IMPROVEMENT OF SUPERVISORY CONTROL INTELLIGENT ADAPTIVE MODULE (SCIAM) FOR INTERSECTION SAFETY AND EFFICIENCY

FINAL REPORT

PennDOT/MAUTC Agreement Contract No. 510401
VT-2008-05
DTRS99-G-0003

Prepared for

Virginia Transportation Research Council

By

Montasir M Abbas
Virginia Tech Transportation Institute

May 2010

This work was sponsored by the Virginia Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of either the Federal Highway Administration, U.S. Department of Transportation, or the Commonwealth of Virginia at the time of publication. This report does not constitute a standard, specification, or regulation.
**Abstract**

We continued our studies on the safety and dilemma zone issues at signalized intersection. Dilemma zone is an area where motorists can neither stop before stop line comfortably nor pass the intersection safely at the yellow onset. It is a leading cause for crashes and casualties at intersections. To improve the safety at intersection, the research team deployed a series of researches on how to better protect those vehicles caught in the dilemma zone. We further developed a model to measure the unsafe level of those vehicles in the dilemma zone according to their instantaneous positions and speeds, helping us better understand the driving behaviors and dilemma zone issues at signalized intersections.

**Key Words**

Dilemma zone, driving behaviors, signalized intersections

---

**Distribution Statement**

No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

---

**Security Classif. (of this report)**

Unclassified

---

**Security Classif. (of this page)**

Unclassified

---

**No. of Pages**

47

---

**Price**

Technical Report Documentation Page Reproduction of completed page
# Table of Contents

1. Introduction ......................................................................................................................................................... 4
   1.1 Summary ......................................................................................................................................................... 4
   1.2 Findings ......................................................................................................................................................... 5
   1.3 Benefits ......................................................................................................................................................... 5

2. A New Zone Protection Algorithm Based On the Vehicles’ Trajectories Prediction and Markov Process ......................................................................................................................................................... 7
   2.1 Introduction .................................................................................................................................................. 7
   2.2 Literature Review ....................................................................................................................................... 8
   2.3 Problem statement ....................................................................................................................................... 10
   2.4 Model description ....................................................................................................................................... 11
   2.5 Algorithm deployment in VISSIM ........................................................................................................... 16
   2.6 STUDY INTERSECTION ............................................................................................................................. 18
   2.7 ANALYSIS RESULTS ............................................................................................................................. 20
   2.8 CONCLUSIONS ......................................................................................................................................... 22

   3.1 Introduction ............................................................................................................................................... 25
   3.2 Literature Review ....................................................................................................................................... 26
   3.3 Problem Statement ....................................................................................................................................... 29
   3.4 Algorithm Description .................................................................................................................................. 29
   3.5 Simulation-based Optimization Using VISSIM ........................................................................................ 33
   3.6 Example Study: optimal maximum green for the multi-detector green extension system .................. 34
      3.6.1 Problem statement: .......................................................................................................................... 34
      3.6.2 Traffic scenario: .................................................................................................................................. 35
3.6.3 The output of the stochastic objective function: ................................................................. 37
3.6.4 Stochastic Optimization Design: .......................................................................................... 38
3.6.5 Preliminary Study of the problem properties: ................................................................. 39
3.6.6 Optimization results: ........................................................................................................ 40
3.6.7 Improvement by the optimal maximum green for GES: ................................................ 41
3.7 Conclusions and Future works ............................................................................................ 42
4 VISSIM-based Signal Control and Operation Test-bed Environment (VISCOTE) ............ 46
5. Recommendation for Future Research .................................................................................. 47
  5.1 Application of the Markov Process to the Traffic Signal Systems ................................ 47
  5.2 Simulation-based stochastic optimization ........................................................................ 47
  5.3 VISCOTE Development .................................................................................................... 47
1. Introduction

1.1 Summary

This report is a summary of the research activities during fiscal year 2008-2009. The research is sponsored by Mid-Atlantic Universities Transportation Centers (MAUTC) and its project title is “Improvement of Supervisory Control Intelligent Adaptive Module (SCIAM) for Intersection Safety and Efficiency”. We continued our studies on the safety and dilemma zone issues at signalized intersection. Dilemma zone is an area where motorists can neither stop before stop line comfortably nor pass the intersection safely at the yellow onset. It is a leading cause for crashes and casualties at intersections. To improve the safety at intersection, the research team deployed a series of researches on how to better protect those vehicles caught in the dilemma zone. We further developed a model to measure the unsafe level of those vehicles in the dilemma zone according to their instantaneous positions and speeds, helping us better understand the driving behaviors and dilemma zone issues at signalized intersections.

We provided a systematic method to optimize one of traditional dilemma zone protection systems, the multi-detector green extension system (GES). GES is a widely used dilemma zone protection system but the design of GES has been partially based on engineering judgment and therefore it has potential to be more effective just by optimizing the design. We developed a simulation engine to simulate GES and optimized its detectors’ configuration.

We designed a new dilemma zone protection algorithm using the Markov process. The Markov process is a random process that can be used to model a wide range of systems. However, it has limited application to traffic signal systems. We investigated the possibility of applying the Markov process to the dilemma zone protection.

We also investigated the recent development of simulation-based optimization and introduced the retrospective-approximation concept (RA) into the community of traffic signal systems. It can help answer such questions as “how many simulation efforts are necessary but not excessive to model a true problem”.

During this research, we developed a simulation test-bed for innovative signal control and operation. Using a middleware developed by us, namely VTDatex, we succeeded in setting up real-time communication between a prevailing commercial traffic simulation package, VISSIM,
and our signal controller emulators. We evaluated/tested many innovative traffic signal control algorithms with this test-bed, which used to be tested in significantly simplified and therefore biased simulation environment. We presented our developing work at PTV User Group Meeting 2009.

1.2 Findings

Dilemma hazard model: There are many potential applications for the dilemma hazard model. For example, it can be used as a new safety measure for the DZ protection systems as well as for the determination of optimal clearance interval.

Optimal design for GES: By optimizing the detectors configuration, the new detectors’ configuration for GES can lower both the dilemma hazard and the control delay. The new optimal design for GES was also compared with other traditional designs under the same traffic condition and the results showed that the new GES could substantially improve the effectiveness of dilemma zone protection. The associated findings were presented at TRB 2009.

Markov-process dilemma zone protection algorithm: The simulation results showed that the new algorithm can outperform the prevailing dilemma zone protections and can be even better if more precise data collection technique is available in future. One associated paper will be published in the winter simulation conference in December, 2009.

Simulation-based optimization study: Inspired by the RA concept, we developed a new optimization algorithm which can converge quickly as well as accommodate the requirements by the RA. We recommend that four replications is a sufficient simulation effort for a typical isolated intersection.

1.3 Benefits

All of these researches help us better understand the driving behaviors and dilemma zone issues at signalized intersections. The new Markov-based dilemma zone protection algorithm can be deployed in the field. It is estimated it can provide better protection to vehicles in the dilemma zone.

Based on the above findings, we studied, designed or optimized dilemma zone systems by either optimizing the existing dilemma zone protection systems or designing new systems. Meanwhile,
the state-of-art traffic signal test-bed significantly facilitated future research of innovative signal control algorithms.

The following sections are detailed explanation of the research efforts.
2. A New Zone Protection Algorithm Based On the Vehicles’ Trajectories Prediction and Markov Process

2.1 Introduction

Dilemma zone (DZ) is an area at high-speed signalized intersections, where drivers face indecisiveness of stopping or crossing when presented with yellow indicator. The dilemma zone problem is a leading cause for rear-end collisions and red-light running. According to some survey, there are more than one million crashes at intersections a year and most of them occurred at signalized intersections [1]. As a result, how to effectively protect the dilemma zone at high-speed intersections has been a widely concerning issue among the transportation community.

Markov process (MP) is a stochastic process whose future-state probabilities are determined only by its most recent state [2]. The Markov process has been proved capable of simulating a wide range of systems. However, in the field of the traffic signal control, few MP-based signal control applications were reported. In this paper, we consider the number of vehicles in DZ during the green as a Markov process. In light of this idea, the predicted number of vehicles in DZ, namely state, is calculated with the current state and the state transition matrix. Specifically, using the transition matrix, the new algorithm first predicts the numbers of vehicles in DZ for all the time steps from now until the maximum green. Then it compares the predicted numbers with the current number of vehicles in DZ to decide whether to end the green immediately or extend one more step. This process is repeated until either the algorithm considers the current time step is the best time (fewest vehicles in DZ) to end the green or the green phase reaches the maximum.

In order to elaborate the new algorithm, we structured this paper into three parts:

The first part includes the literature review on the MP applications to the transportation field and discussion how to apply the Markov Process to the dilemma zone protection issues.

The second part explains how the proposed MP-based algorithm works and describes how a VISSIM-based simulation environment was designed in order to test and evaluate the new algorithm;

The last part is to apply the new algorithm to a high-speed signalized intersection at Christiansburg, VA. The geometry of that intersection and its dynamic traffic patterns were modeled into the simulation environment to obtain a close-to-reality traffic network. The
calculation of the current number of vehicles in DZ (current state) is crucial in the new algorithm and we proposed two methods to calculate the current state: the detector-based method and the full knowledge of the vehicles’ information. The detector-based method is to apply the car-following models with the data from advance detectors and predict the vehicles’ trajectories and the number of vehicles in DZ; the full knowledge of the vehicles’ information means detecting the exact number of vehicles in DZ.

2.2 Literature Review

The Markov process has been proved capable of simulating highly stochastic, non-linear traffic systems. A typical Markov control model is composed of four items: state space, control actions, states transition probabilities and reward matrix.

The state space \( X \): it is a \( \text{Borel}^2 \) Space and each element in the space is called state. In the context of the traffic system, the state space is defined to reflect traffic dynamics and it can be, for example, queue length, number of vehicles in DZ or control delay. Previous studies on stochastic traffic dynamics primarily focused on how to improve the mobility and therefore they usually used the number of vehicles (e.g., the queue length or the platoon length) to define the state spaces. Taken as examples, Botma used the number of vehicles in a queue to model the stochastic traffic [3]; Botma’s model was later used in Hoogendoorn’s study on the robust control of stochastic traffic networks [4]; Alfa and Neuts used the number of vehicles in a platoon to define the state space for the random traffic arrival profile [5]; Viti and Zuylen used the number of vehicles in queues as the state space to re-cast the queuing models at signalized intersections [6]; Geroliminis and Skabardonis used the queue length as the state space to model the stochastic traffic dynamics along signalized arterials [7]; Cascetta modeled traffic assignment evolution as a stochastic process using the Markov process. Obviously, the more detailed a state space is modeled, the closer to the reality it is. Excessive details, however, may also dramatically increase a problem’s dimension and result in excessively long computing time. For this reason, in their Markov-process-based adaptive signal control framework, Yu and Recker used a binary state space, congested vs. uncongested, rather than the queue length on each approach to ensure the optimizing algorithm can work fast. When the number of vehicles in a link is greater than a threshold value, that link is marked as “congested”, otherwise “uncongested” [8]. Similarly, Kim
et al. used “congested” and “uncongested” to mark the states for each node of the network in their vehicle routing studies [9].

The possible control actions $A$: it is a Borel Space and defined as the set of all possible controls (or alternatives). Each element $x$ in the state space $X$ is associated with, or results from, a subset of $A$. From the perspective of the traffic signal control, it is usually translated into control strategies. Taken as examples, some well-known control strategies include: TRANSYT-7F [10], SCOOT [11], SCATS [12], OPAC [13], RHODES [14] and TRPS [15]. However, although most signal control strategies implicitly incorporate the traffic’s stochastic features, they are in general based on deterministic models rather than stochastic models. As a result, those traditional strategies sometimes may not function well under highly dynamic traffic patterns. Another issue about the control strategies is how to efficiently optimize the strategies on-line or off-line. Off-line optimization works well when a traffic pattern is repeatable. Some well-known studies include: Robertson used the Hill Climbing optimizing technique in the package of TRANSYT-7 [10]; Nakatsuji and Kaku applied neural networks in their self-organizing traffic control strategies [16]; Park et al. used the Genetic Algorithm to optimize the signal control strategies in TRANSYT-7F (version 8.1) [17] and Abbas et al. used a revised Multiple objective Genetic Algorithm to optimize their traffic responsive signal control framework [18].

On-line adaptive optimal signal control strategies have a higher requirement for computing efficiency. There are two optimizing approaches for adaptive control strategies:

a) Binary choice logic, in which time is divided into successive small steps. Between the minimum green and the maximum green, a binary decision (i.e., end vs. extend current green) is made at each time step. Examples of the binary control logics include the modern vehicle actuated signal control strategies, the stepwise adjustment of timing plans [19, 20]. The binary choice logic considers a short-term future (10sec~30sec) and can be in general optimized fast.

b) Sequential approach, in which the decision-making window is longer (50sec~100sec) than the binary logic. Dynamic programming (DP) is commonly used here. The original dynamic programming may need excessively long time with the increase of the problem dimensions and therefore it is often revised in practice. For example, the well-known adaptive signal control strategies, OPAC [13] and RHODES [14] uses the rolling-horizon and iterative
dynamic programming technique to increase the optimizing speed. Yu et al. used a rapid-calculate version of dynamic programming to obtain the optimal control strategies [8].

The probability measure space: it is typically described in a matrix $P$ and each element $p^k_{ij}$ in $P$ stands for a transition probability from state $i$ to state $j$ under control measure $k$. A stochastic process is called Markov Process if its future probabilities are only determined by its most recent states. In the field of the traffic signal control, most related studies before considered the traffic dynamics (e.g., traffic arrival profile, queue length) as the Markov process. Viti and Zuylen designed a mesoscopic Markov model for queues at controlled intersections [21]; Yu and Recker used the Poisson Process to analyze the queue length changing probabilities in their MP-based adaptive signal control [8]; Geroliminis and Skabardonis used MP to model arrival profiles and queue lengths along signalized arterials [7]; Alfa and Neuts used a discrete time Markovian arrival process to model traffic platoons [5]; Adam et al. used Markov transition matrix in the dilemma zone protection policy studies [22, 23]; Kim et al. used travel time estimation to calculate the transition probabilities between states [9]; Sun et al. used Gaussian Mixture Model (GMM) model and Maximum Likelihood Estimation (MLE) to estimate the transition matrix [24]; Sherlaw-Johnson et al. used the Maximum Likelihood Estimation (MLE) to estimate the transition matrix [25]; Hazelton and Walting used linear exponential learning filter [26]; Gaussian process to derive the transition matrix their traffic equilibrium distribution studies [26].

One-step reward $R$ is the immediate result from the control action. Reward is important for on-line optimization in adaptive signal control strategies. Reward is defined according to the problems. Taken as examples, Yu and Recker assigned a high reward for a control policy if the traffic state transit from the congested [8]; Adam et al. used the number of vehicles in the dilemma zone [22, 23]; Kim et al. estimated the cost proportional to the travel times [9].

2.3 Problem statement

Although previous studies have proved that the Markov process could be applied to many traffic problems, its applications to the dilemma zone protection issues have been limited. Adam et al. designed a dilemma zone protection policy using the reinforcement learning. The policy compares the current number of vehicles in DZ and the predicted numbers of vehicles in DZ to
make decisions (i.e., end the green vs. extend the green). The state transition matrix used in Adam’s paper is stationary and therefore cannot respond to the changing traffic conditions. Meanwhile, Adam’s method needs more comprehensive evaluation.

This paper addresses these issues as follows:

1. Designed a new algorithm to update the state transition matrix periodically using both historical data and the new incoming data.
2. Evaluated the new MP-based dilemma zone protection algorithm under dynamic traffic conditions. If a new signal control algorithm involves complex computing, it will have difficulty in embedding into most commercial traffic simulation packages. As a result, many previous studies evaluated their signal-control algorithms in simplified simulation environments, which is likely to cause bias since many driving behaviors are ignored. Unlike the traditional methods, the new algorithm was directly deployed and evaluated in VISSIM® via a middleware, namely VTDatex.

2.4 Model description

**Calculate the number of vehicles in DZ with advance detectors**

The advance detectors are typically placed 700–1,000 feet upstream of the intersection on each of its major road approaches and used to identify the approaching vehicles’ speeds and types. According to the locations of the advance detectors and the vehicles’ speeds, each vehicle’s trajectory can be projected in the time-space diagram. If a fast vehicle catches its front vehicle in DZ, the fast vehicle decelerates to maintain a safe headway from its front vehicle. After the vehicles’ trajectories are projected, we can calculate the number of vehicles in DZ. In Fig. 2-1, the fast vehicle 2 slows down to maintain a safe headway from vehicle 1 and its dilemma zone is changed accordingly. The vehicles in DZ can be counted at any time point.

Obviously, this assumption is questionable due to the fact that vehicles may change lane to keep their desired speeds. Nonetheless, some previous research concluded that this assumption holds well in moderate or high traffic volumes [27, 28, 29].
Markov-chain-based dilemma zone protection algorithm
The goal of the new algorithm is to determine the best time to end the green so as to minimize the number of vehicles in DZ. Specifically, when the green is between minimum green and maximum green and there are neither trucks in DZ nor queues at the stop lines, the new algorithm will make the decision whether to end or extend the green at each time step. The decision is made according to the observed vehicles in DZ and the predicted numbers of vehicles in DZ in future using Markov matrix.

Let $N_{t0}$ denote the current number of vehicles in DZ at time $t_0$ and $P(t_0) = \begin{bmatrix} p_y(t_0) \end{bmatrix}$ denote the state transition matrix. Then the state transition matrix at time $t_n$ after $n$ time steps is:

$$P(t_n) = \left[ p_y(t_0) \right]^n = \left[ p_y(t_n) \right]$$  \hspace{1cm} (2-1)

And the probabilities in which the state transits from $N_{t0}$ to other states after $n$ time steps are:

$$\Pi^k = \left( p_{N_{t0},1}(t_n), p_{N_{t0},1}(t_n), \sum_j p_{N_{t0},j}(t_n) = 1 \right)$$  \hspace{1cm} (2-2)
Where $p_{N_{i,j}}(t_n)$ is the corresponding element in the state transition matrix $P(t_n)$.

With Eq. (2-1) and (2-2), the predicted number of vehicles at time $t_n$ is:

$$N_t = 0 \times \pi_0(t_n) + 1 \times \pi_1(t_n) + 2 \times \pi_2(t_n) + \ldots \quad \text{Where} \quad \pi_i(t_n) = p_{N_{i,j}}(t_n)$$

(2-3)

If at certain time point, the observed number of vehicles in DZ is less than any predicted number at all future time steps, the algorithm will end the green. Otherwise, the algorithm extends the green one time step. In case that the observed number and certain predicted number are equally minimal, the green will be extended. The rationale is to lengthen the cycle length and reduce the hourly number of vehicles in DZ.

**Update the state transition matrix**

The state transition matrix $[p_{ij}(t)]$ is critical for the prediction. In order to respond to the latest traffic, the matrix needs updating periodically according to the new incoming observations. We applied the *rolling horizon* concept to the matrix updating. The rolling horizon concept is used by operations research analysts in production-inventory control and was introduced into the signal control field by Gartner in his adaptive signal control algorithm, OPAC [13]. OPAC divides time into stages and, in each stage, uses the observation during the “head” time to predict the traffic during the “tail” time. Then OPAC decides the timing plans. The time is then shifted ahead a head time long and this process is repeated. In light of the rolling horizon concept, the new algorithm collects state transitions during the head time of each stage, updates the matrix according to the new data then applies the new matrix during the tail time (Fig. 2-2).

![Figure 2-2 updating Markov matrix using rolling horizon technique](image-url)
Let $\Omega$ denote the finite state space and $X_1, X_2, \ldots$ be the transitions between states. Then $X$ is a Markov chain on $\Omega$ with the transition matrix $\left[ p_{ij}(t) \right]$. The empirical state transition matrix can be derived using the new observations with entries:

$$
\left[ p_n(i,j) \right] = \frac{\sum_{k=1}^{n} I(X_k = i, X_{k+1} = j)}{\sum_{k=1}^{n} I(X_k = i)} (i, j \in \Omega) \quad (2-4)
$$

We reasonably assume that the denominator in Eq. (4) is positive for all the possible states after long observation (i.e., the head time is long enough).

While deriving the transition matrix with the observations, one issue is that the observed transitions within one head time may be biased due to the system’s inherent randomness. To mitigate this bias, the algorithm does not derive the transition matrix merely with the new incoming observations. Rather, it estimates the number of observations using both the historical observations and the new observations. The underlying rationale is that although the traffic is highly dynamic, the traffic pattern is repeated day by day. As a result, the traffic patterns at the same time of different days are similar and associated.

As in Fig. 3, the historical observations were stored in a series of time-dependent matrixes and each cell of those matrixes stands for the transitions between states during certain time. Whenever new observations come in, the algorithm estimates the transitions for each cell using the new incoming transitions, the historical transitions at the same time of a day, the historical transitions head-time units before that time of a day, the historical transitions two head-time units before that time of a day, the historical transitions one head-time units after that time of a day and the historical transitions two head-time units after that time of a day.

Least-square estimation is used and its mathematical expression is as follows:

Let $N_{i,j}(t)$ (variable) denote the estimation of the transitions between $i$ and $j$ at time $t$; $N_{i,j}^{old}(t)$ denote the historical transitions between $i$ and $j$ at time $t$; $N_{i,j}^{new}(t)$ denote the new incoming transitions between $i$ and $j$. Then the estimation using the least-square estimation can be formulized as:
Min \[ \left( N_{i,j}(t) - N_{i,j}^{new}(t) \right)^2 + \left( N_{i,j}(t) - N_{i,j}^{old}(t-1) \right)^2 + \left( N_{i,j}(t) - N_{i,j}^{old}(t-2) \right)^2 \\
+ \left( N_{i,j}(t) - N_{i,j}^{old}(t+1) \right)^2 + \left( N_{i,j}(t) - N_{i,j}^{old}(t+2) \right)^2 \]  

(2-5)

After all \( N_{i,j}(t) \) s are estimated, they are first used to derive the new transition matrix, and then they replace the corresponding data in the historical data matrixes and the approximating function for each cell is updated as well. This method will not only mitigate the possible bias generated during the matrix updating but also prevent the historical data from getting obsolete.

Figure 2-3 illustrates how to update the transition matrix using both historical data and new incoming data.

In summary, the MP-based dilemma zone protection algorithm can be described as in Fig. 2-4:
2.5 Algorithm deployment in VISSIM

The new algorithm was deployed and evaluated in a prevailing microscopic traffic simulation environment, VISSIM [30]. The advantages of VISSIM over other simulation packages include:

1. VISSIM provides the largest flexibility for users to calibrate the driving behaviors and traffic conditions;
2. VISSIM was developed under .NET framework, which brings flexibility for add-on program development;

Figure 2-4 Flow chart of the new MP-based dilemma zone protection algorithm
3. VISSIM provides the best tools for the development of the signal control strategies, such as the NEMA controller emulator, Vehicle Actuated Programming (VAP) language, signal control Application Programming Interfaces (SCAPI), etc.

VISSIM SCAPIs method was used to develop the MP-based signal control emulator in this research. SCAPIs were written in C++ language and the original version of SCAPI controller requires signal control algorithms be embedded into a single dynamic link library (DLL) file. To facilitate the development, we developed a middleware namely “VTDatex”, which can synchronously collect all the real-time detectors/ phases states from the VISSIM network to the control emulator then return the new desired phase states back to the VISSIM network.

The advance detectors are used to collect vehicles’ speeds/types and predict vehicles’ trajectories/the number of vehicles in DZ. Meanwhile, the actual number of vehicles in DZ is also collected through VISSIM Common Object Module (VISSIM COM) to evaluate the new algorithm’s performance.

At each time step, the controller runs the algorithm, makes decisions according to the current state and the Markov state-transit matrix then returns the new desired phase states to the VISSIM network. The concept of this simulation environment is illustrated as in Fig. 2-5.

![Figure 2-5 Illustration of VISSIM Simulation Environment](image-url)
2.6 STUDY INTERSECTION

The study intersection is located on the Peppers Ferry Road and the North Franklin Street at Christiansburg, VA. Its geometry is illustrated in Fig. 2-6. It has high-speed (45 MPH) lanes dedicated to the through traffic on the north-bound and south-bound approaches. Those lanes are our study lanes.

![Figure 2-6 Geometry of the study intersection](image)

**Traffic volumes:** the purpose of this experiment is to make an informal evaluation of the new algorithm under a close-to-reality traffic condition. We counted the traffic volume on the study lanes every 15 minutes with a data acquisition system on the high-speed approaches and the whole counting lasted 9 hours. The through-traffic volumes were plotted as in Fig. 2-7. These volumes were modeled into the VISSIM network to provide a close-to-reality traffic volume pattern. The volumes on other approaches are listed in Table 1.
Figure 2-7 traffic volume profiles on the study lanes

Other VISSIM networks inputs: Table 1 lists all the other necessary inputs for the network modeling except the signal control settings.

Table 2-1 Network Inputs for the VISSIM network

<table>
<thead>
<tr>
<th>Driver Behavior</th>
<th>Model Type:</th>
<th>Reaction to amber Signal:</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic O-D distribution</td>
<td>Wiedemann 99</td>
<td>One Decision Model</td>
<td></td>
</tr>
</tbody>
</table>

Signal control emulator phasing: The NEMA phasing sequence is shown in Fig. 2-8.
Phase 2 and 6 were equipped with the dilemma zone protection system and therefore their possible durations range from the minimum green to the maximum green. Meanwhile, phase 2 and 6 were also equipped with the stop line detectors to ensure the queues were cleared before the green ended. The other phases were set as fixed.

2.7 ANALYSIS RESULTS

We compared the new MP-based algorithm with the detection-control system (DCS). DCS is a DZ protection system developed by Texas Transportation Institute and it also uses advance detectors to predict the number of vehicles in DZ and determine the best time to end the green accordingly. All but the control logic were set exactly the same for DCS and the MP-based algorithm. Simulation was conducted with multiple replications to reduce randomness and the results are as in Fig. 2-9. From Fig. 2-9, the MP-based algorithm caught fewer vehicles in DZ than DCS in both high and moderate traffic volumes.
Figure 2-9 Comparison between the MP-based algorithm and DCS

The detector-based prediction of the vehicles’ trajectories is of low fidelity and therefore may have a negative impact on the algorithm’s performance. To address this issue, we directly input the algorithm with the actual number of vehicles rather than the predicted number from the detectors. The results show that the performance can be significantly improved by replacing the detector-based data collection with the new data collection (Fig. 2-10). In other words, the new MP-based algorithm will have a better performance if the vehicles’ trajectories are totally known.
2.8 CONCLUSIONS

The dilemma zone problem is a leading cause for crashes at intersections and therefore the studies on how to protect vehicles caught in the dilemma zone has been a widely concerning issue. The authors of this paper addressed this issue by applying the Markov process concept to the designing of the dilemma zone protection algorithms. The new algorithm predicts the future states with the Markov state-transit matrix and compares them with the current state. Then it determines the best time to end the green. The Markov state-transit matrix is crucial in the algorithm. In order to respond to the latest traffic, the state transition matrix is periodically updated using rolling-horizon approach.

The authors also compared the new algorithm with another dilemma zone protection system, the detection-control system. The simulation was conducted with multiple replications and results show that the new algorithm has a better performance than DCS in dynamic traffic conditions.

How to calculate the number of vehicles in DZ is another important issue in the new algorithm. We adopted a practical method based on advance detectors and car-following models to address
this issue. However, the detector-based method is of low fidelity and so it may have a negative impact on the overall performance. In order to investigate the impact, we input the algorithm with the actual number of vehicles in DZ and results showed the algorithm’s performance considerably improved if the number of vehicles in DZ can be precisely calculated.

ACKNOWLEDGMENTS

This research was sponsored by Middle Atlantic Universities Transportation Center (MAUTC). The authors are especially thankful to Dr. Vicki Neal and Dr. Zac Doerzaph at Virginia Tech Transportation Institute for providing accessibility to their vehicle data at the intersection.

REFERENCES


3.1 Introduction

Traffic signal control systems have a significant impact on the performance of transportation networks. According to a study by Texas Transportation Institute, the unit cost caused by the control delay is as high as $17.02 per vehicle-hour [31]. As a result, many research efforts have been devoted to how to optimize the traffic signal systems to reduce delays and travel times.

The traditional optimization methodology for the traffic signal systems is first setting up analytical models for the particular problems then seeking the best solutions using optimization techniques. Although the traditional methodology made great achievements, two major issues make it questionable:

(1) The traditional methodology in general ignores driving behaviors (e.g., car following, lane changing). However, these “trivial” driving behaviors turn out to be decisive in many cases.

(2) Many traditional optimization algorithms were designed to optimize deterministic problems and therefore they cannot tackle the inherent randomness in the traffic systems. As a result, the solutions suggested by such optimization algorithms cannot answer such questions as “how robust is the optimal solution?” or “How much likelihood will the optimal solution fail if something unexpected occurs”.

The purpose of this paper is to address these two issues by introducing the retrospective approximation technique into the traffic signal optimization. We structure this paper into 4 parts: in the first part, we review the applications of the stochastic optimization and the retrospective approximation techniques; in the second part, we analyze the major issues residing in the latest practice of traffic signal optimization; in the third part, we designed a new optimization framework based on a commercial traffic simulation package, VISSIM, and the RA technique; in the last part, we conducted the RA-based stochastic optimization to provide the optimal green maximum green for the multi-detector green extension system and the results showed that …..
3.2 Literature Review

Stochastic optimization methods, also called stochastic root-finding methods in other literature, are a family of optimization algorithms which incorporate random elements either in the problem structure (objective function, constraints, etc) or in the algorithms itself (random choice of parameters, etc). There are two major issues in the stochastic optimization issues: (1) how to approximate a stochastic system that can only be observed with random errors; (2) how to search statistically optimal solutions to the random but observable systems by using algorithms with statistical features. In this section, we briefly discuss some important works of stochastic optimization study and their applications in traffic signal systems.

The first stochastic optimization algorithm was developed by Robbins and Munro [32] to solve stochastic equations. It will be relevant to stochastic optimization if the equations are interpreted as the gradient of objective functions. The simple iterative structure is

\[
X_{k+1} = X_k - a_k (\bar{Y}_k - \gamma), \quad (1)
\]

Where \( k = 0, 1, \ldots; X_0 \) is an initial guess of the root \( x^* \), \( \bar{Y}_k = \frac{Y(X_k)}{m}, \{Y(x), \ldots, Y_m(x)\} \) is a set of random samples from the distribution of \( Y(x) \), and \( \{a_k\}_{k=0}^\infty \) is a predetermined sequence of positive constants which satisfies \( \sum_{k=0}^\infty a_k = \infty, \sum_{k=0}^\infty a_k^2 < \infty \). This method is also called Class Stochastic Approximation (CSA). Many related literatures published later are variants of CSA to address such issues as how to increase convergence rates or relax convergence conditions [33, 34, 35, 36].

Sample Average Approximation (SAA) is another technique to solve stochastic optimization problems. The SAA technique was first suggested by Healy and Schruben [37] and later referred to by Rubenstein and Shapiro [38] and Shapiro and Homem-de-Mello [39] to solve stochastic optimization problems. The idea of SAA is straightforward: since the actual simulated-based optimization is difficult to solve, we may solve an approximate problem \( S \) obtained by substituting the true (also unknown) objective function \( G(x) \) by the sampled approximation \( \bar{y}_m(x; \omega) \). \( \bar{y}_m(x; \omega) \) is the realization of the unbiased estimator \( \bar{Y}_m(x) \) of \( G(x) \) and \( \bar{y}_m(x; \omega) \) is generated using the vector of random numbers \( \omega = \{\omega_1, \omega_2, \ldots, \omega_m\} \) and sample size \( m \). In light of SAA, instead of optimizing the original stochastic problem \( (G) \), the approximate deterministic
problem \((S)\) generated with a large sample size can be solved to optimality with proper optimization algorithms. The solution to \(S\) will converge to the solution to \(G\) with probability 1 when \(m \to \infty\).

Nearly all the stochastic optimization studies in the traffic signal systems followed the idea of SAA. Taken as examples, Park et al. first applied the SAA concept by coupling the genetic algorithms with the simulation-based optimization to provide best signal timings under oversaturated traffic conditions [40]. Later Park et al. released a series of related publications on several applications [41, 42, 43]. Stevanovic et al. integrated the genetic algorithm with VISSIM to optimize the signal system and transit priority systems in Park City, UT and Albany, NY [44].

*Retrospective Approximation* (RA), proposed by Chen and Schmeiser [45], is a variant of SAA. Later on, Pasupathy and Schmeiser extended the RA technique [46] to multiple dimensions and Pasupathy investigated how to make the RA algorithm converge fast [47]. RA reflects the latest development of the simulation-based optimization. Instead of generating a single approximate function \(S\), a sequence of approximate functions \(\{S_i\}\) are generated with increasing sample sizes \(\{m_i\} \to \infty\) and solved to decreasing tolerances \(\{\varepsilon_i\} \to 0\). The underlying rationale is that in the early iterations, the approximate objective functions \(\{S_i\}\) are not very close to the true objective function \(G\) due to their small sample sizes. Therefore, the optimization in early iterations is primarily to better understand the problem. For instance, with certain heuristic searching algorithms, the optimum in the early iteration may be used as the initial guess for the next iteration. This measure, consequently, can increase the chance to obtain the global optimum in the next iteration.

The small sample size and large error tolerance can ensure the early iterations will not expend excessive efforts. With the sample size increasing, it is necessary to lower the error tolerance so that the sample path of the optimal solutions in all iterations can gradually approach the true optimum with probability one (Fig. 3-1).
Figure 3-1 conceptual illustration of the RA technique

The following work is necessary when the RA technique is applied:

1. Provide an initial sample size $m$ and a rule for successively increasing $m_i$, $m_i \geq 2$
2. Provide a rule for computing an error-tolerance sequence $\{\epsilon_i\} \rightarrow 0$
3. Provide a rule for stopping the RA algorithm.
4. Provide a mechanism of sampling the simulation system in order to generate the approximate problem $S_i$. The sampling is conducted with $m_i$ independent random seeds then averaging to obtain $\bar{y}_{m_i}(x_1^i, \omega)$, which is defined as above;
5. Select a numerical optimization algorithm (e.g., the Genetic Algorithm, the Simulated Annealing Algorithm) to search the optimal solution $x_i^*$ to the approximate deterministic problem $S_i$ to the error tolerance $\epsilon_i$. 
The iterations continue until the preset stopping criteria are satisfied. It is also worth noting that the RA framework is very general and therefore we have the flexibility of selecting the appropriate algorithms according to the problem.

### 3.3 Problem Statement

The loss of driving behavior in the analytical model has been well addressed in previous studies. The latest research efforts can directly connect the commercial traffic simulation packages, which can mimic vehicles’ various driving behaviors, with the optimization algorithms. Nevertheless, the converging rate of the stochastic optimization has been a major issue in practice. For example, the commonly used genetic algorithm requires a huge amount of time to obtain the “acceptable” solution. When dealing with large-scale traffic networks, the required computing time may be too long to afford. To address this issue, the state-of-practice methods either used small sample sizes, such as five repetitions, to approximate the objective function or decreased the population size/generations. Although these measures can help maintain an acceptable computing time, the globality and unbiasness of the solutions may be compromised.

As for the robust analysis of the optimal solutions, there are only limited efforts devoted to developing optimization algorithms that can provide robust analysis. Although the prevailing genetic algorithm can provide improved solutions, it does not converge to the global optimum. Meanwhile, the genetic algorithm was originally developed for deterministic problems and therefore it cannot efficiently analyze the robustness of the solutions on its own.

We address these issues by applying the latest RA technique to the optimization of traffic signal systems. The RA technique provides a mechanism to conduct the robust analysis as well as the flexibility of designing/adjusting proper optimization algorithms that converges fast.

### 3.4 Algorithm Description

In light of the RA technique, the optimization algorithms can be selected according to the problem properties. The proposed optimization algorithm in this section is designed specifically for the later example problem and the readers can definitely either expand this algorithm or design their own optimization algorithm following the same procedure.
Initialization: the sample size and error tolerance for the first iteration are set as $m_1 = 1$ and $\varepsilon_1 = \frac{1}{\sqrt{m_1}}$.

Rules for successive increasing sample size and decreasing error tolerance:

$$m_i = \text{RoundUp}[(1+10\%)m_{i-1}], (i \geq 2)$$

and $\varepsilon_i = \frac{1}{\sqrt{m_i}}$.

Self-learning simulated annealing algorithm: the simulated annealing algorithm (SA) is a global optimization algorithm based on random search in the state space [48] and it is used to obtain the optimum in each RA iteration. Each step of the SA algorithm replaces the current solution by the best random "nearby" solution, which is selected with the probability proportionally to a global parameter $T$ (called the “temperature”) and the difference between the corresponding objective function values. $T$ is gradually decreased during the process. When $T$ is large, the current solution may change almost randomly. However, when $T$ keeps decreasing to zero, the new solution will approach the local optimum with certainty.

The converging rate of the SA algorithm is sensitive to its parameters and may be rather slow if the parameters are poorly set. This is a major reason why it has been underutilized. In this paper, we customized the simulated annealing algorithm to increase the converging rate and accommodate the requirements by the RA technique. Rather than the “pure” random search, the new variant of the SA algorithm in a particular iteration will learn from the previous iteration where the global optimum is more likely to be and therefore expend more searching efforts in that area. The rationale is that the sequence of approximate functions with increasing sample sizes should have similar pattern. In other words, the optimal solutions obtained from the different approximate functions are supposed to be close. The pseudo code is as follows:
T=T_0, X=X_0 \quad //\text{Choose initial values for temperature } T \text{ and current solution } X

\text{While (TRUE)}
{

\text{X}_n[10]=\text{NewSolutionGenerate}(); \quad //\text{generate 10 new solutions with the empirical pdf from last RA iteration}

\text{For each } \text{X}_n \text{ in } \text{X}_n[10] 
{

\text{If } F(\text{X}_n)<F(\text{X}) \quad // F(x) \text{ is the sample average approximate function(sample size m)}

\quad \text{X}=\text{X}_n \quad //\text{current solution is replaced with the new (better) solution}

\text{else}

\quad p=\text{UniformRand}(); \quad //p \text{ is a random number uniformly distributed between 0 and 1}

\quad \text{If } \exp \left(\frac{F(\text{X})-F(\text{X}_n)}{T}\right)>p \quad //\text{current solution is replaced with the new (worse) solution with certain accepting probability}

\quad \quad \{ \text{X}=\text{X}_n \}

\text{If (X no longer changes)}
{

\quad \text{Return X}

\}

\text{T=0.95*T} \quad //\text{cooling schedule of the annealing}
}

In a particular RA iteration, all the attempted solutions during the random search are ranked in ascending order according to their corresponding objective function values. Those solutions with less than 80th percentile objective function value are selected as “better” solutions and the nearby areas around them are more likely to have the global optimum. We can ascertain where the SA algorithm should put more search efforts in the next iteration according to the empirical distribution of the “better” solutions. In other words, we should put emphasis on searching where more “better” solutions are. This idea is illustrated in Fig. 3-2.
Now we are ready to describe the RA-based stochastic optimization method, which is shown in Fig. 3.
The new algorithm was coupled with a commercial microscopic traffic simulation environment, VISSIM [30]. The advantages of VISSIM over other simulation packages include:

1. VISSIM provides the largest flexibility for users to calibrate the driving behaviors and traffic conditions;
2. VISSIM was developed under .NET framework, which brings flexibility for integrating the optimizing algorithm with the simulation engine;

The optimization algorithm sends each candidate solution into VISSIM then drive the simulation via VISSIM COM. After each simulation run, the output is sent back to the optimization algorithm. Then the optimizing algorithm will determine the search direction accordingly. Fig. 3-4 illustrates the VISSIM-based stochastic optimization.
3.6 Example Study: optimal maximum green for the multi-detector green extension system

In this section, we conducted the optimization for the dilemma zone protection issues to illustrate the potentials of the RA-based stochastic optimization.

3.6.1 Problem statement: Dilemma zone (DZ) is an area at high-speed signalized intersections, where drivers can neither cross safely nor stop comfortably at the yellow onset. The dilemma zone problem is a leading cause for crashes at intersections and therefore how to protect vehicles in DZ is a widely concerning issue. The multi-detector green extension system (GES) is a widely used dilemma zone protection system. Its mechanism is to hold the green using multiple advance detectors until there are no vehicles caught in DZ. Meanwhile, to avoid having excessively long queues on the conflicting approaches, the GES-equipped phases have maximum greens. When the maximum green is reached, the green will end regardless of the number of vehicles in DZ (i.e., max-out). The longer the maximum green is, the more likely the green is to end before max-out. As a result, the maximum green setting will have impact on the performance of GES. In this section, we conducted an illustrative study using the RA technique to provide an optimal maximum green for GES.

Significance of the problem: there is neither analytical model to describe GES nor systematic method to optimize the maximum green for the GES-equipped phases. As a result, the maximum
green is often calculated indirectly. A typical method is to calculate the phase splits under pre-timed signal control then derive the maximum green for the GES-equipped phases accordingly. The HCM model is commonly used to calculate the phase splits of fixed timing plans [49]:

\[
G_0 = \frac{v^*C}{S*N*(\frac{v}{c})}
\]

(4)

Where:

- \(G_0\): Maximum green (sec);
- \(v/c\): Ratio of volume to capacity, the design objective and 0.95 is used in this paper;
- \(C\): Cycle Length (sec);
- \(S\): Saturation flow (1,600 vehicles per hour per lane in this paper);
- \(N\): Number of lanes;

The suggested maximum green needs to be longer than \(G_0\) to ensure the low max-out probability. Taken as an example, the prevailing GES design by Bonneson [50] suggests that the maximum green should be at least 1.5 times longer than \(G_0\). In practice, many GES designs follow this suggestion. Nonetheless, this method is partially from engineering judgments and thus it may not guarantee the optimum. Meanwhile, since we cannot set up an analytical objective function for this problem, we can only approximate the objective function through simulation sampling and deploy simulation-based optimization.

3.6.2 Traffic scenario: the GES design needs to reflect the local traffic scenarios. Fig. 3-4 shows the geometry of the local intersection at Christiansburg, VA and the traffic volumes.
Figure 3-5 traffic scenario used in optimal maximum green design

**Signal settings:** phase 2 is equipped with GES and the other phases are fixed. We selected 120 seconds as the cycle length to calculate the green length for phase 1, 3 and 4. The yellow interval and all-red clearance are set as 3 seconds and 2 seconds respectively. According to Eq.4 and Fig.5, the green lengths for the fixed phase 1, 3 and 4 are $G_1=10$ seconds, $G_3=31$ seconds and $G_4=21$ seconds. We increased $G_1$ to 15 seconds to meet the minimum green requirement. The possible maximum green of the GES-equipped phase can in theory range from 15 seconds (minimum green) to infinitive. However, in practice, excessively long cycle length may make drivers disrespect the signal and therefore we allowed the possible maximum green of phase 2 to range from 15 seconds to 100 seconds.

**Advance detector configurations:** we used the “Constant-Speed” detectors configuration suggested by Bonneson et al. [50]. The locations and extensions of the advance detectors are listed in Table 1:
Table 3-1: Bonneson’s Constant-speed advance detectors configuration (45 MPH design speed)

<table>
<thead>
<tr>
<th>Detector ID</th>
<th>Position (feet)</th>
<th>Extension (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>445</td>
<td>0.9</td>
</tr>
<tr>
<td>2</td>
<td>364</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>283</td>
<td>3.1</td>
</tr>
</tbody>
</table>

3.6.3 The output of the stochastic objective function: The traffic volumes, the controller settings and advance detectors configuration are all modeled into VISSIM. The VISSIM simulation engine is used to sample and approximate GES. There are two variables in the VISSIM network: the maximum green for phase 1 and the random seed. For one particular maximum-green value, the VISSIM network is sampled with various random seeds then averaged to approximate the objective function value.

Both delay and safety are taken into account. The unit delay cost is about $17.02/hour/vehicle [31]; the safety is evaluated with the hazard caused by the dilemma zone. The number of vehicles in the dilemma zone is traditionally used to measure the safety but this measure implies the vehicles in the dilemma zone are equally unsafe regardless of their speeds and locations. As a result, we used a traffic-conflict-based dilemma hazard model to measure each vehicle safe level at the yellow onset. In light of the dilemma hazard model, the unsafe level, namely dilemma hazard, of the vehicle in the dilemma zone primarily depends on the its time to intersection (TTI) at the yellow onset [51]. Each caught vehicle’s TTI-dependent dilemma hazard can be calculated with Eq. 5 [51].

\[
H = -0.202 * TTI^2 + 1.565 * TTI - 2.218
\]  

(5)

Previous studies also concluded that the probability that a traffic conflict becomes a real accident is about 0.0001 [52], and the average cost for each real accident is $56,706 [53]. Therefore the unit cost for dilemma hazard is approximately $5.67.

When converted to monetary values, the composite outputs from simulations can be calculated as:
\[ Y = 5.67 \times \text{DilemmaHazard(safety)} + \frac{17.02}{3600} \times \text{delay} \]  

(6)

### 3.6.4 Stochastic Optimization Design:

The proposed algorithm was customized to solve this one-dimension optimization problem as follows:

- **Initialization for the 1st RA iteration:**
  - Initial maximum green (initial guess) \( x_0 \): 57 seconds (calculated with the Bonneson’s method);
  - The initial state probability \( P = P_0 \): uniformly distributed between 15 seconds and 100 seconds;
  - Initial sample size for VISSIM: 1;

- **Rules for the RA progressing:**
  - From the 2nd RA iteration, the sample size \( m_i \) is increased as
    \[ m_i = \text{RoundUp}[(1 + 10\%)m_{i-1}] (i \geq 2) \; ; \]
  - Decreasing error tolerance: \( \varepsilon_i = \frac{1}{\sqrt{m_i}} \; ; \)
  - The Common Random Numbers (CRN) rule applies;
  - When \( m_i \) reaches 5, the whole RA iterating ends;

- **Inheritable Markov Monotonic Search Algorithm in each RA iteration:**
  - Fig. 6 illustrates the optimizing process of IMMSA;
3.6.5 Preliminary Study of the problem properties: Before the optimization was deployed, we used small simulation replications to investigate the composite costs under various maximum greens. The maximum greens were selected from 15 seconds to 100 seconds with half-second increments (Fig. 3-7). Although such selection is of low fidelity due to its large increments, it will help us better understand the problem properties as well as validate the results from the optimization. Meanwhile, it is clear that the two approximate curves in Fig. 7 show very similar pattern and their optimum should be very close. This finding is supportive to the speculation in the proceeding section.
3.6.6 Optimization results: Fig. 8 shows how the optimal solution evolved while the RA-based stochastic optimization was progressing and Fig. 9 shows how the generated composite cost by VISSIM decreased. From Fig. 8, we can conclude that increasing the sample size (i.e., increase the approximate efforts) can help reduce the necessary optimization efforts because the large sample size will lower the random noise residing in the simulation-based objective functions. For instance, in Fig. 8, the 1st RA iteration needs 13 random searches to reach the global optimum whereas the search efforts considerably decreases when the sample size increase to 2~5.

Fig. 3-8 and 9 also show that the generated composite cost under the same maximum green value may change due to the increase of the sample size in VISSIM. However, when the sample size becomes large enough, such difference will be negligible. From Fig.9, the optimal maximum green value (solution) from the previous RA iteration always generates a higher composite cost when the sample size of VISSIM is increased in the next RA iteration. However, such increase is nearly negligible when the sample size increases from 3 to 4 or from 4 to 5. As a result, Fig. 9 can answer the question that how many approximate efforts are necessary but not excessive when the simulation-based optimization is deployed. For the problem in this paper, sample size 4 is sufficient since the optimum in the 4th RA iteration increases the composite output less than 1% and the optimum stays unchanged when the sample size increase from 4 to 5 (Fig. 3-9).
3.6.7 Improvement by the optimal maximum green for GES: Since we have concluded that the sample size 4 or higher is sufficient (and thus valid) for the approximate efforts, we only compared the optimal maximum green (99 seconds) obtained in the 4\textsuperscript{th} and 5\textsuperscript{th} RA iterations with the maximum green value (57 seconds) calculated with the Bonneson’s method. The comparison
is listed in Table 3-2 and it shows that the new optimal maximum green can significantly lower the hourly composite cost compared with the traditional maximum green.

Table 3-2 Improvements by the optimal maximum green

<table>
<thead>
<tr>
<th>Sample Size</th>
<th>Hourly composite cost under the optimal maximum green (USD)</th>
<th>Hourly composite cost under the Bonneson's maximum green (USD)</th>
<th>Improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>828</td>
<td>968</td>
<td>14.46%</td>
</tr>
<tr>
<td>5</td>
<td>844</td>
<td>944</td>
<td>10.59%</td>
</tr>
</tbody>
</table>

3.7 Conclusions and Future works

Most of traffic signal systems have to be optimized through traffic simulation samplings. The state-of-practice method is to approximate the true (and unknown) problem with a single approximate problem then optimize the approximate function. The authors in this paper introduce the Retrospective Approximation (RA) concept into the optimization of traffic signal systems. RA reflects the latest theoretical development of the simulation-based optimization and is able to synchronize the approximate efforts and optimization efforts when the simulation-based optimization is deployed. When the approximate efforts are low, RA will lower the optimization efforts and put emphasis on the better understanding of the system; when the approximate efforts are gradually increased, RA will also increase the optimization efforts to obtain the global optimum.

Based on the RA concept, we developed a stochastic optimization engine using VISSIM and applied it to the maximum green setting issue for the multi-detector green extension systems. The result shows that the sample size 4 is sufficient for VISSIM to approximate a typical multi-detector green extension system and the optimal maximum green for the GES-equipped phase at the intersection of Christiansburg, VA is 99 seconds. The new design can significantly decrease the hourly composite cost (safety and delay cost) compared with the traditional value.

In the future, we are planning to extend the IMMSA algorithm to multiple dimensions and further improve the converging rate. We are also planning to integrate other prevailing traffic simulation engines with the optimization engine developed in this paper and investigate how adaptive those simulation engines in the market will be when the simulation-based optimization is deployed.
REFERENCE


4 VISSIM-based Signal Control and Operation Test-bed Environment (VISCOTE)

The author designed the framework of VISCOTE (Fig. 4-1). VISCOTE is not an executable program. Rather, it is a conceptual solution with five modules aiming to facilitate evaluating/emulating state-of-art signal algorithms in VISSM. Readers can apply the concept of VISCOTE to their development of signal emulator or evaluating tools in VISSIM. Like most commercial microscopic simulation software, VISSIM does not provide an easy way to deploy/evaluate state-of-art signal algorithms. Nonetheless, VISSIM does provide interfaces for research users to do so. In this chapter, the author explained how to associate the five modules in VISCOTE with the research on the signal control.

![VISCOTE Framework Diagram](image-url)

Figure 4-1 VISCOTE framework
We have applied the VISCOTE concept to a couple research topics. And we plan to enrich the framework with more research efforts

5. Recommendation for Future Research

5.1 Application of the Markov Process to the Traffic Signal Systems

Markov process is a commonly used random process. The previous studies proved it can model a wide range of systems. In the field of traffic signal systems, the Markov process has been underutilized. Besides the dilemma zone issues, we plan to investigate the potential of applying the Markov process to other related research.

5.2 Simulation-based stochastic optimization

The state-of-practice optimization efforts in traffic signal systems can be classed as “sample average approximation”, which does not reflect the latest development of simulation-based simulation. We plan to collaborate with the researchers in other fields to further investigate how to optimize the signal systems with simulation. For example, develop more efficient optimizing algorithm.

5.3 VISCOTE Development

Extend VISCOTE framework from isolated intersections to coordinated intersections.