	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900	1900
Lane Width	3	12	12	12	12	12	12	12	12	12	12	12	12
Grade(%)	6	-	0	-	-	0	-	-	0	-	-	0	-
Area Type		N/A											
Storage Length	4	-	-	-	-	-	-	-	-	-	-	-	-
Storage Lanes	5	-	-	-	-	-	-	-	-	-	-	-	-
Total Lost Time	14	3	3	3	3	3	3	3	3	3	3	3	3
Leading detector (ft)	8	-	50	-	-	50	-	50	50	-	50	50	-
Trailing detector (ft)	9	-	0	-	-	0	-	0	0	-	0	0	-
Turning Speed (mph)		N/A											
Lane Utilization Factor		N/A											
Right Turn Factor		N/A											
Left Turn Factor (prot)		N/A											
Saturated Flow Rate (prot)		N/A											
Left Turn Factor(perm)		N/A											
Right Ped Bike Factor		N/A											
Left Ped Factor		N/A											
Saturated Flow Rate(perm)		N/A											
Right Turn Factor		N/A											
Saturated Flow Rate(RTOR)		N/A											
Headway Factor		N/A											

Figure 4 A User Interface Showing Lane Data in GeoMedia Professional Application

Updating data

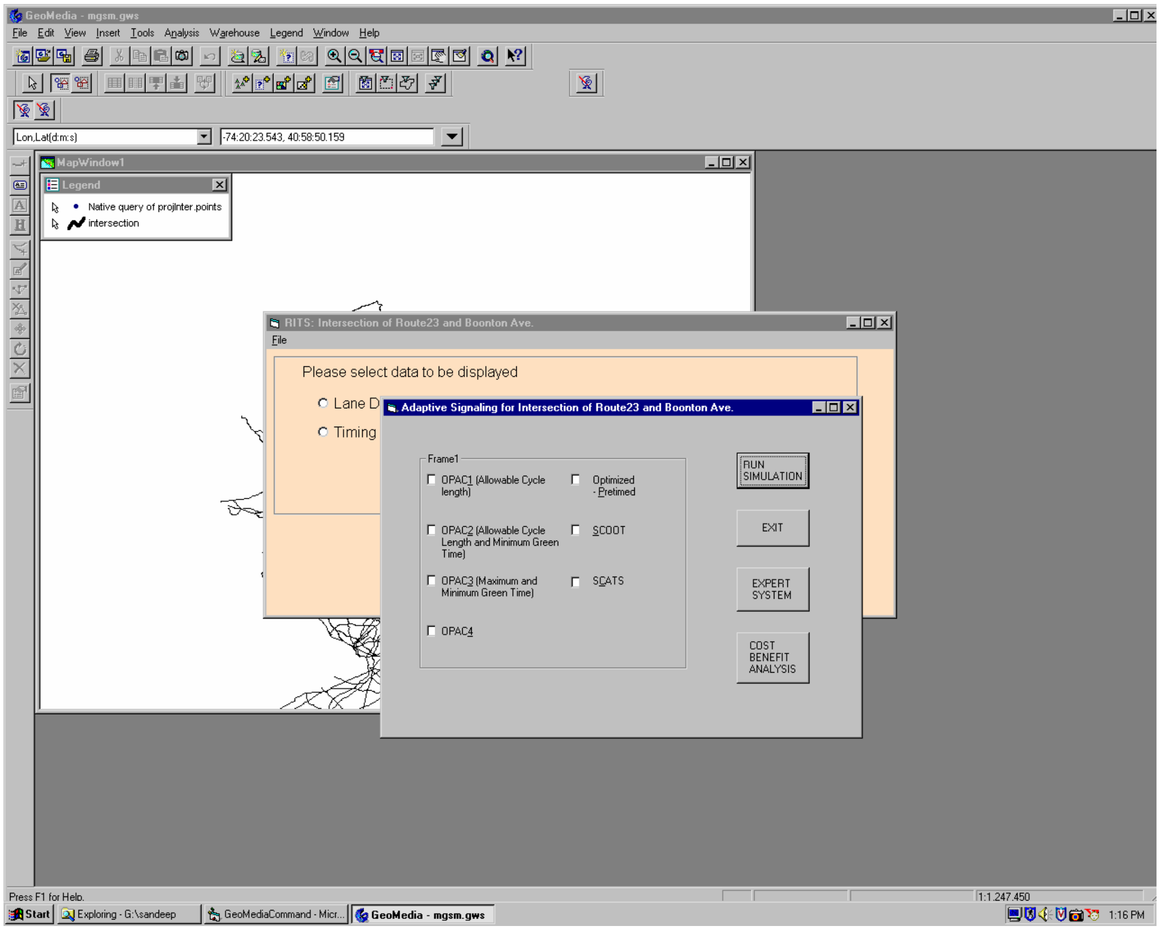
The user may update data for any intersection on the GIS map through the customized user interface by the “Update data” button. The user simply selects the cell in Flex Grid, enters the data, and activates the “Update Data” button. Because ODBC is created for each database, performing an “update query” completes updates. The user also views and analyzes the effects of the updates in Synchro. To maintain permanent data consistency, the update process takes place only in the database and not in Synchro. The user always exports data back to Synchro to view the updates in Synchro.

Exporting Data

The user may export data for any intersection on the GIS map through the customized user interface by the “Export data” button. The user exports data back to Synchro by clicking the command button on the main VB form for each intersection. A flat file is then created, and data is written in this file along with the field name, using comma (,) as a delimiter with ODBC. After that Synchro is able to read data from the UTDF files (Refer to Figure 3).

Adaptive Signal Control

Select adaptive control button from the main menu to run adatpive traffic control. A new window with options to select different Adaptive Signal Strategies will open. This window also has options for expert system and cost benefit analysis. It is shown in Figure 5.



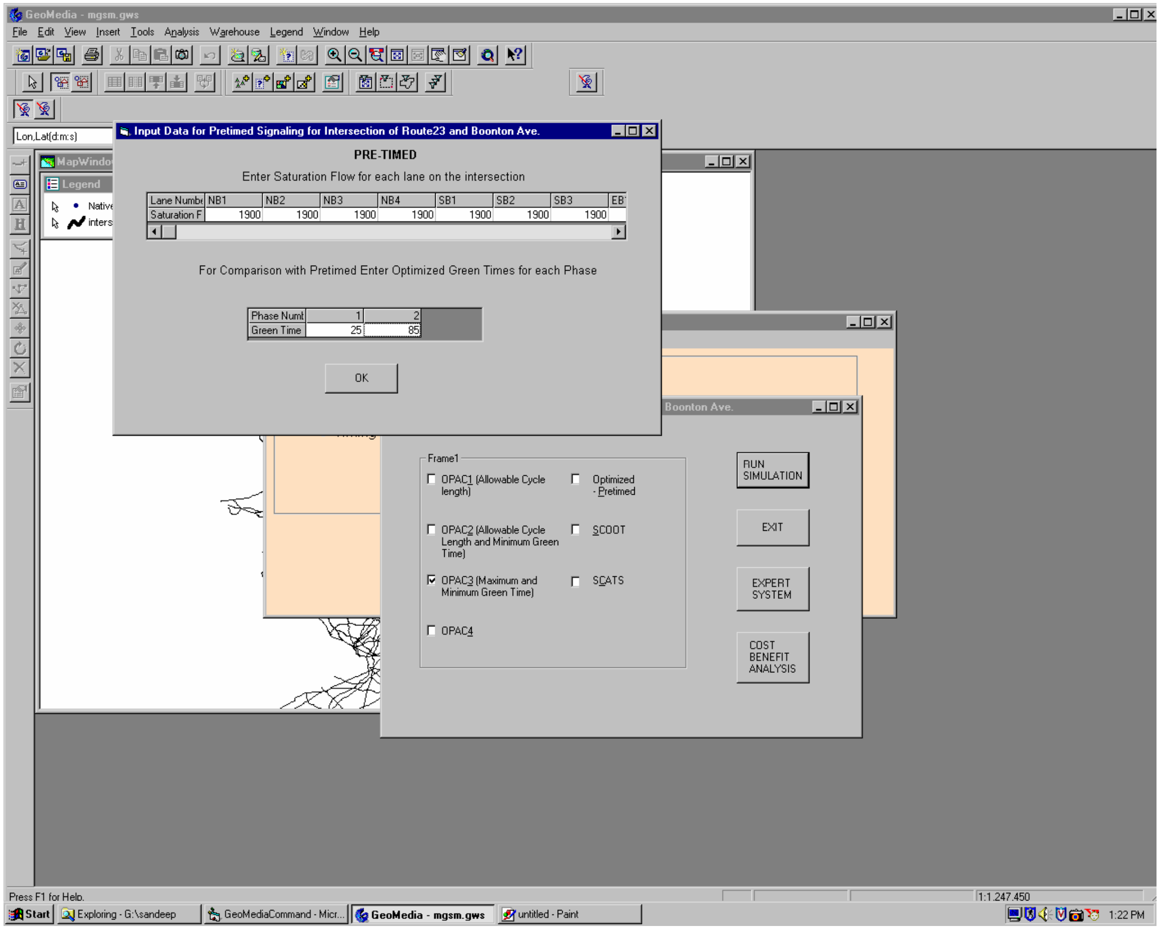
Macroscopic Simulation

STEP - 1

The adaptive control strategies available with the software are:

1) OPAC-Like Prototype – 1
2) OPAC-Like Prototype – 2
3) OPAC-Like Prototype – 3
4) SCOOT
5) SCATS

A user can select only one strategy at a time. The simulation output is compared with pretimed signal case. Hence selecting any one of these strategies, opens a window, which prompts to input optimized timing plan for pre-timed signal. The default saturation flow rate in all lanes for the selected intersection is 1900 vphpl, which can be changed in the window. The window prompts for green time for each phase in seconds.



Input Data for Pretimed Signaling for Intersection of Route23 and Boonton Ave.

PRE-TIMED

Enter Saturation Flow for each lane on the intersection

Lane Numbe	NB1	NB2	NB3	NB4	SB1	SB2	SB3	EB
Saturation F	1900	1900	1900	1900	1900	1900	1900	

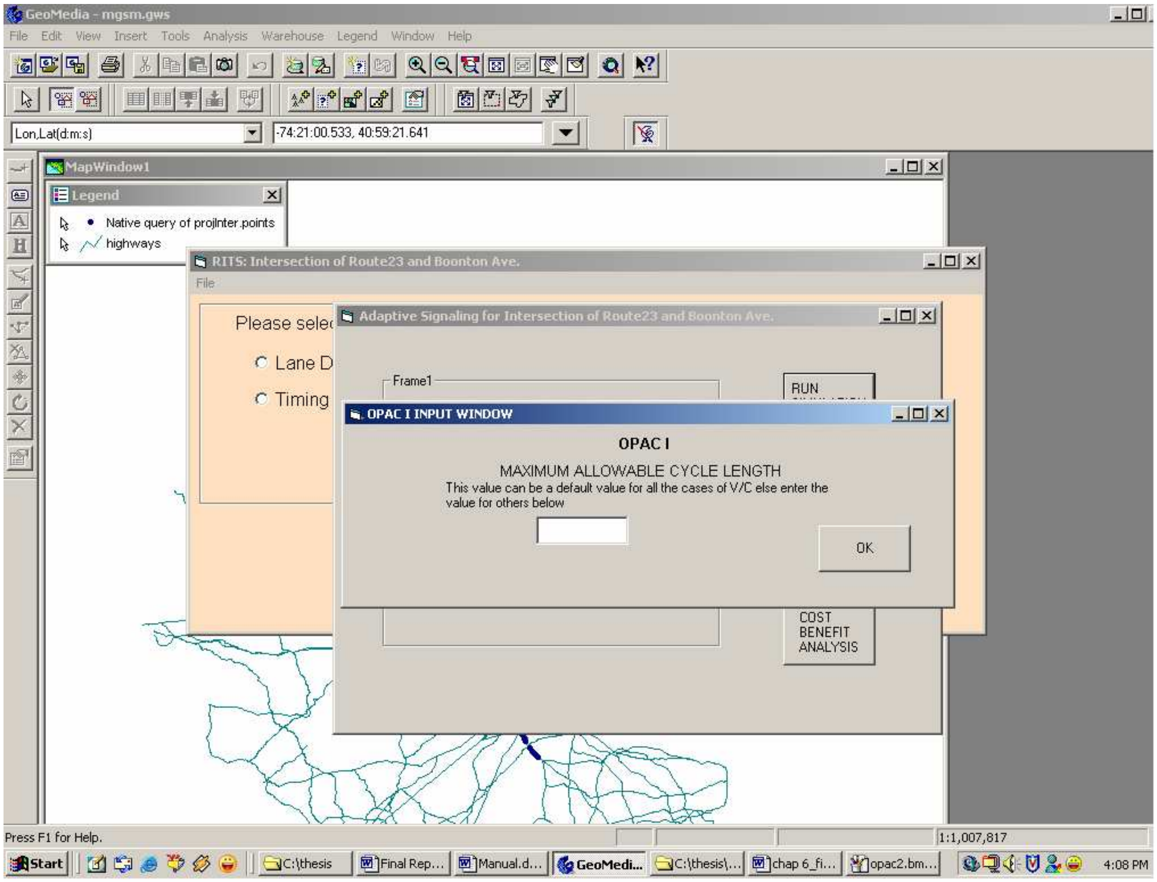
For Comparison with Pretimed Enter Optimized Green Times for each Phase

Phase Numb	1	2
Green Time	25	85

☐ OPAC1 (Allowable Cycle length)
☐ Optimized - Pretimed
☐ OPAC2 (Allowable Cycle Length and Minimum Green Time)
☐ SCOOT
☑ OPAC3 (Maximum and Minimum Green Time)
☐ SCATS
☐ OPAC4

STEP-2

Press "OK" button to proceed with adaptive control simulation. A new window specific to the type of selected adatpive signal strategy opens prompting for more input. For Opac-like prototype 1 the only input required is maximum allowable cycle length (in seconds). For Opac-like prototpye-2 the required inputs are maxium allowable cycle length (in seconds) and minimum green time for each phase (in seconds). Opac-like prototype-3 required minimum and maximum green time for each phase (in seconds). SCOOT requires normal green time, all red time, maximum and minimum green time (in seconds). Normal green time can be same value as optimized pre-timed control green time values for each phase entered earlier. SCATS requires all the inputs as in SCOOT with addition of minimum and maximum cycle lengths (in seconds).

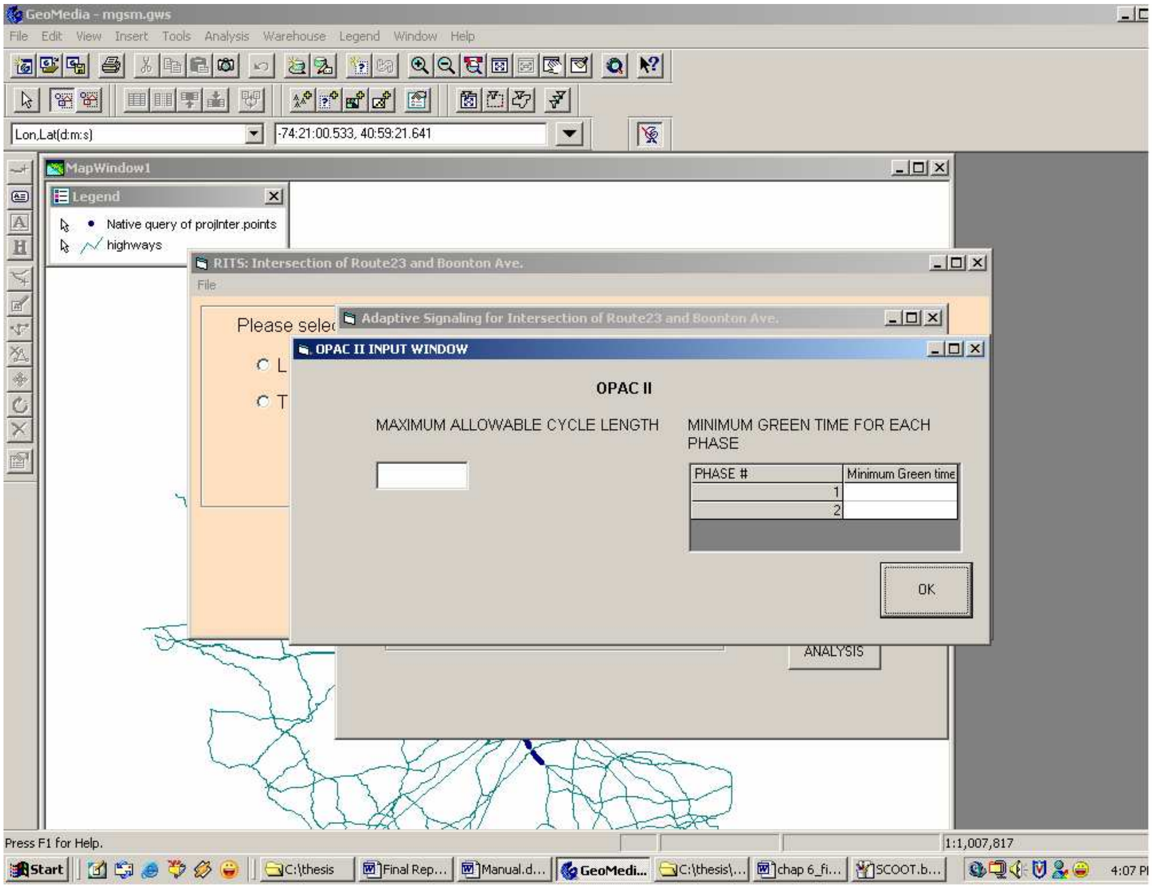


Lon,Lat(d:m:s) -74:21:00.533, 40:59:21.641

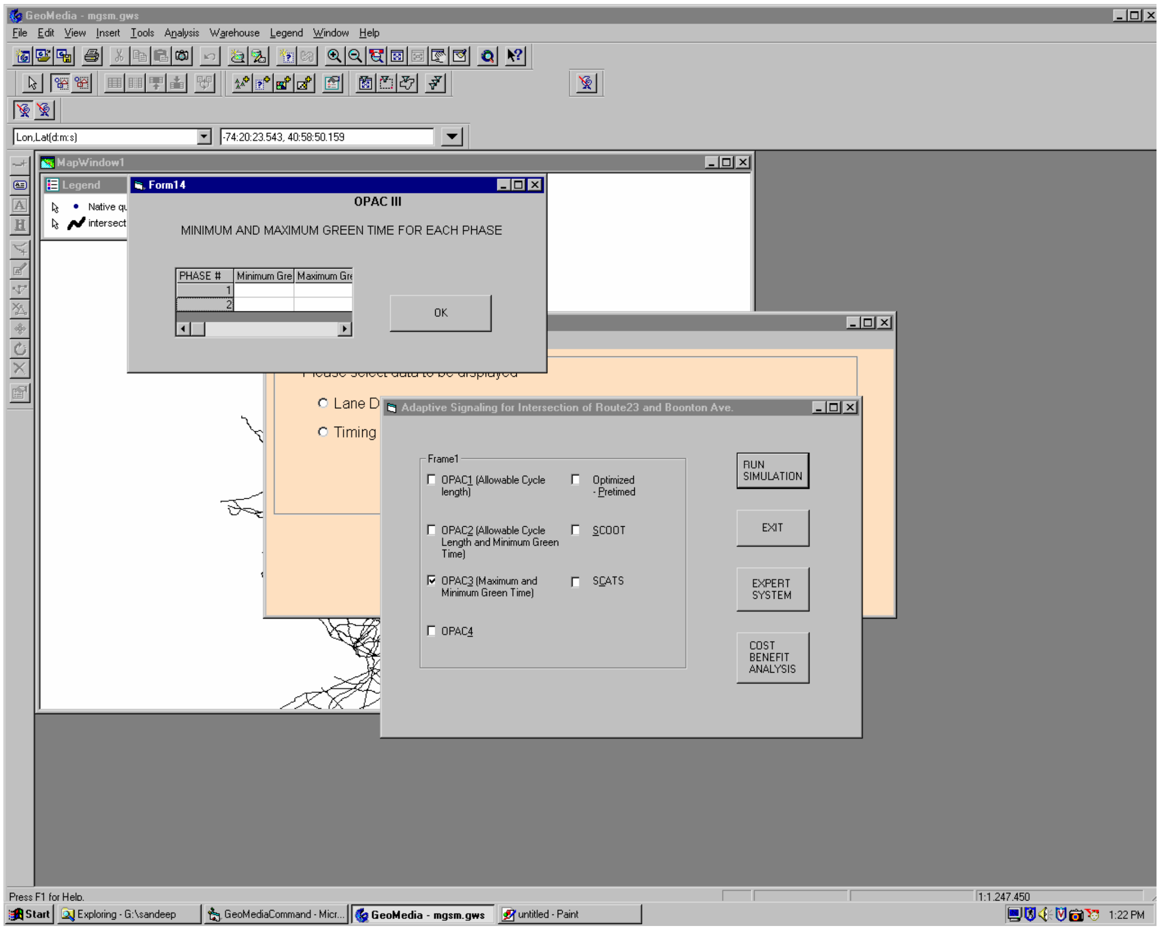
OPAC I

MAXIMUM ALLOWABLE CYCLE LENGTH

This value can be a default value for all the cases of V/C else enter the value for others below



PHASE #	Minimum Green time
1	
2	



OPAC III

MINIMUM AND MAXIMUM GREEN TIME FOR EACH PHASE

PHASE #	Minimum Gre	Maximum Gre
1		
2		

☐ OPAC1 (Allowable Cycle length)

☐ Optimized - Pretimed

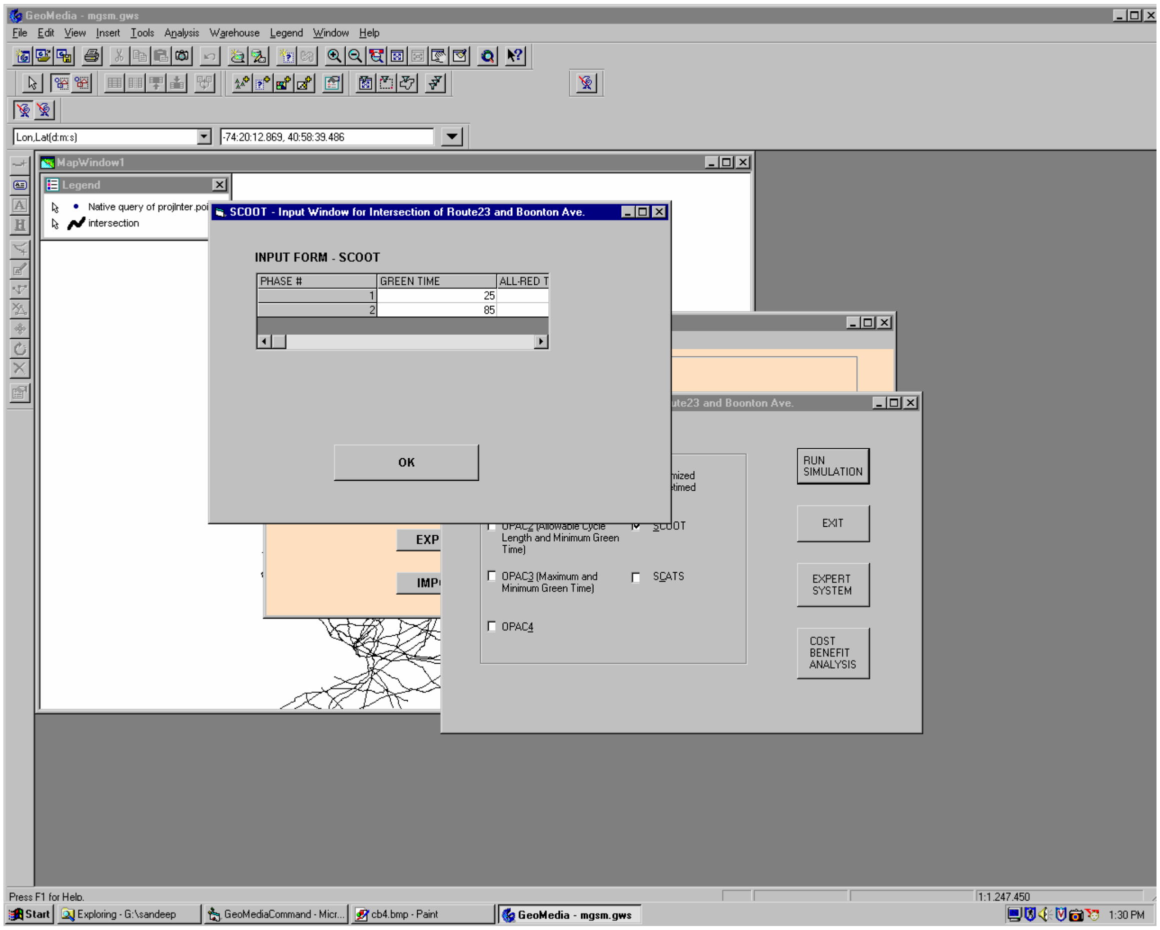
☐ OPAC2 (Allowable Cycle Length and Minimum Green Time)

☐ SCOOT

☑ OPAC3 (Maximum and Minimum Green Time)

☐ SCATS

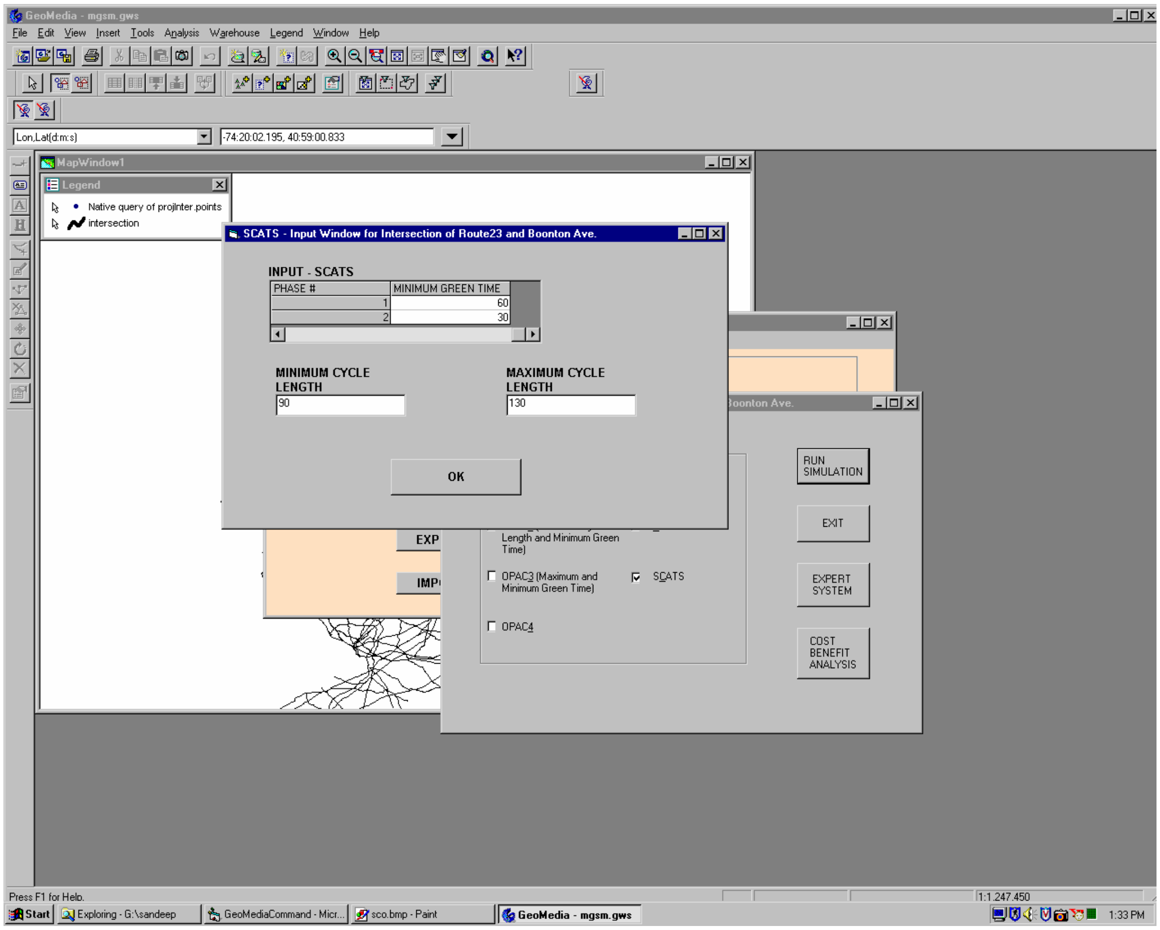
☐ OPAC4



SCOOT - Input Window for Intersection of Route23 and Boonton Ave.

INPUT FORM - SCOOT

PHASE #	GREEN TIME	ALL-RED T
1	25	
2	85	



Lon,Lat(d:m:s) -74:20:02.195, 40:59:00.833

SCATS - Input Window for Intersection of Route23 and Boonton Ave.

INPUT - SCATS

PHASE #	MINIMUM GREEN TIME
1	60
2	30

MINIMUM CYCLE LENGTH: 90

MAXIMUM CYCLE LENGTH: 130

Length and Minimum Green Time)

☐ OPAC3 (Maximum and Minimum Green Time)

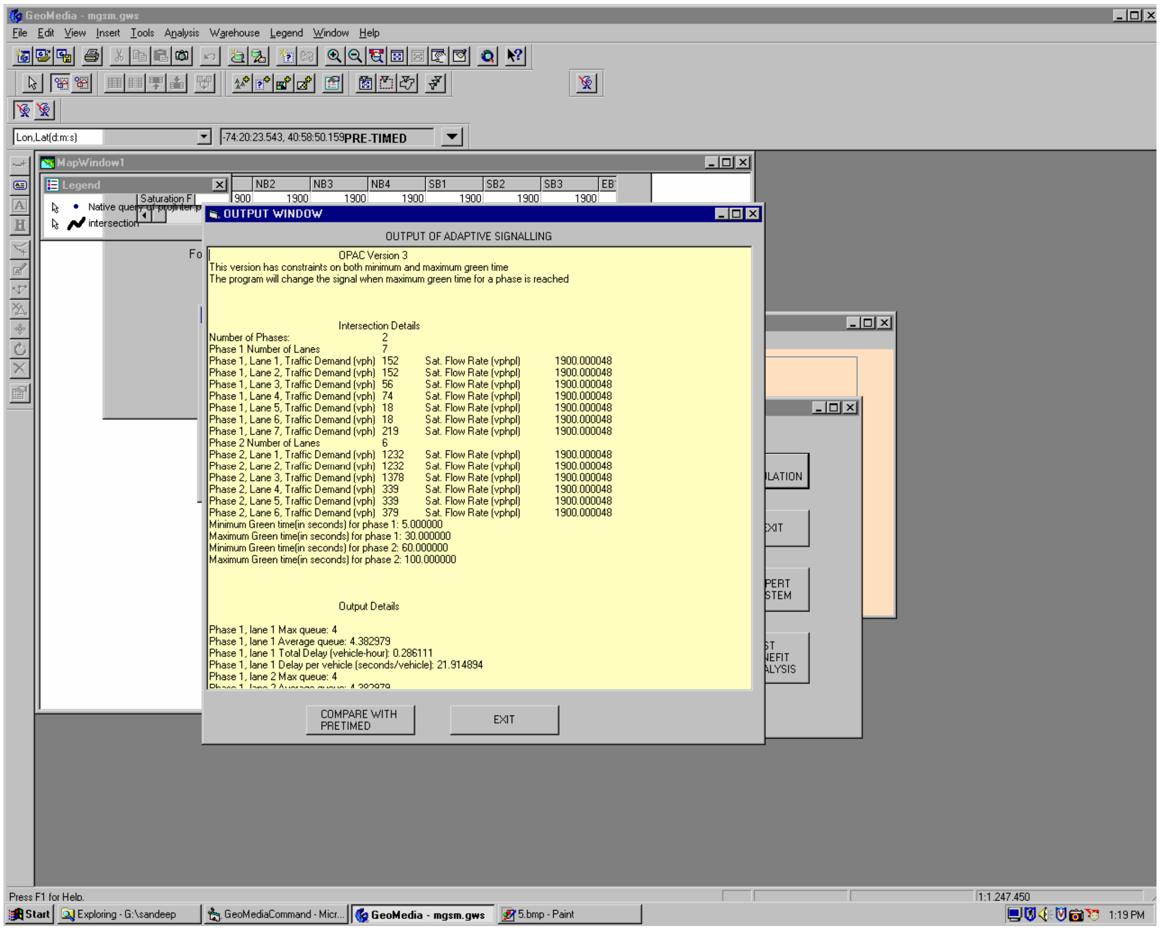
☑ SCATS

☐ OPAC4

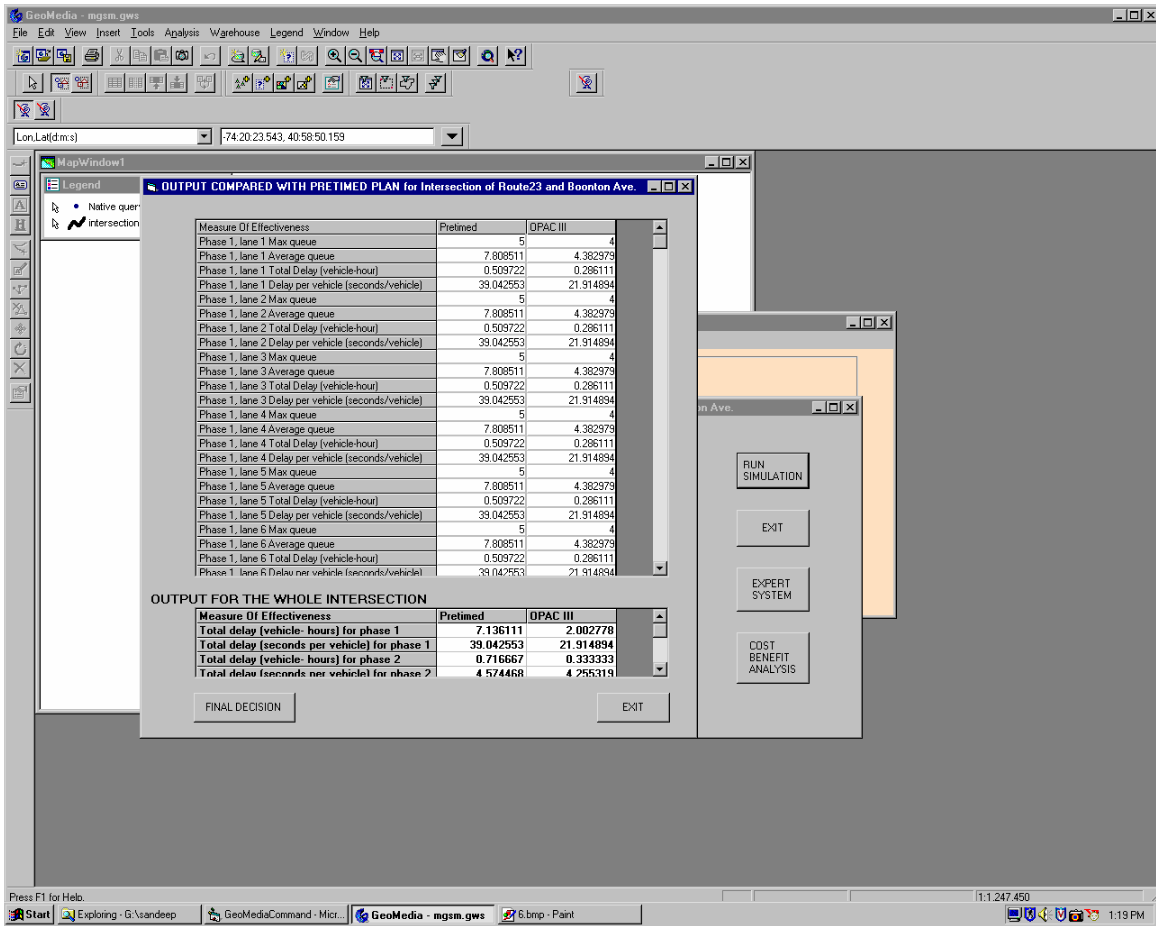
1:1 247 450

STEP-3

Once microscopic simualtions are completed, a message requesting to hit “OK” button appears to view the output. The output window has output in terms of maximum queue per lane, total delay per second and total delay in vehicle-hours per lane and phase. There is an additional optional to compare the output with pre-timed output. The two output windows appear as shown in the following figures.



```
                        OPAC Version 3
This version has constraints on both minimum and maximum green time
The program will change the signal when maximum green time for a phase is reached



                        Intersection Details
Number of Phases:                     2
Phase 1 Number of Lanes               7
Phase 1, Lane 1, Traffic Demand (vph) 152     Sat. Flow Rate (vphpl)     1900.000048
Phase 1, Lane 2, Traffic Demand (vph) 152     Sat. Flow Rate (vphpl)     1900.000048
Phase 1, Lane 3, Traffic Demand (vph) 56      Sat. Flow Rate (vphpl)     1900.000048
Phase 1, Lane 4, Traffic Demand (vph) 74      Sat. Flow Rate (vphpl)     1900.000048
Phase 1, Lane 5, Traffic Demand (vph) 18      Sat. Flow Rate (vphpl)     1900.000048
Phase 1, Lane 6, Traffic Demand (vph) 18      Sat. Flow Rate (vphpl)     1900.000048
Phase 1, Lane 7, Traffic Demand (vph) 219     Sat. Flow Rate (vphpl)     1900.000048
Phase 2 Number of Lanes               6
Phase 2, Lane 1, Traffic Demand (vph) 1232    Sat. Flow Rate (vphpl)     1900.000048
Phase 2, Lane 2, Traffic Demand (vph) 1232    Sat. Flow Rate (vphpl)     1900.000048
Phase 2, Lane 3, Traffic Demand (vph) 1378    Sat. Flow Rate (vphpl)     1900.000048
Phase 2, Lane 4, Traffic Demand (vph) 339     Sat. Flow Rate (vphpl)     1900.000048
Phase 2, Lane 5, Traffic Demand (vph) 339     Sat. Flow Rate (vphpl)     1900.000048
Phase 2, Lane 6, Traffic Demand (vph) 379     Sat. Flow Rate (vphpl)     1900.000048
Minimum Green time(in seconds) for phase 1: 5.000000
Maximum Green time(in seconds) for phase 1: 30.000000
Minimum Green time(in seconds) for phase 2: 60.000000
Maximum Green time(in seconds) for phase 2: 100.000000



                        Output Details

Phase 1, lane 1 Max queue: 4
Phase 1, lane 1 Average queue: 4.382979
Phase 1, lane 1 Total Delay (vehicle-hour): 0.286111
Phase 1, lane 1 Delay per vehicle (seconds/vehicle): 21.914894
Phase 1, lane 2 Max queue: 4
```



OUTPUT COMPARED WITH PRETIMED PLAN for Intersection of Route23 and Boonton Ave.

Measure Of Effectiveness	Pretimed	OPAC III
Phase 1, lane 1 Max queue	5	4
Phase 1, lane 1 Average queue	7.808511	4.382979
Phase 1, lane 1 Total Delay (vehicle-hour)	0.509722	0.286111
Phase 1, lane 1 Delay per vehicle (seconds/vehicle)	39.042553	21.914894
Phase 1, lane 2 Max queue	5	4
Phase 1, lane 2 Average queue	7.808511	4.382979
Phase 1, lane 2 Total Delay (vehicle-hour)	0.509722	0.286111
Phase 1, lane 2 Delay per vehicle (seconds/vehicle)	39.042553	21.914894
Phase 1, lane 3 Max queue	5	4
Phase 1, lane 3 Average queue	7.808511	4.382979
Phase 1, lane 3 Total Delay (vehicle-hour)	0.509722	0.286111
Phase 1, lane 3 Delay per vehicle (seconds/vehicle)	39.042553	21.914894
Phase 1, lane 4 Max queue	5	4
Phase 1, lane 4 Average queue	7.808511	4.382979
Phase 1, lane 4 Total Delay (vehicle-hour)	0.509722	0.286111
Phase 1, lane 4 Delay per vehicle (seconds/vehicle)	39.042553	21.914894
Phase 1, lane 5 Max queue	5	4
Phase 1, lane 5 Average queue	7.808511	4.382979
Phase 1, lane 5 Total Delay (vehicle-hour)	0.509722	0.286111
Phase 1, lane 5 Delay per vehicle (seconds/vehicle)	39.042553	21.914894
Phase 1, lane 6 Max queue	5	4
Phase 1, lane 6 Average queue	7.808511	4.382979
Phase 1, lane 6 Total Delay (vehicle-hour)	0.509722	0.286111
Phase 1, lane 6 Delay per vehicle (seconds/vehicle)	39.042553	21.914894

OUTPUT FOR THE WHOLE INTERSECTION

Measure Of Effectiveness	Pretimed	OPAC III
Total delay (vehicle- hours) for phase 1	7.136111	2.002778
Total delay (seconds per vehicle) for phase 1	39.042553	21.914894
Total delay (vehicle- hours) for phase 2	0.716667	0.333333
Total delay (seconds per vehicle) for phase 2	4.574468	4.255319

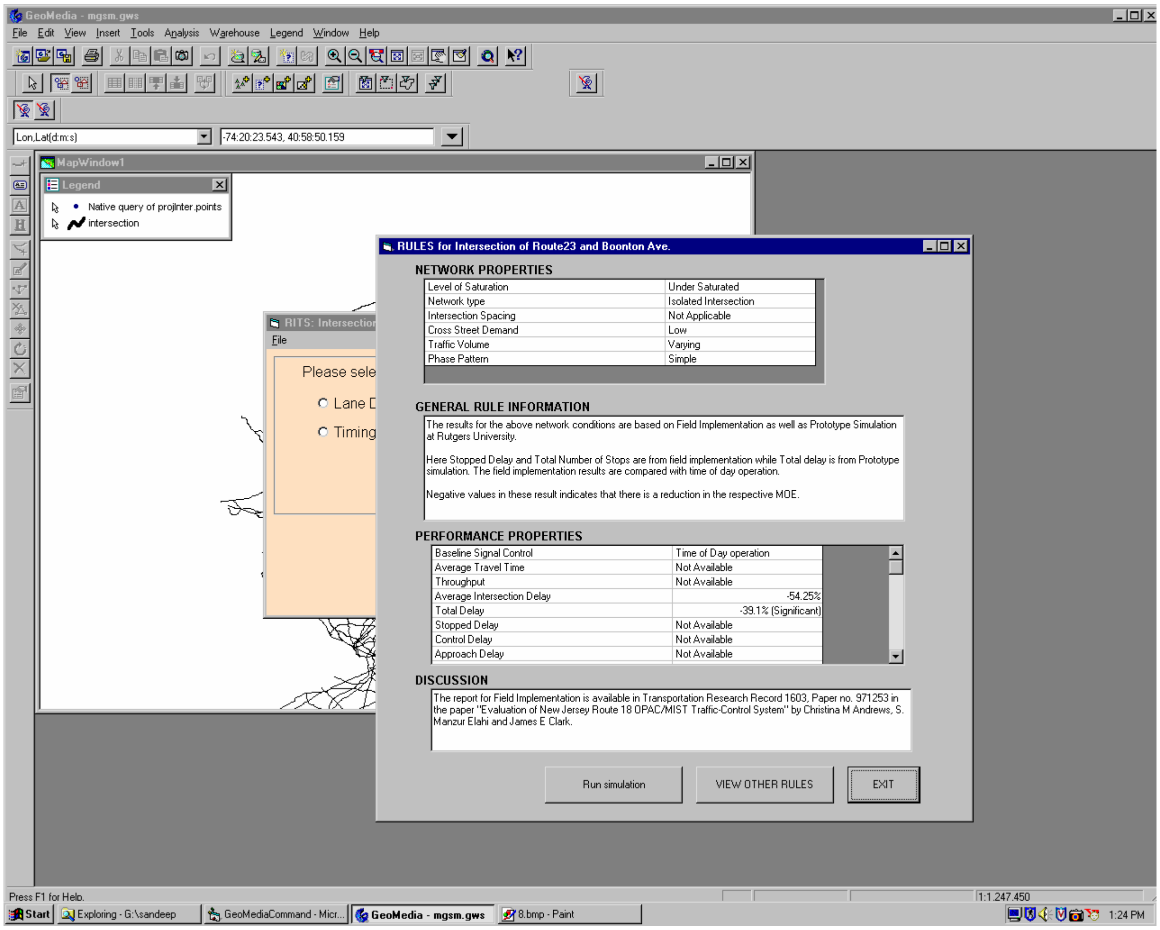
STEP-4

The output window that compares results with pre-timed simulation has an option to give a “decision” regarding the adatpive signal strategy performance on the selected intersection. Pressing the “Final Decision” button opens a window that describes messages if the signal strategy has performed better or not compared with pre-timed signal based on macrosopic simulations and knowledge base.

Expert System

STEP-1

After selecting the adpative signal strategy, a user can choose the option of expert system. This opens a window describing performance of the selected strategy available in literature. It give general information, information regarding the type of network, performance properties and respective reference for the rule. This is as shown in figure below:



NETWORK PROPERTIES

Level of Saturation	Under Saturated
Network type	Isolated Intersection
Intersection Spacing	Not Applicable
Cross Street Demand	Low
Traffic Volume	Varying
Phase Pattern	Simple

GENERAL RULE INFORMATION

The results for the above network conditions are based on Field Implementation as well as Prototype Simulation at Rutgers University.

Here Stopped Delay and Total Number of Stops are from field implementation while Total delay is from Prototype simulation. The field implementation results are compared with time of day operation.

Negative values in these result indicates that there is a reduction in the respective MOE.

PERFORMANCE PROPERTIES

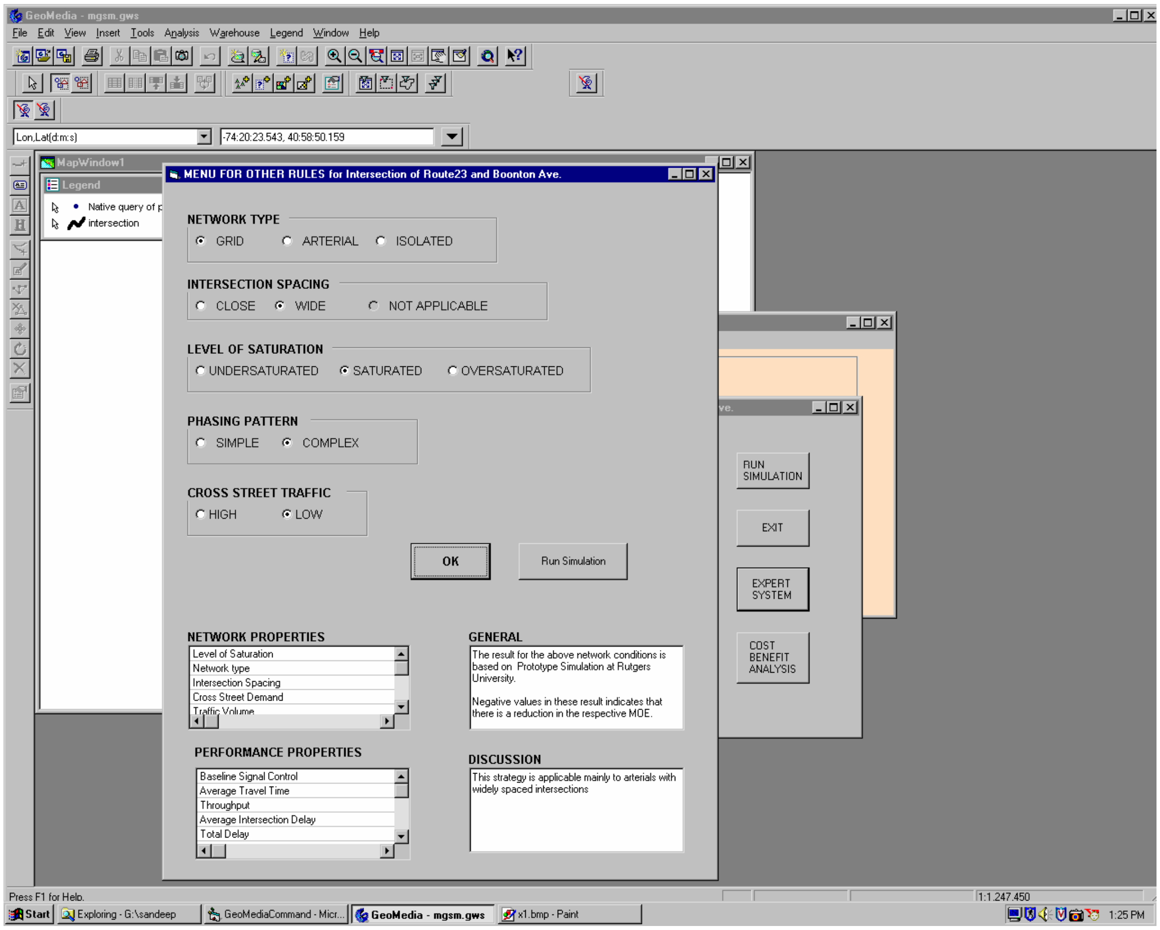
Baseline Signal Control	Time of Day operation
Average Travel Time	Not Available
Throughput	Not Available
Average Intersection Delay	-54.25%
Total Delay	-39.1% (Significant)
Stopped Delay	Not Available
Control Delay	Not Available
Approach Delay	Not Available

DISCUSSION

The report for Field Implementation is available in Transportation Research Record 1603, Paper no. 971253 in the paper "Evaluation of New Jersey Route 18 OPAC/MIST Traffic-Control System" by Christina M Andrews, S. Manzur Elahi and James E Clark.

STEP-2

The user also has the option to view other rules based on different network and traffic characteristics. The “View Other Rules” button opens a form that asks for the type of network and traffic and pulls respective rule. If there is no rule available, a small message appears and prompts the user to select other type of network.



MENU FOR OTHER RULES for Intersection of Route23 and Boonton Ave.

NETWORK TYPE
● GRID ○ ARTERIAL ○ ISOLATED

INTERSECTION SPACING
○ CLOSE ● WIDE ○ NOT APPLICABLE

LEVEL OF SATURATION
○ UNDERSATURATED ● SATURATED ○ OVERSATURATED

PHASING PATTERN
○ SIMPLE ● COMPLEX

CROSS STREET TRAFFIC
○ HIGH ● LOW

NETWORK PROPERTIES
- Level of Saturation
- Network type
- Intersection Spacing
- Cross Street Demand
- Traffic Volume

GENERAL

The result for the above network conditions is based on Prototype Simulation at Rutgers University.

Negative values in these result indicates that there is a reduction in the respective MOE.

PERFORMANCE PROPERTIES
- Baseline Signal Control
- Average Travel Time
- Throughput
- Average Intersection Delay
- Total Delay

DISCUSSION

This strategy is applicable mainly to arterials with widely spaced intersections

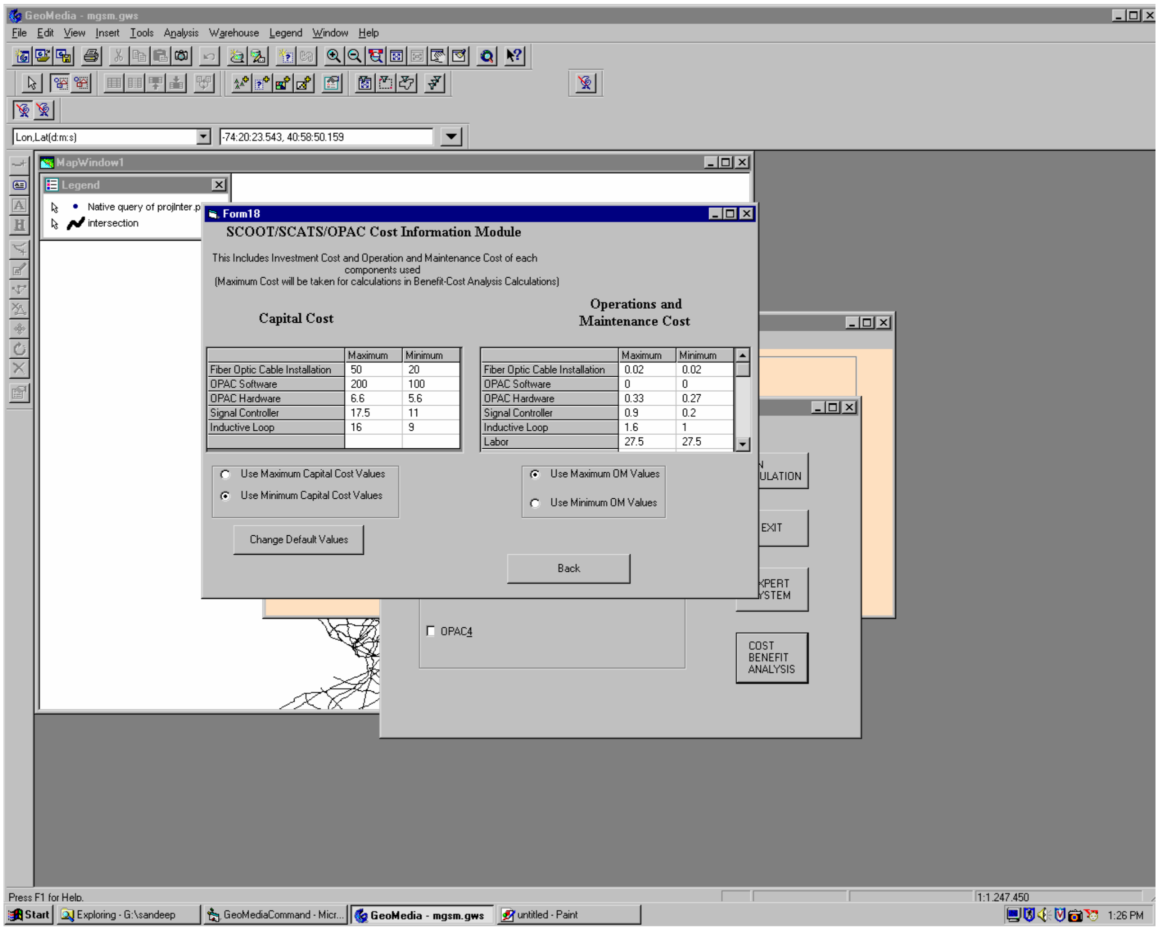
COST BENEFIT ANALYSIS

STEP-1

After selecting the type of control strategy, a user can select the cost-benefit analysis option. A new form opens which has the following options:

1) View/Edit Cost Information

   This lists the cost of equipment used to implement the adaptive control strategy. A user can either select maximum or minimum value or can add custom value using the “Change Default Value” button. This opens a new form where new values can be entered and updated to a custom file. Note that new values entered will not be deful t values for future simulations



SCOOT/SCATS/OPAC Cost Information Module

This Includes Investment Cost and Operation and Maintenance Cost of each components used
(Maximum Cost will be taken for calculations in Benefit-Cost Analysis Calculations)

Capital Cost

	Maximum	Minimum
Fiber Optic Cable Installation	50	20
OPAC Software	200	100
OPAC Hardware	6.6	5.6
Signal Controller	17.5	11
Inductive Loop	16	9

Operations and Maintenance Cost

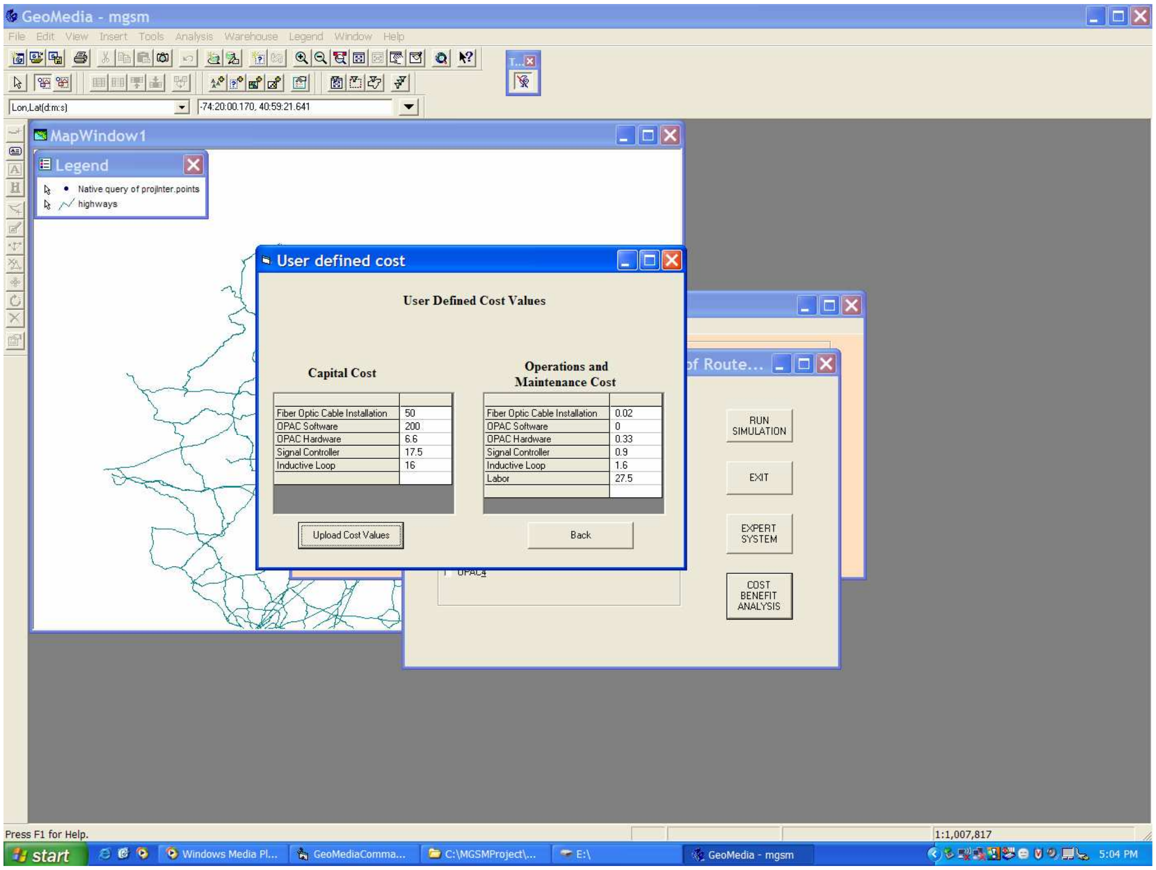
	Maximum	Minimum
Fiber Optic Cable Installation	0.02	0.02
OPAC Software	0	0
OPAC Hardware	0.33	0.27
Signal Controller	0.9	0.2
Inductive Loop	1.6	1
Labor	27.5	27.5

○ Use Maximum Capital Cost Values
● Use Minimum Capital Cost Values

● Use Maximum OM Values
○ Use Minimum OM Values

Change Default Values

Back



User Defined Cost Values

Capital Cost

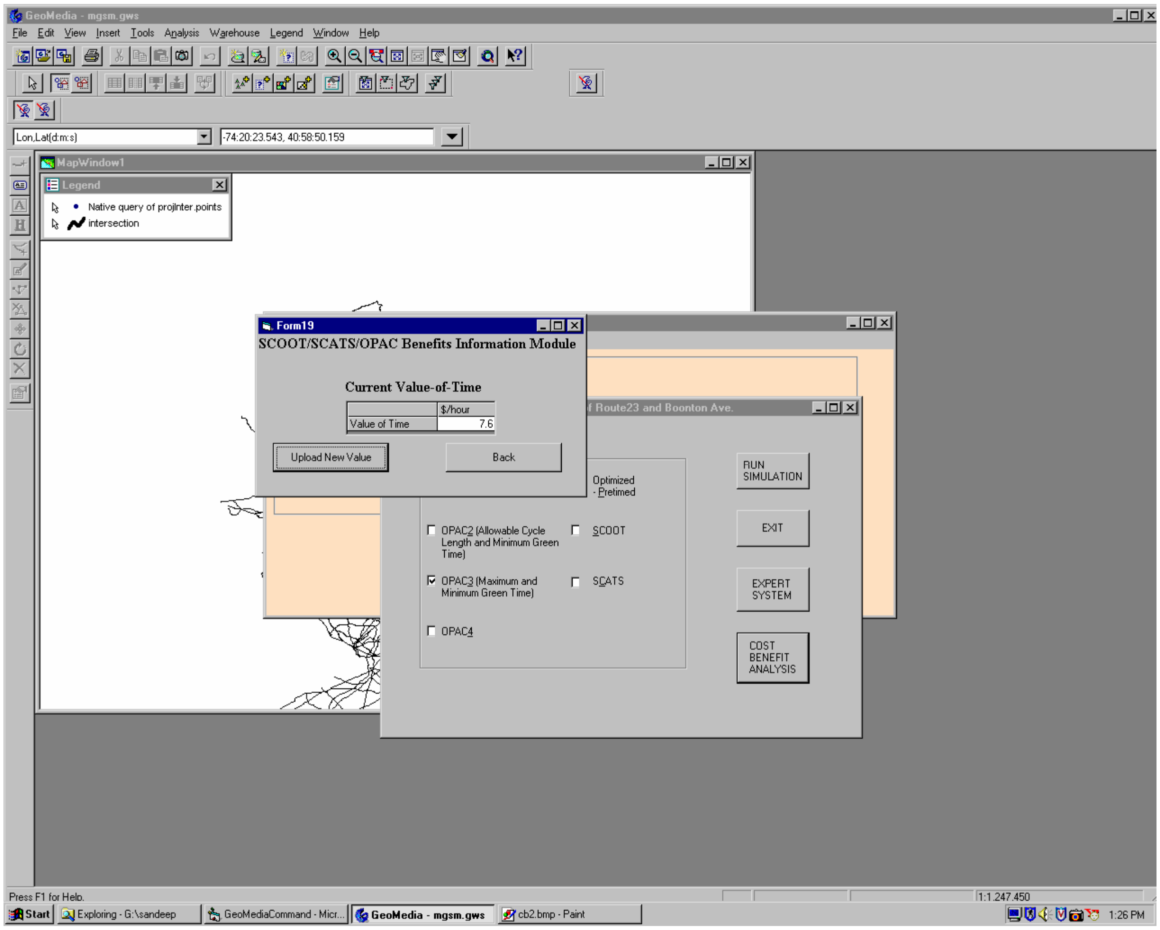
Fiber Optic Cable Installation	50
OPAC Software	200
OPAC Hardware	6.6
Signal Controller	17.5
Inductive Loop	16

Operations and Maintenance Cost

Fiber Optic Cable Installation	0.02
OPAC Software	0
OPAC Hardware	0.33
Signal Controller	0.9
Inductive Loop	1.6
Labor	27.5

2) View/Edit Benefit Information

The only benefit that can be achived from the simulations is reduction in total delay. The default monetary value for travel time saving is 7.6$/hour. This can be changed using the update value button.



Form19

SCOOT/SCATS/OPAC Benefits Information Module

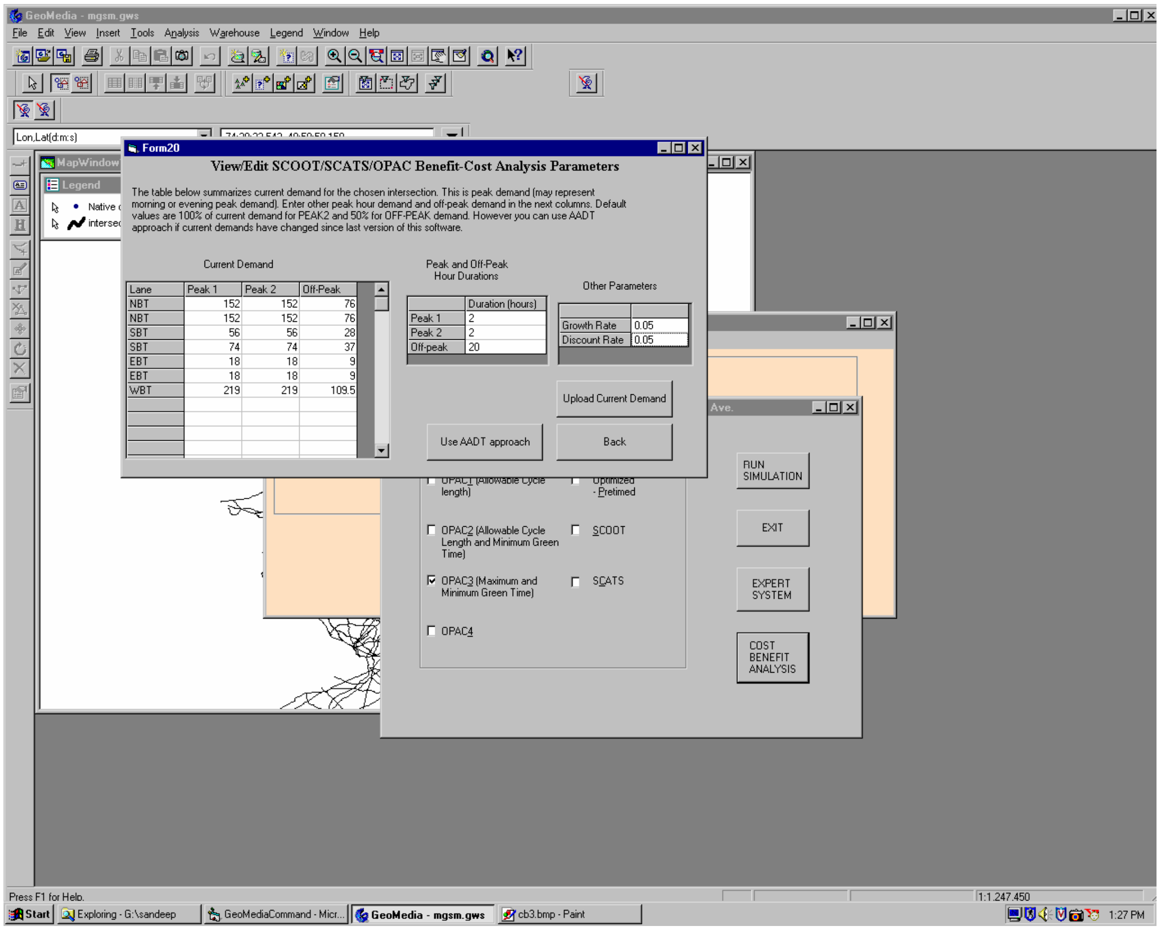
Current Value-of-Time

	$/hour
Value of Time	7.6

☐ OPAC2 (Allowable Cycle Length and Minimum Green Time)
☑ OPAC3 (Maximum and Minimum Green Time)
☐ OPAC4
☐ SCOOT
☐ SCATS

3) View Other CoBA Parameters

This opens a form that prompts to enter new demand, demend periods and growth value. These can be updated to new vaules using the “Update” button.



View/Edit SCOOT/SCATS/OPAC Benefit-Cost Analysis Parameters

The table below summarizes current demand for the chosen intersection. This is peak demand (may represent morning or evening peak demand). Enter other peak hour demand and off-peak demand in the next columns. Default values are 100% of current demand for PEAK2 and 50% for OFF-PEAK demand. However you can use AADT approach if current demands have changed since last version of this software.

Current Demand

Lane	Peak 1	Peak 2	Off-Peak
NBT	152	152	76
NBT	152	152	76
SBT	56	56	28
SBT	74	74	37
EBT	18	18	9
EBT	18	18	9
WBT	219	219	109.5

Peak and Off-Peak Hour Durations

	Duration (hours)
Peak 1	2
Peak 2	2
Off-peak	20

Other Parameters

Growth Rate	0.05
Discount Rate	0.05

☐ OPAC1 (Allowable Cycle length)
☐ Optimized - Pretimed
☐ OPAC2 (Allowable Cycle Length and Minimum Green Time)
☐ SCOOT
☑ OPAC3 (Maximum and Minimum Green Time)
☐ SCATS
☐ OPAC4

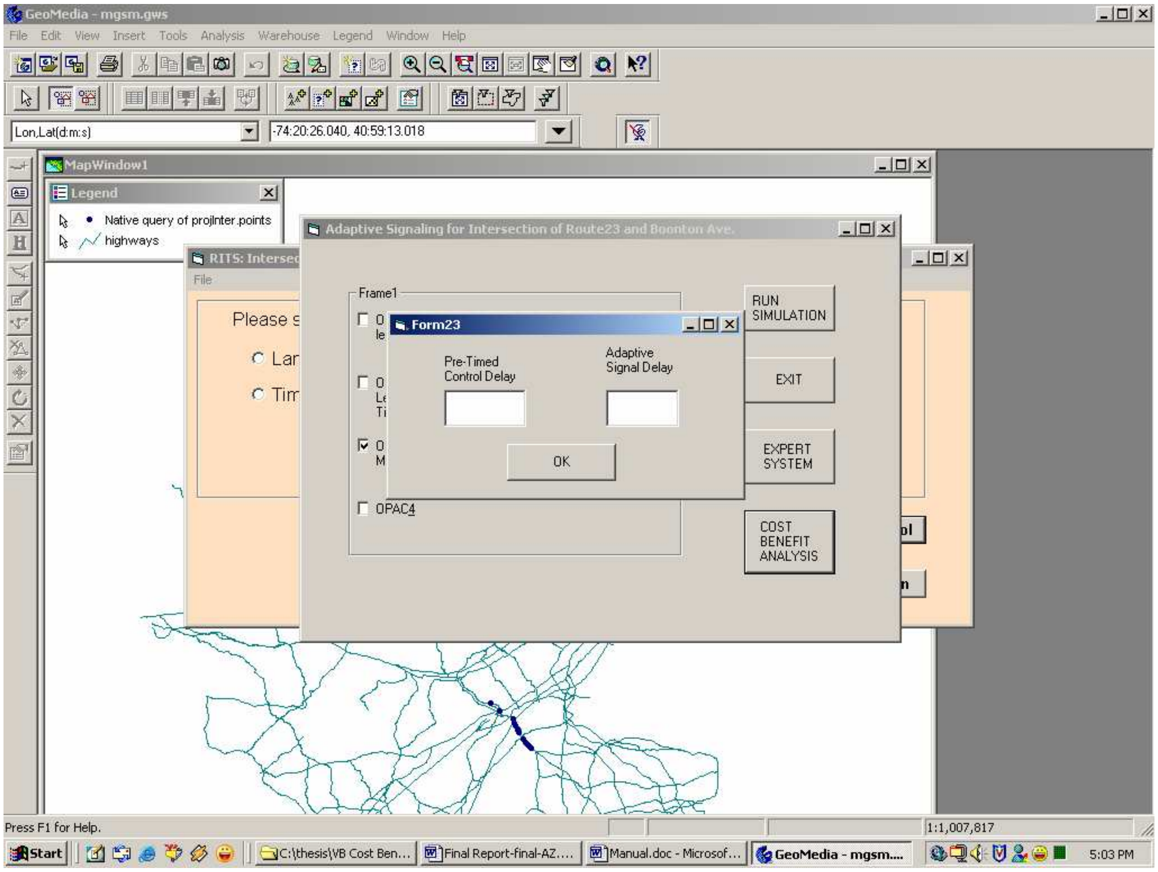
STEP-2

1) Run Cost Benefit Analysis

   This option runs simualtion for cost benefit analysis for a period of 20 years. It increments demand based on growth rate, generates an optimized signal timing plan and sets maximum and minimum green time values for signal control strategy. However this is option is not recommeneded for running with SCATS and SCOOT since they require more time and the system may hang. While running with OPAC, it might give an error, which goes away if Geomedia is started again and simulation is run. This error is due to software limitation that does not allow opening a file used by other application.

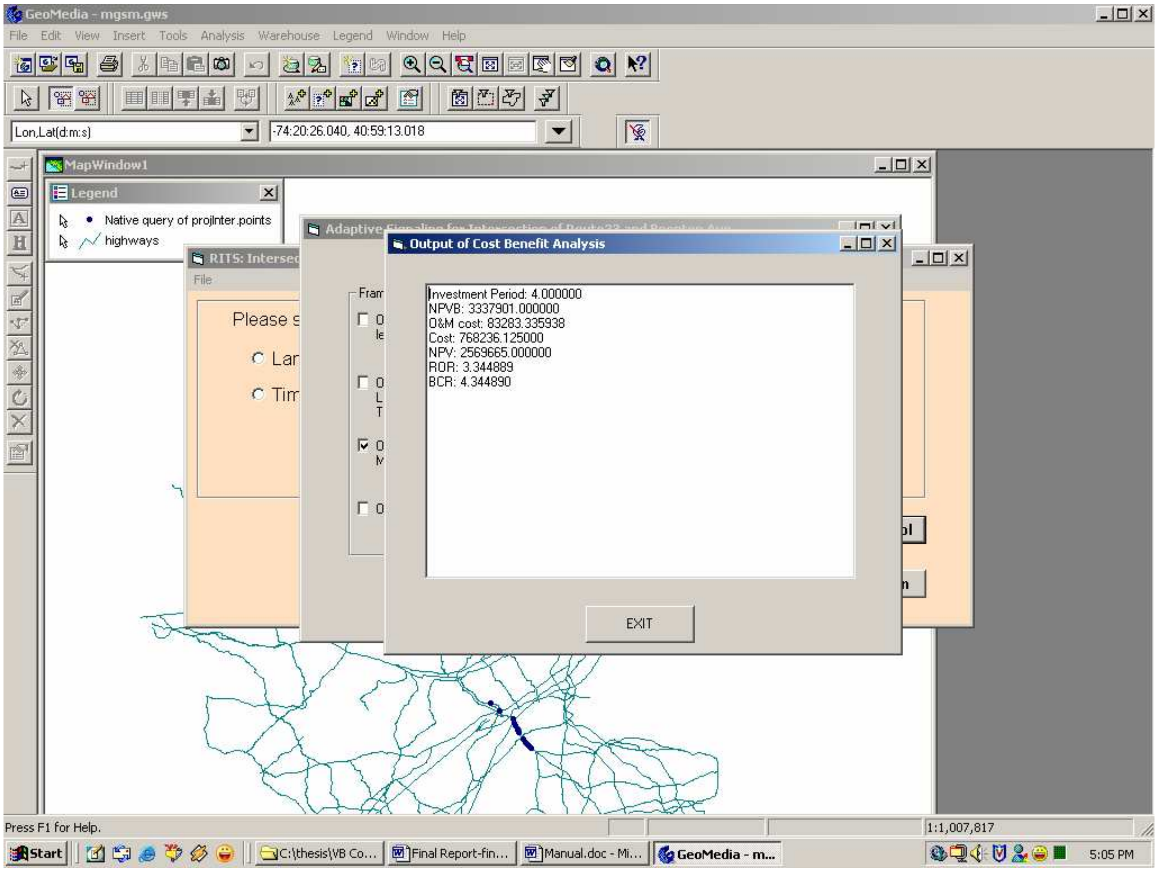
2) Run Cost Benefit Analysis Using Single Run

   This option was specially made for performing faster Cost Benefit analysis. It requires total delay for both adaptive signal strategy and pre-timed signal operation in vehicle hours. It then runs cost benefit analysis based on this data.



STEP-3

The output window appears for the benefit cost analysis after the simulation is completed. It appears as shown below:



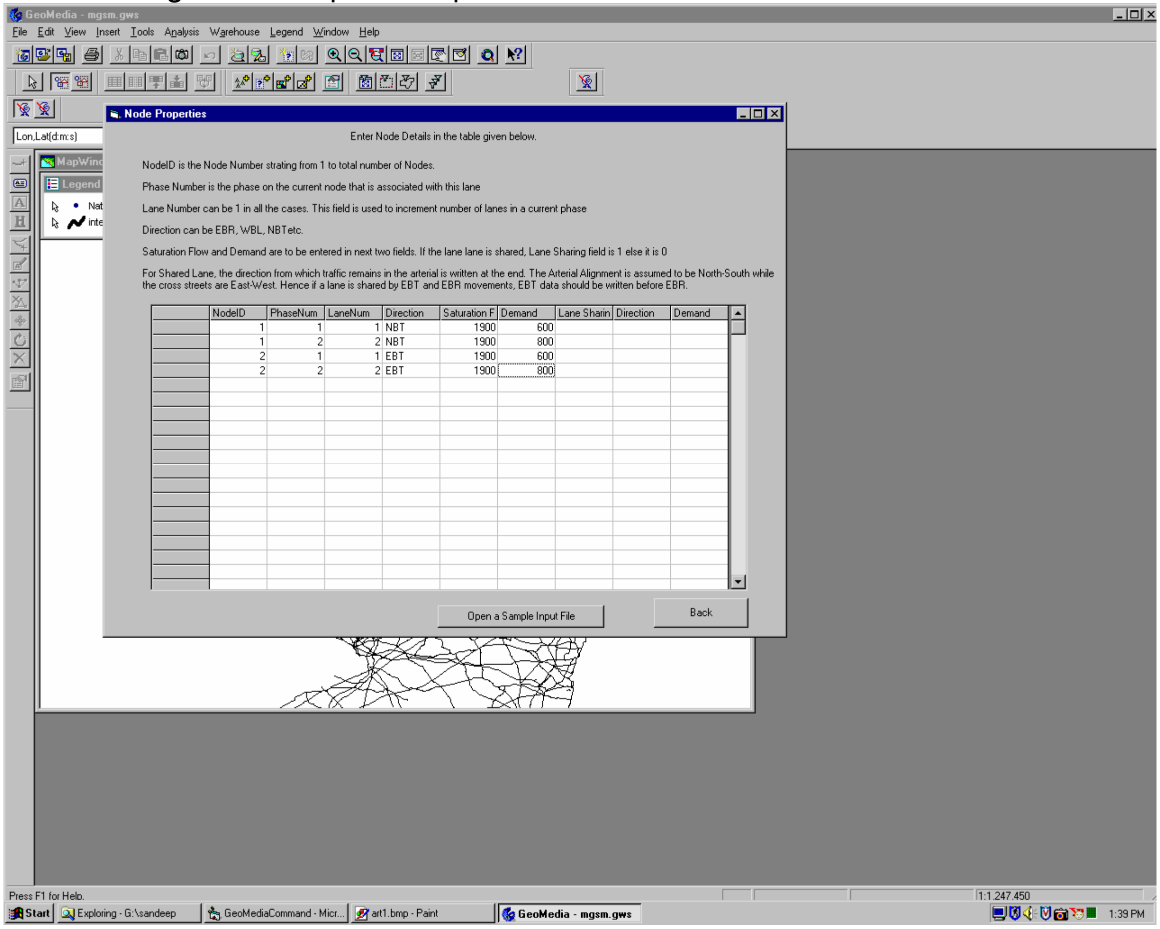
```
Investment Period: 4.000000
NPVB: 3337901.000000
O&M cost: 83283.335938
Cost: 768236.125000
NPV: 2569665.000000
ROR: 3.344889
BCR: 4.344890
```

ARTERIAL VERSION

Arterial version of the adaptive signal program is only for OPAC prototypes. It can simulate an arterial with a maximum of 15 intersections. The arterial version is indepenedent of the intersection selected and uses three files to input data.

STEP 1

The node detail form generates the node file requried for simulation. A sample node file is given to help user input correct values.



Enter Node Details in the table given below.

NodeID is the Node Number strating from 1 to total number of Nodes.

Phase Number is the phase on the current node that is associated with this lane

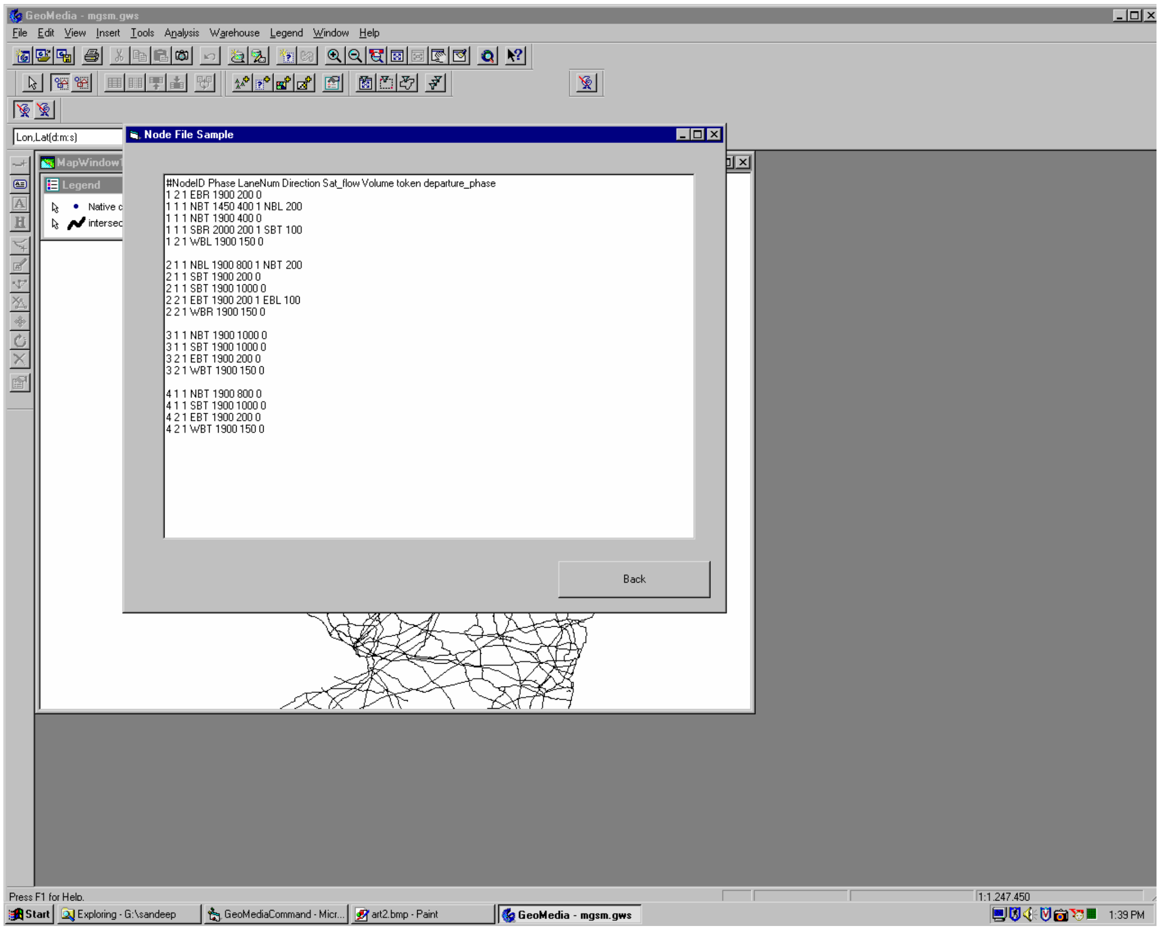
Lane Number can be 1 in all the cases. This field is used to increment number of lanes in a current phase

Direction can be EBR, WBL, NBTetc.

Saturation Flow and Demand are to be entered in next two fields. If the lane lane is shared, Lane Sharing field is 1 else it is 0

For Shared Lane, the direction from which traffic remains in the arterial is written at the end. The Arterial Alignment is assumed to be North-South while the cross streets are East-West. Hence if a lane is shared by EBT and EBR movements, EBT data should be written before EBR.

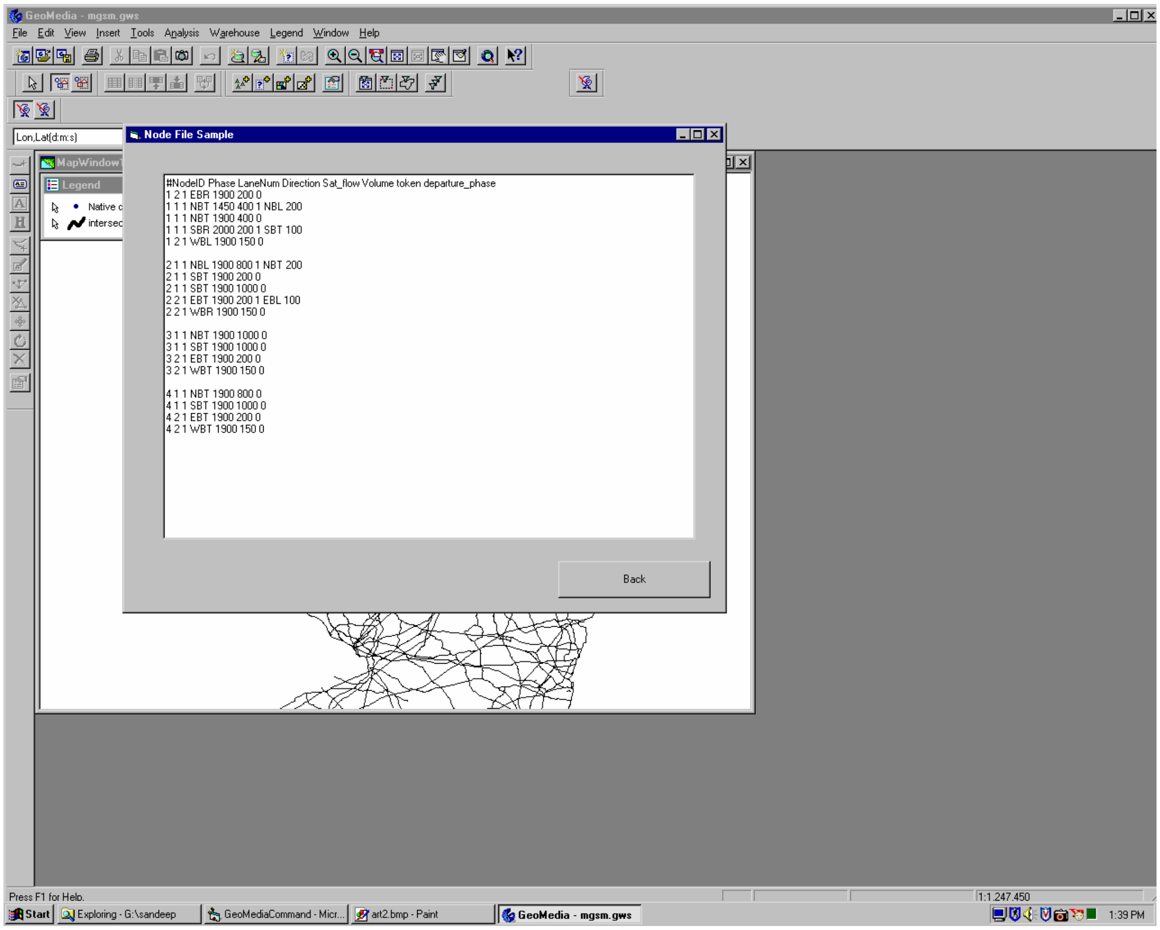
NodeID	PhaseNum	LaneNum	Direction	Saturation F	Demand	Lane Sharin	Direction	Demand
1	1	1	NBT	1900	600			
1	2	2	NBT	1900	800			
2	1	1	EBT	1900	600			
2	2	2	EBT	1900	800			



```
#NodeID Phase LaneNum Direction Sat_flow Volume token departure_phase
1 2 1 EBR 1900 200 0
1 1 1 NBT 1450 400 1 NBL 200
1 1 1 NBT 1900 400 0
1 1 1 SBR 2000 200 1 SBT 100
1 2 1 WBL 1900 150 0

2 1 1 NBL 1900 800 1 NBT 200
2 1 1 SBT 1900 200 0
2 1 1 SBT 1900 1000 0
2 2 1 EBT 1900 200 1 EBL 100
2 2 1 WBR 1900 150 0

3 1 1 NBT 1900 1000 0
3 1 1 SBT 1900 1000 0
3 2 1 EBT 1900 200 0
3 2 1 WBT 1900 150 0

4 1 1 NBT 1900 800 0
4 1 1 SBT 1900 1000 0
4 2 1 EBT 1900 200 0
4 2 1 WBT 1900 150 0
```

STEP_2

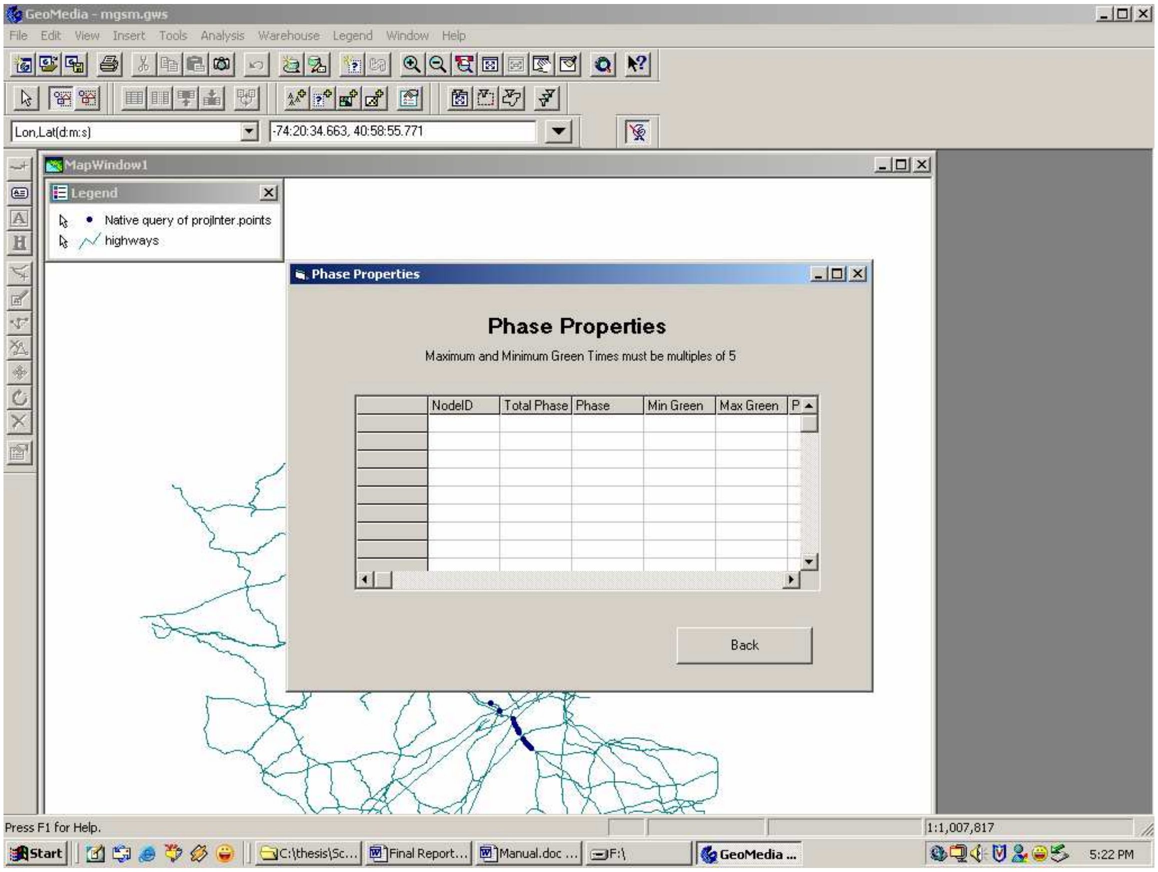
The link detail option asks a user to input the length and speed of the link connecting adjacent nodes



```
#NodeID Phase LaneNum Direction Sat_flow Volume token departure_phase
1 2 1 EBR 1900 200 0
1 1 1 NBT 1450 400 1 NBL 200
1 1 1 NBT 1900 400 0
1 1 1 SBR 2000 200 1 SBT 100
1 2 1 WBL 1900 150 0

2 1 1 NBL 1900 800 1 NBT 200
2 1 1 SBT 1900 200 0
2 1 1 SBT 1900 1000 0
2 2 1 EBT 1900 200 1 EBL 100
2 2 1 WBR 1900 150 0

3 1 1 NBT 1900 1000 0
3 1 1 SBT 1900 1000 0
3 2 1 EBT 1900 200 0
3 2 1 WBT 1900 150 0

4 1 1 NBT 1900 800 0
4 1 1 SBT 1900 1000 0
4 2 1 EBT 1900 200 0
4 2 1 WBT 1900 150 0
```

STEP_3

The phase detail file asks for minimum and maximum green time (in second) for each phase and each intersection in the arterial.

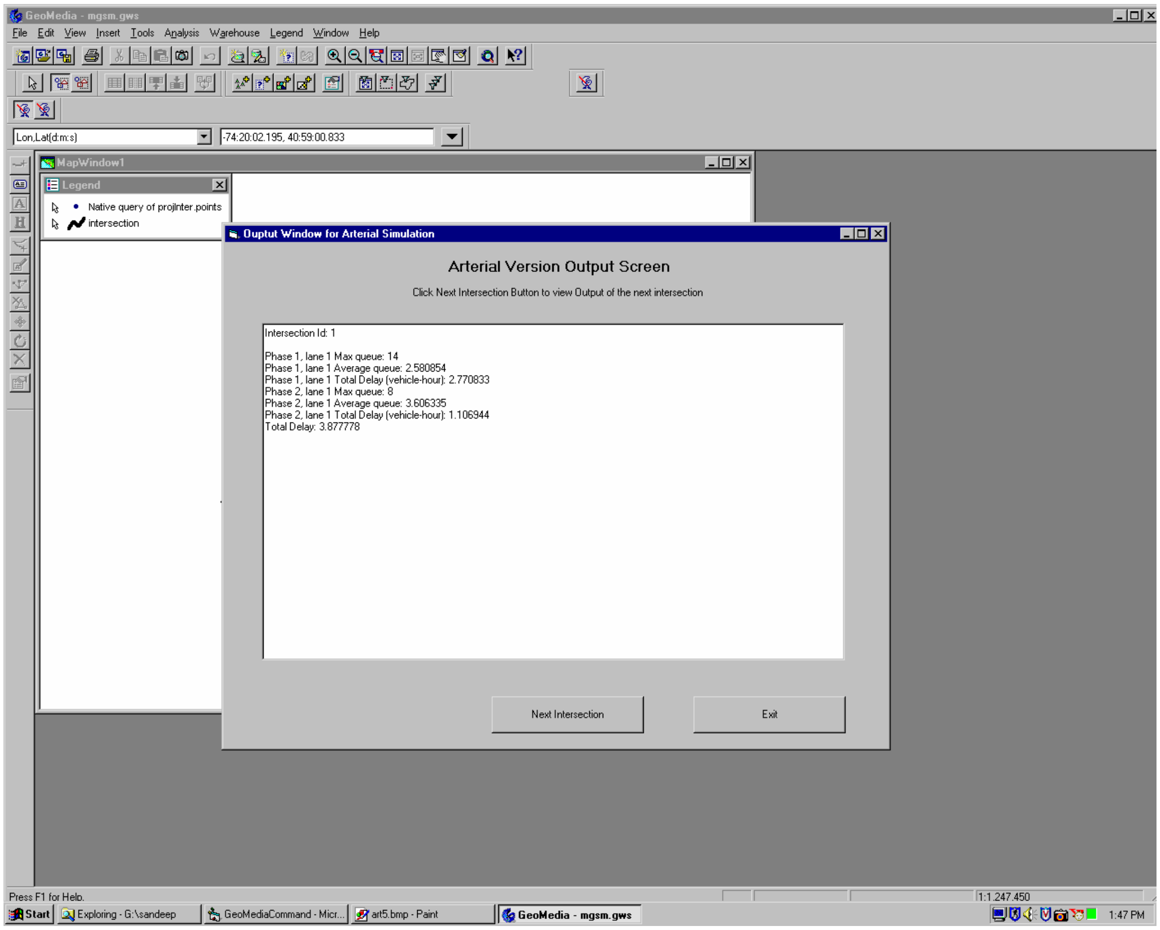


Phase Properties

Maximum and Minimum Green Times must be multiples of 5

STEP-4

The run simulation button starts opac simulation for arterial version and opens an output window, which shows detailed simulation results for each intersection.



```
Intersection Id: 1

Phase 1, lane 1 Max queue: 14
Phase 1, lane 1 Average queue: 2.580854
Phase 1, lane 1 Total Delay (vehicle-hour): 2.770833
Phase 2, lane 1 Max queue: 8
Phase 2, lane 1 Average queue: 3.606335
Phase 2, lane 1 Total Delay (vehicle-hour): 1.106944
Total Delay: 3.877778
```