wctr'04 4-8 july 2004 / istanbul-turkey

ADAPTIVE CONTROL STRATEGY FOR A CO-ORDINATED TRAFFIC SIGNAL NETWORK: THE VIRTUAL-FIXED-CYCLE APPROACH

Nathan H. Gartner
Department of Civil & Environmental Engineering, University of Massachusetts
Lowell, MA 01854, U.S.A.
Nathan_Gartner@uml.edu

Abstract

OPAC (Optimization Policies for Adaptive Control) was the first strategy to be developed in the U.S. for real-time traffic-adaptive control of signal systems. The strategy concentrated first on individual intersection control. This paper presents an extension of this strategy for co-ordinating and synchronizing signals in a network using the virtual-fixed-cycle concept. It presents the theoretical basis of the VFC-OPAC algorithms and describes the implementation and field testing of OPAC within the RT-TRACS system. The Real-time Traffic Adaptive Control System (RT-TRACS) is a state-of-the-art system in advanced traffic signal control which was sponsored by the Federal Highway Administration (FHWA) of the U.S. Dept. of Transportation.

Keywords: Intelligent transportation systems; Traffic signal network; Virtual-fixed-cycle concept; Optimization policies for adaptive control; Real-time traffic adaptive control system
Topic Area: C3 Traffic Control

1. Introduction

In the early 90's the U.S. Federal Highway Administration (FHWA) set out to advance the state-of-the-art in Intelligent Transportation Systems (ITS) by initiating "a program of research, development and operational tests designed to combat traffic congestion." One of the key elements of this program was "to develop and field evaluate a real-time, traffic-adaptive signal control system," which became to be known by its acronym RT-TRACS (FHWA, 1991). The objective of the RT-TRACS project was the development of a system capable of adapting to fluctuating traffic conditions by selecting an optimal control strategy from a "suite" of real-time traffic signal timing control strategies. RT-TRACS serves as a platform for the implementation of a variety of traffic signal control algorithms, including new adaptive algorithms as well as existing signal timing systems.

The first version of RT-TRACS incorporating the coordinated OPAC real-time adaptive algorithm was implemented in a network of 16 intersections on Reston Parkway in Northern Virginia in the Spring of 1998. Type 2070 traffic controllers were employed to control a network of signals under both coordinated and isolated modes of the OPAC adaptive algorithm as well as under Time Base Coordination (TBC). Integration of these technologies into an operating traffic management and signal control system involved a number of technical challenges and institutional issues.

This paper describes a new adaptive control strategy for co-coordinating and synchronizing signals in a network using the virtual-fixed-cycle concept that was developed and implemented in RT-TRACS. The strategy is based on the single intersection dynamic-programming-based OPAC (Optimization Policies for Adaptive Control) adaptive controller that was previously developed and tested (Gartner, 1982b and

1983); hence, it is labeled *VFC-OPAC*. *VFC-OPAC* consists of a distributed control strategy featuring a dynamic optimization algorithm that calculates signal timings to minimize a performance function of total intersection delays and stops. The algorithm uses a combination of measured and modeled demand to determine, in a distributed manner, phase durations at each signal that are constrained by minimum and maximum green times and, when running in a coordinated mode, by coordination and synchronization parameters that can be updated based on real-time data. The paper also discusses field implementation issues and findings of the Reston Parkway field research test.

2. OPAC adaptive control algorithm

OPAC was developed from the outset as a distributed strategy featuring a dynamic optimization algorithm for traffic signal control without requiring a rigid, fixed cycle time. Signal timings are calculated to directly minimize performance measures, such as vehicle delays and stops, and are only constrained by minimum and maximum phase lengths and, if running in a coordinated mode, by a virtual cycle length and by a virtual offset. Development of the OPAC strategy was based on the following principles (Gartner, 1982a and 1985):

a. *The strategy must provide better performance than off-line methods*. Although this principle may seem self-evident, it is not always recognized in the development of responsive strategies and many such strategies fail on this account.

b. *The strategy must be truly demand-responsive*, i.e., it must adapt to actual traffic conditions and not be responsive to historical or predicted values that are unreliable and may be far off from reality.

c. *The strategy must not be restricted to arbitrary control periods* (e.g., 10, 15 mins, etc) but should be capable of providing continuously optimized controls. Effective responsive-ness cannot be achieved merely by implementing off-line methods at shorter intervals.

d. *Development of new control concepts that are better suited to the variability in traffic flows is required*, rather than the extrapolation of existing concepts. The conventional notions of cycle time, splits and offsets, which are inherent in traditional signal optimization methods and were established to support the development of fixed-time timing plans, are not well suited for adaptive control. The premise is that direct on-line minimization of performance measures based on real-time information can provide much improved performance.

e. Finally, *the strategy should not be encumbered by a rigid network structure*; rather, it should be based on decentralized decision-making. This implies that there need not be a different model for different network configurations (e.g., intersections, arterials, grids, networks, etc.). By employing a flexible decentralized decision-making logic one should be able to address a variety of configurations in an optimal manner. Furthermore, the logic should be scalable to facilitate gradual expansion of existing systems.

Development of the strategy has progressed through several versions, each one serving as a stepping stone for a subsequent version. The principal features of each version are outlined below.

2.1. OPAC-1: Dynamic programming

The first version, designated OPAC-1, was designed to serve as a basis for subsequent OPAC strategy development. OPAC-1 utilizes a Dynamic Programming (DP) model for the deter-mination of the traffic control parameters. Since DP is a global optimization strategy for multistage decision processes, it provides a standard against which all other strategies can be compared (Bellman and Dreyfus, 1962).

The optimization process is decomposed into N stages, where each stage corresponds to a discrete time interval in which the arrivals are measured, (2 to 5-sec intervals). The total number of stages N corresponds to the *horizon length* (HL) of the input predictions. For exploratory purposes the horizon length was assumed to be several minutes long, e.g., 5, 10, or 15 mins. A typical stage i is illustrated in Figure 1. At stage i we have an input state vector I_i, an arrival vector A_i, output state vector O_i, input variable x_i, economic return (cost) output r_i, and a set of transformations:

$$O_i = T_i\ (I_i,\ A_i,\ x_i) \qquad (1)$$

$$r_i = R_i\ (I_i,\ A_i,\ x_i) \qquad (2)$$

The state of the intersection is characterized by the state of the signal (green or red) and by the queue-length on each of the approaches. The input decision variable indicates whether the signal is to be switched at this stage or to remain in its present state. The return cost output is the intersection's index of performance (e.g., total delay time and /or number of stops), which has to be minimized. The functional relationships between the input and output variables are based on the queuing-discharge process occurring at the intersection, i.e., the vehicle inflow and outflow as a function of the signal settings.

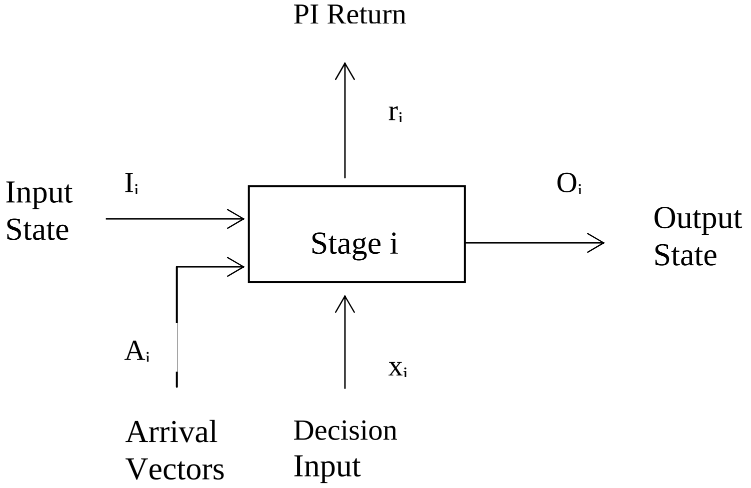

Figure 1. Dynamic Programming stage in OPAC-1

We define the following variables (all corresponding to stage i):

a = approach designation, by direction, a = N, S, E, W

A^a = number of arrivals during the stage (for a four-leg intersection)

D^a = number of departures (discharges) during the stage

Q^a = queue-length on approach at beginning of stage

S^a = status of signal at start of stage (green/red)

x = stage input decision variable (change/no-change)

The input state vector (transposed) is: $I^t = [\ I^N, I^S, I^E, I^W\ ]$, where for each approach the state is

$$I^a = [S^a, Q^a]^t$$

The output state vector contains the same elements as the input state vector and equals to the input state of the succeeding stage, i.e., $O_i = I_{i+1}$. The signal status for each approach a, is a binary variable:

$$S^a = \{0 \text{ for Green}; 1 \text{ for Red}\}$$

The input decision variable is also binary

$$x = \begin{cases} 0 & \text{no change in signal status} \\ 1 & \text{change current signal status} \end{cases}$$

The transformation of input to output, at stage i, is as follows:

$$S^a_{i+1} = (S^a_i + x_i)_{\mathrm{mod}\,2} \quad (3)$$

$$Q^a_{i+1} = Q^a_i + A^a_i - D^a_i \quad (4)$$

The *mod 2* operator ensures that S^a_{i+1} will always be 0 or 1. The arrivals at each stage i, A^a_i are an observed input (e.g., from detectors); the departures are a function of the state and decision variables:

$$D = \begin{cases} 0 & \text{if } S = 1 \\ \min(Q + A, d_{max}) & \text{if } S = 0 \end{cases} \quad (5)$$

where d_{max} is the saturation discharge rate (in veh/int). The DP algorithm goes backward in time, i.e., starting from the last interval and back-tracking to the first, at which time an optimal switching policy for the entire horizon is determined. The switching policy consists of the sequence of phase switch-ons and switch-offs throughout the horizon. The recursive optimization functional is:

$$f^*_i(I_i) = \min_{x_i} \{R_i(I_i, A_i, x_i) + f^*_{i+1}(I_i, A_i, x_i)\} \quad (6)$$

The Performance Index (return) at stage i is the queuing delay and number of stops N_s incurred at this stage:

$$r_i = R_i(I_i, A_i, x_i) = \sum_a (Q^a_i + A^a_i - D^a_i) + \sum_a N^a_s \quad (7)$$

When the optimization is terminated at stage i = 1 we have,

$$f^*_1(I_i) = \min_{x_i} \left\{ \sum_{i=1}^{N} R_i(I_i, A_i, x_i) \right\} = \sum_{i=1}^{N} R_i(I_i, A_i, x^*_i) \quad (8)$$

which is the minimized Performance Index over the horizon period for a given initial input state I_1. Since the initial conditions at stage 1 are specified (i.e., the queue-lengths on all approaches are given as well as the initial signal status), we can retrace the optimal policy by taking a forward pass through the arrays of $X^*_i(I_1)$. The policy consists of the optimal sequence of switching decisions { x_i^*, i = 1,…, N} at all stages of the optimization process.

While this procedure assures globally optimal controls for the given horizon length, it requires complete information of arrivals over the entire control period. It cannot be used for real-time implementation due to both the (excessive) amount of processing involved and due to the lack of a practical method to gather real-time information for such a length of time. Much of the output generated by the procedure is never implemented because optimized policies are calculated for all possible combinations of initial conditions at each stage of the control period. In practice, only one 'optimum policy' is implemented. Nonetheless, *OPAC-I* serves an important function as a standard for the evaluation of the relative effectiveness of other, more practical strategies.

2.2. OPAC-2: Sequential optimization

OPAC-2 breaks up the horizon into sequential optimization stages to speed up the optimization process. The model is a reformulation of the OPAC-1 algorithm. The purpose is to create a building block for a distributed on-line strategy. OPAC-2 has the following features:

- The control period is divided into successive (back-to-back) horizon lengths of T seconds each (T may encompass one or more cycle lengths).
- Each horizon is divided into an integral number of intervals 't' seconds long; typically, t = 2 - 5 sec.
- During each horizon there must be a sufficient number of phase changes to guarantee that no optimal solution is missed. The phase change (switching) times are measured from the start of the horizon in time units of t.
- For any given switching sequence in a horizon, the performance function for each approach computes the total delay and/or stops.

The optimization problem in OPAC-2 can be stated as follows*: For each horizon length, given the initial queues on each approach and the arrivals for each interval of the horizon, determine the sequence of switching times, in terms of intervals, which yield the least delay and/or stops to vehicles over the entire horizon.*

The procedure used for solving the problem consists of an intelligent search over the set of all possible combinations of feasible switching times within the horizon to determine the optimum sequence. Valid switching times are constrained by minimum and maximum phase durations. The problem can be re-formulated as an alternative dynamic programming problem by re-defining the control variable x_j to denote the amount of green plus yellow time allocated to stage s_i (Bertsekas, 1987; Sen and Head, 1997). The stages, in this case, correspond to the phase lengths during the horizon (see Figure 2). The recursive optimization functional (forward DP), is:

$$f_j(s_j) = \underset{x_j}{\mathrm{Min}}\left\{R_j(s_j, x_j) + f_{j-1}^*(s_{j-1})\right\} \tag{9}$$

The return (or Performance Index) is now

$$R = \sum_a \int_{s_{j-1}}^{s_j} A^a(t)dt + \sum_a N_s^a \tag{10}$$

This calculation results in the optimal sequence of signal phase (stage) lengths during the horizon. By reformulating the DP model, computation is considerably more economical than in *OPAC I*. *OPAC II* lends itself more readily to operation in real-time than does *OPAC I*; however, it still requires information on arrivals (flows) over the entire horizon length.

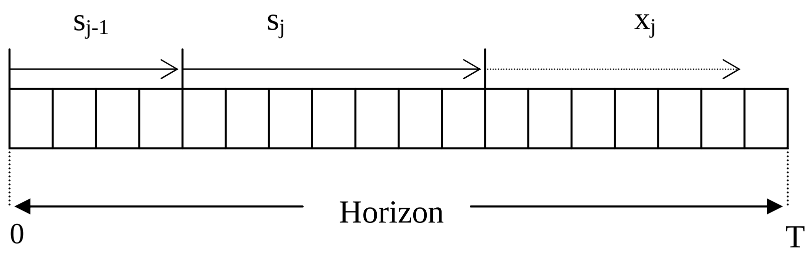

Figure 2: Dynamic Programming stage in OPAC-2.

2.3. OPAC-3: A rolling horizon approach

A typical horizon length is several minutes long. Obtaining accurate arrival predictions for this length of time is not feasible with current technology. To use only readily available flow data without degrading the performance of the optimization procedure, a 'rolling horizon' strategy is applied to the OPAC-2 algorithm. In this version, the horizon length, or the *Projection Horizon* is the period for which traffic patterns are projected and optimum phase change information is calculated. The key feature is that real-time data are required for only a small portion of the horizon.

Figure 3 is an illustration of the rolling horizon procedure. From detectors placed upstream of each approach actual arrival data for k intervals can be obtained for the beginning, or head, portion of the horizon. For the remaining n-k intervals, the tail of the horizon, flow data may be obtained from a model. A simple model consists of a moving average of all previous arrivals on the approach. An optimal switching policy is calculated for the entire horizon, but only those changes which occur within the head portion are actually being implemented. In this way, OPAC-3 can dynamically revise the switching decisions as more recent (i.e., more accurate) real-time data continuously become available.

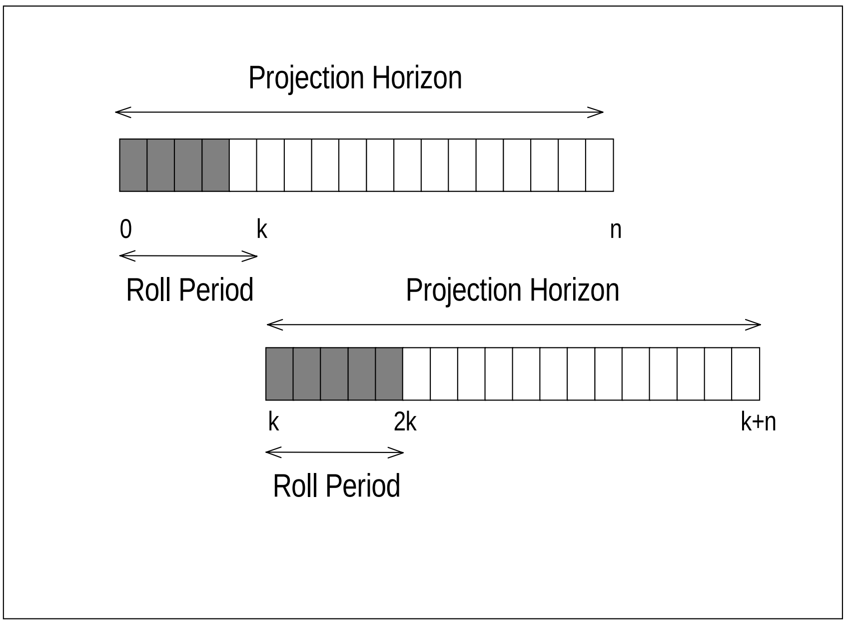

Figure 3: Implementation of the rolling horizon approach in OPAC.

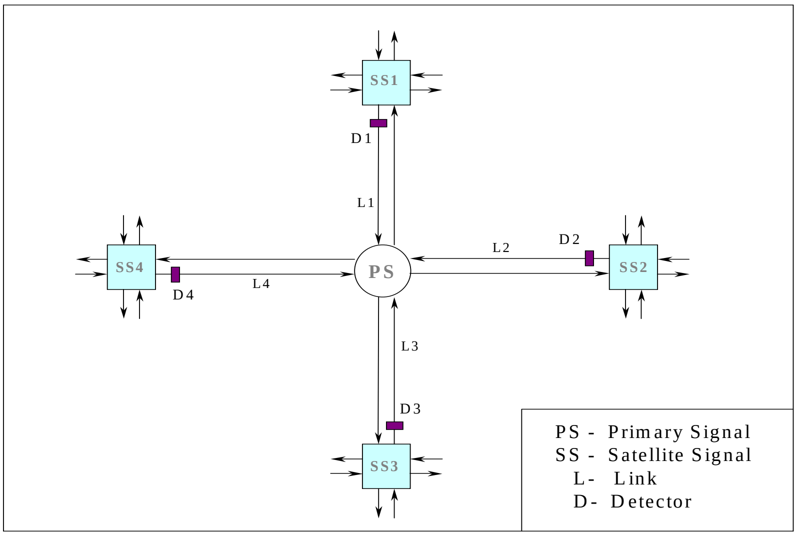

Figure 4: Information processing at an OPAC controlled intersection.

2.4. OPAC-4: A virtual fixed cycle approach

Whereas an independent OPAC controller is cycle-free, linking of adaptive signals in a network configuration requires coordinated operation to facilitate opportunities for unimpeded progression. Cyclic operation is essential for coordination of signals. This is achieved in the VFC-OPAC model by re-introducing the concepts of cycle time and offsets in an indirect manner that allows for increased flexibility of signal timing selection and control strategy implementation. VFC-OPAC controlled intersections interact with neighboring intersections (fixed-time, or other VFC-OPAC controlled) in response to projected arrival flows from upstream feeder signals. The model offers, at the option of the user, a coordination-synchronization strategy that is suitable for implementation in arterials and in networks. The strategy is referred to as *virtual-fixed-cycle* because from cycle to cycle the yield point, or local cycle reference point, is allowed to range about the fixed yield points dictated by the virtual cycle length and the virtual offset. This allows the synchronization phases to terminate early or extend later to better manage dynamic traffic conditions. VFC-OPAC consists of a three-layer control architecture as shown in Figure 5.

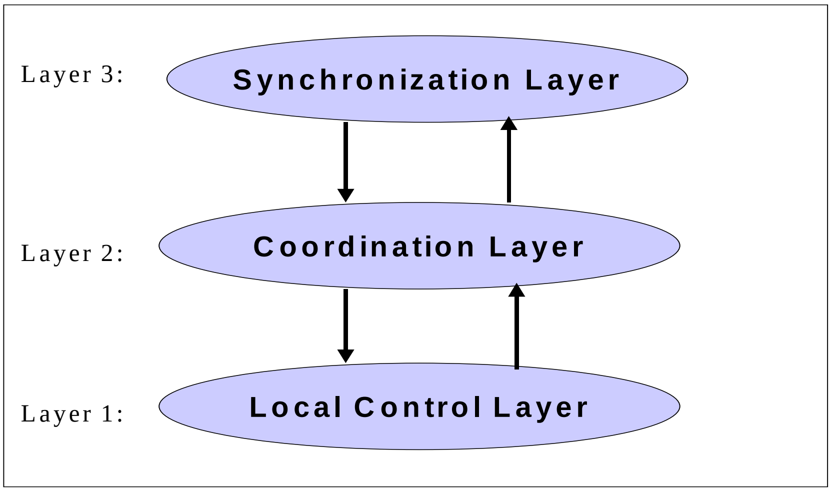

Figure 5: Control architecture in VFC-OPAC.

Layer 1: The *Local Control Layer* implements the OPAC-3 rolling horizon procedure using the dynamic programming model of OPAC-2. It continuously calculates optimal switching sequences for the Projection Horizon, subject to the VFC constraint communicated from Layer 3.

Layer 2: *The Coordination Layer* optimizes the offsets at each intersection (once per cycle). This is done by searching for the best offset of the PS (primary signal) within the mini-network shown in Figure 4. A choice of three offset increments is being considered: 0 (no change), +2-sec (move right one interval), -2-sec (move left one interval). The cyclic flow profile associated with each incoming link is being discharged by the model through the intersection and projected to the downstream intersections, SS (satellite signals). All other parameters are being kept at their latest values in a relaxation mode. Since the coordination process is carried out in a distributed fashion at each intersection, each SS, in its turn, is also considered a PS of its own mini-network once during each cycle.

Layer 3: *The Synchronization Layer* calculates the network-wide *virtual-fixed-cycle* (once every few minutes, as specified by the user) in order to maintain a rhythmic operation of the signals in the network. The objective is to provide maximum leeway of phase switching timings as dictated by local conditions, yet maintain a capability for coordination with neighboring intersections by maintaining synchronicity of the signals which are linked in the network. The *virtual-fixed-cycle* (VFC) is calculated in a way that provides sufficient capacity at the most heavily loaded intersection while, at the same time, maintaining suitable progression opportunities among adjacent intersections. The VFC is calculated as follows:

- Check if pre-set time period (3-5 min), or number of cycles counted, has elapsed
- Identify the dominant intersection (based on flow/saturation flow ratios)
- Establish bounds on VFC to satisfy the following requirements:

a. Provide sufficient capacity.

b. Keep the degree of saturation under a preset maximum k_m (e.g., k_m =0.90).

c. Do not exceed maximum phase length limitations.

The virtual-fixed-cycle is then chosen as the lowest value satisfying all these requirements. In addition, there may be exogenously determined limits on the cycle time:

$$C_{min} \leq VFC \leq C_{max}$$

The virtual-fixed-cycle model is illustrated in Figure 6. The cycle of a VFC-controlled inter-section can start/terminate only within a prescribed window Δ. The center of this window is the location of the start/end point of a (hypothetical) fixed-cycle signal if a fixed-time controller were used. The line denoting the center is called the “marker” line. Since the actual cycle start/end points do not occur, necessarily, at the marker line itself, we have a “virtual-fixed-cycle” situation. At any particular signal one would observe a variable cycle operation which, however, maintains a steady frequency or rhythm over the longer run.

The three-layer architecture is an effective means for implementing the distributed dynamic programming (DDB) algorithm (Bertsekas, 1995). The VFC can be calculated separately for groups of intersections, as desired. The position of the marker line also determines the “virtual offset” of the signal. Over time the flexible cycle length and offsets are updated as the system adapts to changing traffic conditions. This flexibility can provide improved local adaptiveness while, at the same time, maintaining good network co-ordination.

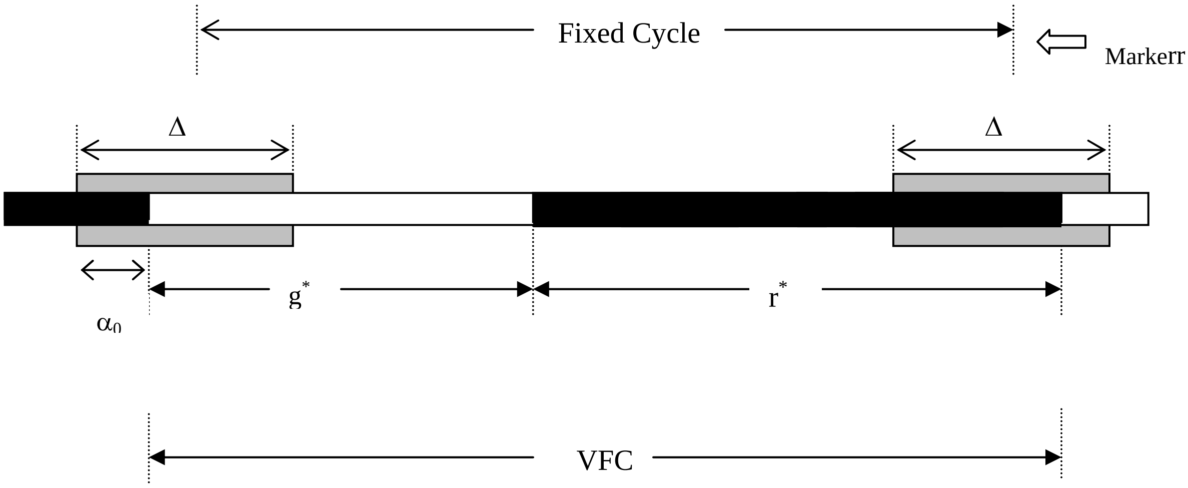

Figure 6: Essentials of virtual fixed cycle operation.

3. Implementation of OPAC in RT-TRACS

Based on the principles of operation described above, coordinated OPAC was designed as a truly adaptive control algorithm with numerous features, including:

- Full intersection simulation with a platoon identification and modeling algorithm - Data from detectors upstream of the intersection are used to develop expected arrival patterns for all phases. The signal timings and the arrival patterns are used to estimate delays and stops. Depending on the composition of the detector data, the patterns may be uniform, random, or platooned.
- Split optimization for up to 8 phases in a dual-ring configuration - The phases whose splits are to be optimized is configurable. Minor phases, for example, can be left to the default control while only major phases are optimized. Phases with no detectors can also be left out of the optimization since their calculation would be based on unreliable estimates of demand.
- Configurable performance function of total intersection delay and/or stops - The performance function is a weighted function of total intersection delay and stops. The weights are configurable to eliminate either delay or stops, or set their relative importance. Emphasizing delay causes *opac* to equalize delay among phases, which generally leads to shorter cycles. Emphasizing stops causes *opac* to equalize stops among phases, which tends to mean longer cycles.
- Optional cycle length and offset optimization - The central system optimizes the (virtual) cycle length for each section or group of intersections. A 'critical intersection' is determined periodically and the virtual cycle length is calculated based on flow data from the critical intersection. Offset optimization is performed in the field computer using peer-to-peer data from adjacent intersections. Offset adjustments may be made as often as once per cycle.
- Free and explicit coordinated modes - *opac* may operate 'free' where there are no cycle or offset constraints. Split optimization is constrained only by phase-specific minimum and maximum green. In the 'coordinated' mode, split optimization is also constrained by the dynamic *vfc* and offset values.
- Phase skipping in the absence of demand – *opac* may skip the user selected phases when there are no demands.

- Automatic response to changes in phase sequence - It is sometimes advantageous to have phase sequence change by time of day. For example, with lead/lag left turns, the leading phase can be changed between the morning and evening peak periods. *opac* automatically detects when these changes have been made and responds accordingly, although it does not itself determine or optimize phase sequence.

The OPAC strategy in its various versions has undergone a number of enhancements and field implementations starting in 1986 (Gartner et al, 1991; Ghaman and Curtis, 1998). It has also been tested and implemented in numerous simulation studies which demonstrated the superior suitability of the rolling horizon concept for adaptive control (see, for instance, Shelby, 2004). The earlier field tests evaluated the single intersection versions of the program. The coordinated version was tested for the first time in conjunction with the RT-TRACS project in a major U.S. suburban arterial corridor in 1998. RT-TRACS (Real-Time Traffic-Adaptive Control System) is a US-sponsored research project for the development and testing of advanced traffic control strategies (Gartner et al, 1995 and 2002). The system was installed in a selected corridor and evaluated against the best fixed-time plans that could be developed prior to the implementation. The scope of work for the field research test included: upgrading the site to meet advanced hardware and communications requirements; retiming signals to provide the best base case scenario; installing the adaptive software and performing calibration and fine tuning; and, collecting before/after study data to evaluate system performance. This process began in the spring of 1996 and the system became operational in the spring of 1998.

The test network is depicted in Figure 7. It consists of 16 signalized intersections along a 4-mile section of Reston Parkway in Northern Virginia. The corridor is a major travel and commuter route between Reston and the Washington Metropolitan Area through the Dulles Access/Toll Road. The test area, comprised of residential and commercial buildings, is a highly congested area during peak periods, as well as mid-day, evening hours and weekends. The Reston Town Center located in the middle of the corridor is a major shopping and entertainment center. Furthermore, the Washington Old Dominion (WOD) trail attracts hundreds of bikers and joggers every day of the week. This trail intersects the corridor at Bluemont Way, where a 32 sec all-red pedestrian crossing time is pedestrian actuated. During good weather conditions, the pedestrian traffic at this intersection would hamper the signal coordination and cause vehicle queues extending to the upstream intersection 450 ft from Bluemont. A detailed description can be found in Ghaman and Curtis, 1998.

Field Operation

The Reston system configuration includes a two-level distributed system as shown in Figure 8. As seen in this figure, the local level consists of Type 2070 controllers which host the OPAC real-time adaptive control strategy. The adaptive strategy resides on a separate CPU card (68040) within the VME chassis of the controller. The central system functionality includes operator interface, server, database, and communications between the central system and field controllers as well as communications between adjacent controllers. Upstream loop detectors, installed on all through approach lanes to the intersection, were used to provide real-time traffic data (count and occupancy) to OPAC.

Performance of the loop detectors was validated at various locations using system and stop bar detectors as well as traffic counters (tubes). Using actual and historical detector data provided by the operating system, the total vehicle counts registered by the loops were compared with those collected by stop bar and system detectors. Traffic tubes were also installed at various locations to check total traffic counts versus detector data.

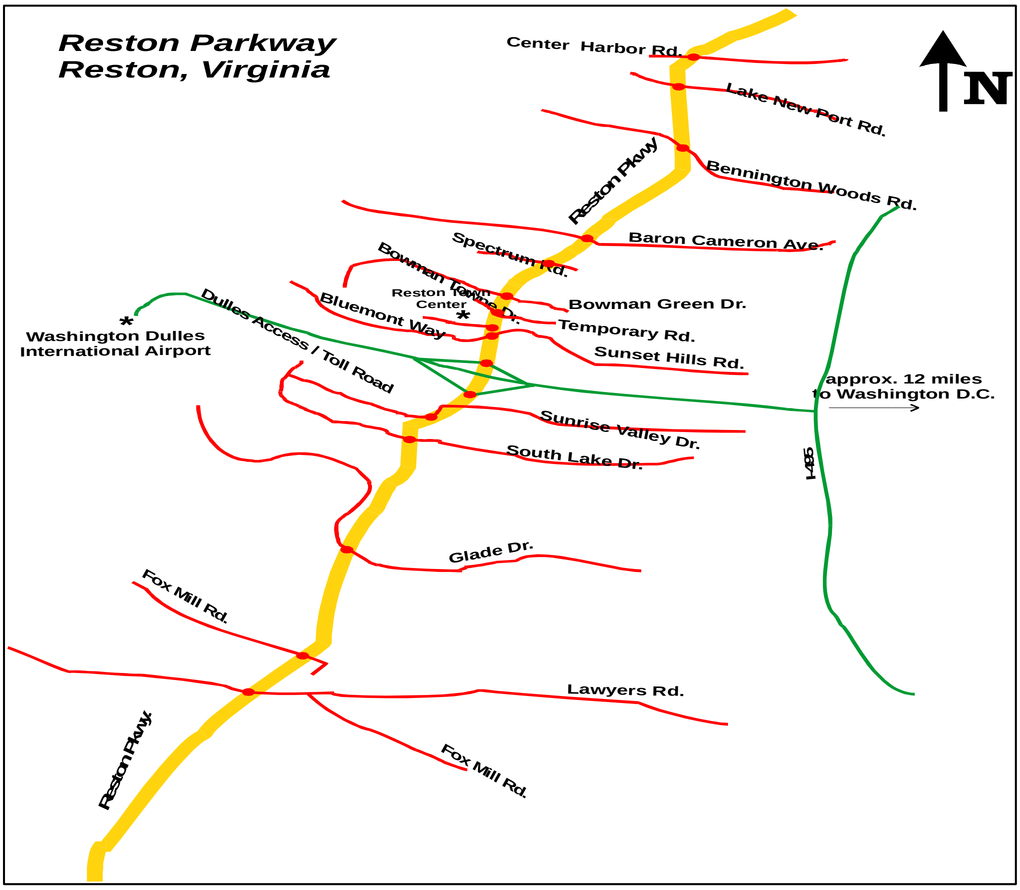

Figure 7: Reston Parkway RT-TRACS demonstration site.

Using actual detector data from loops and manual data collected at some intersections to get turning percentages, signal timing plans were optimized by the TRANSYT-7F program. These time-base coordination (TBC) plans provided the best possible fixed-time base condition in order to compare the performance of advanced RT-TRACS strategies.

Evaluation

The ‘before’ study data collections were done in November 97 once the signal timings were optimized and implemented. Due to delays in upgrading the hardware and communications system, the ‘after’ data collection was carried out four months later, in March 98. In addition to the typical hardware and software problems that are expected in implementing a new traffic signal control system, several other issues emerged. One major problem was caused by the local phone service provider in upgrading the system, which resulted in frequent loss of communications at several intersections. This had a serious impact on coordination of the signalized intersections. Another issue were the construction activities that began in early spring 98 concurrent with the system implementation and evaluation activities. The deadline imposed by this activity prevented adequate calibration and fine-tuning of the newly installed system before the ‘after’ study data collection was to be carried out.

The results given below were prepared based on real-time data collected by the central system using the OPAC-generated timing parameters, the signal status, the traffic volume and some data provided by the evaluator. The examples presented below are for the periods that the network was running under OPAC control without interruption due to communication system malfunction.

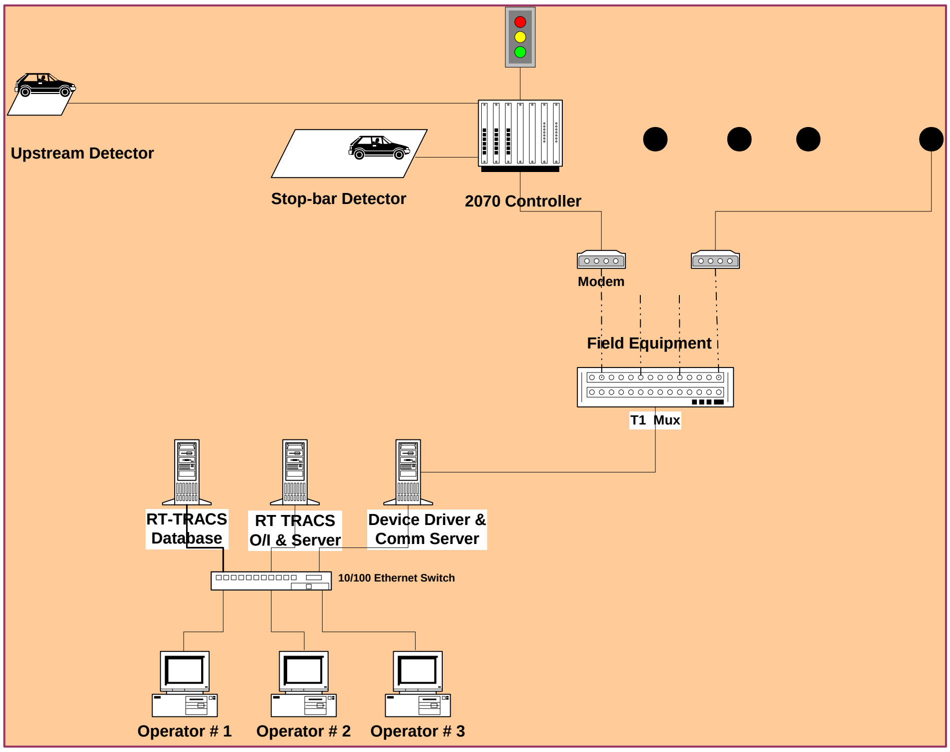

Figure 8: Reston RT-TRACS System Architecture.

Data Analysis

The objective of the field test was to evaluate the effectiveness of coordinated OPAC under different traffic conditions and network geometry. It was also intended to provide insight into the various functionalities of the program, including calibration of OPAC parameters to satisfy the needs of different segments of the network. The evaluation needed to consider separately different times of day, different days of the week and separate segments of the study area, including critical intersections and coordination with closely spaced non-OPAC controlled intersections. One of the interesting observations was the level of variation of traffic conditions between the before and after study periods. This included changes in vehicular traffic patterns as well as a significant increase in pedestrian presence in the middle of the corridor during the after study period. Pedestrian traffic was a significant factor during the after study but was minor during the before study. Figure 9 presents a typical example of changes in travel time under the before study condition (with TBC) within four months. One set of travel time data was collected in November 97 as part of the before study data collection and the second set in March 98 prior to the after study conditions. Both cases show the southbound direction with long delays at two major intersections; namely, Sunset Hill (on Link 8) and Baron Cameron (on Link 13).

Furthermore, collected data in March 98 show much higher overall delays compared to those in November 97. Similar observations were made for the northbound direction with less variation. Figures 10 and 11 present a comparison of OPAC cycle length and phase durations (WB Baron Cameron), respectively, versus existing pre-timed plans. As seen in these figures, while OPAC was closely following the best fixed-time parameters, it was also responsive to traffic demand by adjusting phase durations and cycle length with respect to phase demand and total intersection volume, respectively. In addition, it was also following the constraints imposed by the local agency (Virginia DOT) on minimum and maximum green times and cycle lengths. The TBC phase and cycle lengths, on the other hand, even though they reflected recently optimized values, could not meet the network needs to deal with dynamically varying traffic volumes.

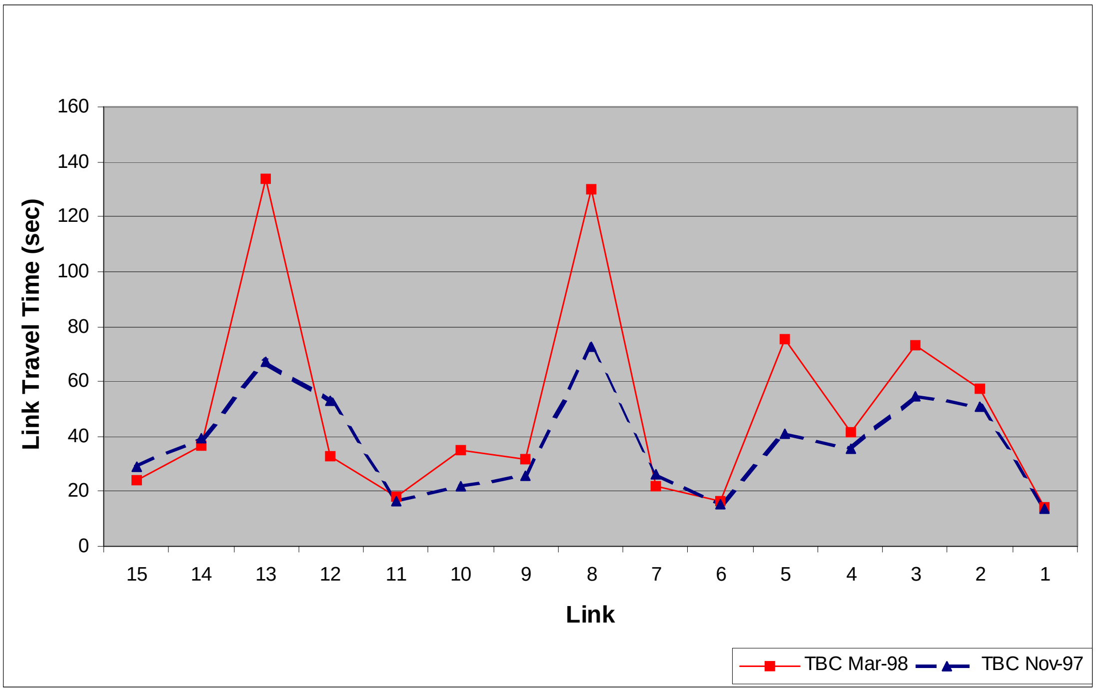

Figure 9: Variation of travel time due to seasonal change.

Figure 12 is an example of a travel time study during the AM peak period. This is the average travel time for weekdays in the northbound direction. The results show that OPAC performance was very close to that of TBC (overall changes for the corridor were within ±3%). Overall, OPAC showed improvements on the order of 5 to 6% in delays and stops on the Reston Parkway research test bed (NAWGITS, 2000). This is quite significant considering that the base condition against which it was compared was a fine-tuned TRANSYT-optimized timing plan. These results are comparable to those reported from other adaptive control system evaluations (Dey, 2001).

The first implementation of RT-TRACS provided a great deal of insight about the performance of the system. It also provided valuable experience with the installation of hardware and software as well as evaluation of the RT-TRACS strategy. The findings of this study were used for further enhancements and improved functionality of the OPAC algorithm.

Two key observations about OPAC were:

- Intersections operating under OPAC control can operate effectively even if adjacent signals are operating in time-of-day mode and even if there is no communication with adjacent signals. This is due to the fact that OPAC is a distributed system consisting of self-optimizing, self-coordinating capabilities.
- OPAC automatically responds to the loss of peer-to-peer data and communication with the central monitoring system to continue to operate as effectively as possible.

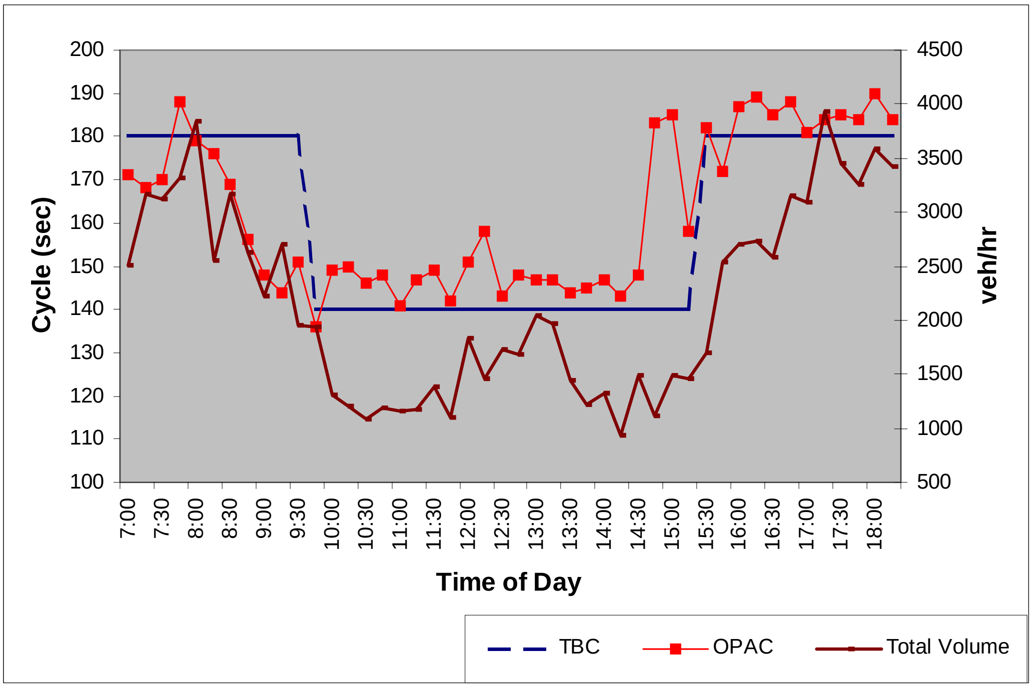

Figure 10: Cycle optimization.

The implementation provided insights into the performance of coordinated OPAC under various traffic conditions and site geometry. The hierarchical structure implementing the distributed dynamic programming algorithm worked effectively. The virtual-fixed-cycle concept was shown to be an effective tool in balancing the conflicting requirements of local intersection adaptability versus system-wide benefits of co-ordination and synchronization. Based on these findings additional enhancements were made to OPAC to improve functionality of the algorithm. The findings also suggest that this strategy could yield even better results when operational difficulties (such as communication system failures, or construction work within the test area) can be avoided.

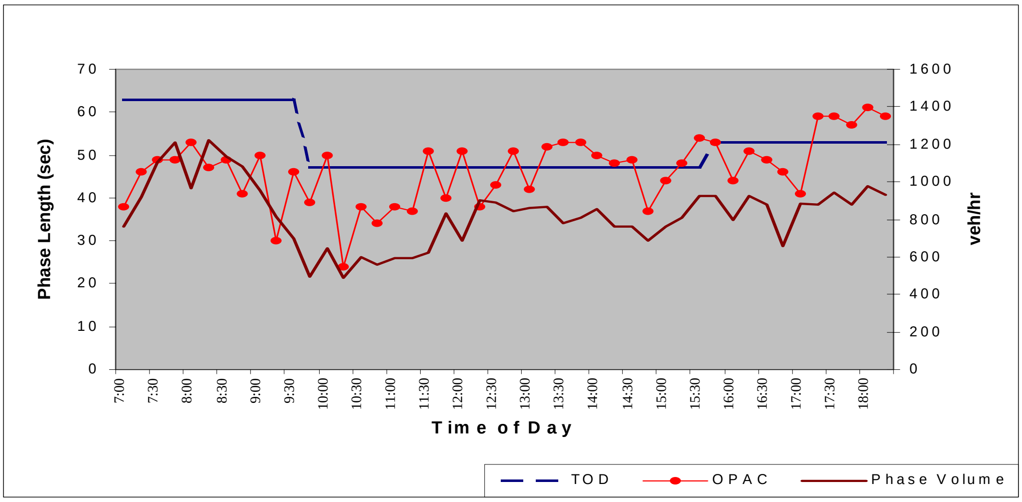

Figure 11: Phase optimization.

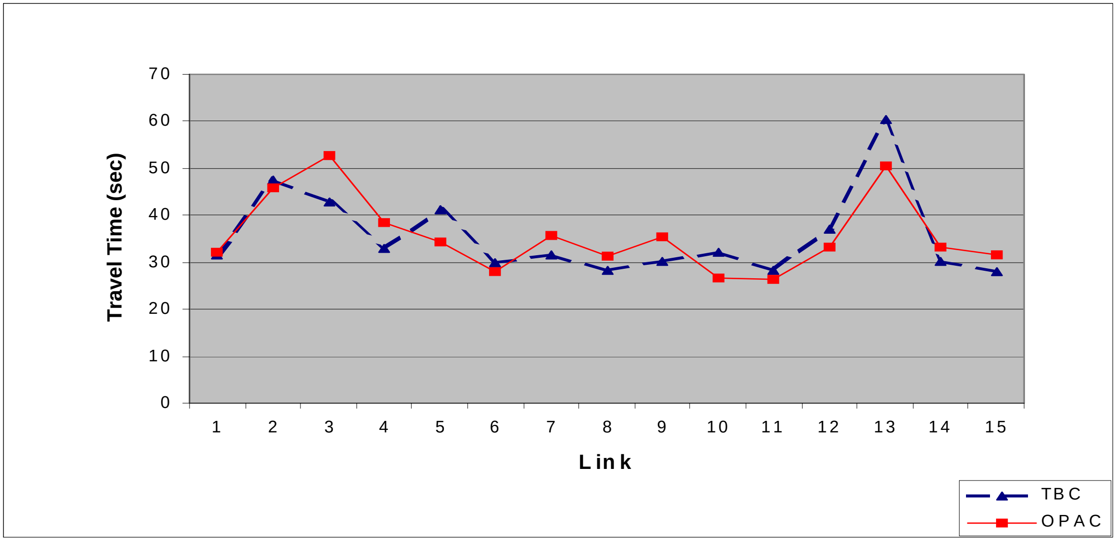

Figure 12: Travel time study.

4. Conclusions

The VFC version of OPAC was successfully implemented and evaluated under the RT-TRACS project in a 16-intersection corridor in Reston, Virginia. Results of the evaluation are reported in this paper. The evaluation provided considerable insights into the performance of the network version of OPAC under various traffic conditions and site geometry and offered the opportunity for additional enhancements to improve functionality of the algorithm. It showed improvements on the order of 5 to 6% in delays and stops in the field research test bed in Northern Virginia. This is significant considering the base condition against which OPAC was compared was a fine-tuned fixed-time plan. One has to consider the fact that adaptive control systems are able to continuously optimize signal timings, whereas off-line plans, such as those prepared by TRANSYT, are steadily ageing and performance is deteriorating at an average reported rate of 3-4% per year if not continuously updated. The findings suggest that the DP-based strategy can provide a true adaptive control algorithm for traffic signal networks.

References

Bellman, R., and Dreyfus, S., 1962. Applied Dynamic Programming. Princeton University Press, Princeton, NJ, USA.

Bertsekas, D.P., 1987. Dynamic Programming: Deterministic and Stochastic Models. Prentice-Hall.

Bertsekas, D.P., 1995. Dynamic Programming and Optimal Control, Volumes I+II. Athena Scientific, Belmont, Massachusetts, USA.

Dey, D., 2001. Workshop on Adaptive Signal Control Systems. TRB Annual Meeting, Washington DC, January 2001 (http://signalsystems.tamu.edu/archive.html).

FHWA, 1991. Real Time, Traffic Adaptive Control for IVHS. Solicitation No. DTFH61-92-R-00001, Federal Highway Administration, November 1991.

Gartner, N.H., 1982a. A Prescription for Demand-Responsive Urban Traffic Control. Transpor-tation Research Record 881. TRB, 73-76.

Gartner, N.H., 1982b. Development and Testing of a Demand-Responsive Strategy for Traffic Signal Control. Procs., 1982 American Control Conference, 578-583.

Gartner, N.H., 1983. OPAC: A Demand-Responsive Strategy for Traffic Signal Control. Transportation Research Record 906. TRB, 75-81.

Gartner, N.H., 1985. Demand-Responsive Traffic Signal Control Research. Transportation Research, 19A, 369-373.

Gartner, N.H., Tarnoff, P.J., and Andrews, C.M., 1991. Evaluation of the Optimized Policies for Adaptive Control (OPAC) Strategy. Transportation Res. Record 1324, TRB, 105-114.

Gartner, N.H., Stamatiadis, C., and Tarnoff, P.J., 1995. Development of Advanced Traffic Signal Control Strategies for ITS: A Multi-Level Design. Transportation Res. Record 1494, TRB, 98-105.

Gartner, N.H., Pooran, F.J., and Andrews, C.M., 2002. Implementation and Field Testing of the OPAC Adaptive Control Strategy in RT-TRACS. Annual Meeting of the Transportation Research Board, Washington DC, January 2002.

Ghaman, R.S., and Curtis, D., 1998. Real Time Traffic Adaptive Control Software: The First Implementation. Compendium of the 68th Annual ITE Meeting (CD-ROM), Toronto, Canada, August 1998. Institute of Transportation Engineers (www.ite.org).

NAWGITS, 2000. Progress and Prospects along the RT-TRACS: Exploring Adaptive Control Software. Newsletter of the ITS Cooperative Deployment Network. The National Associations Working Group for ITS (www.nawgits.com/icdn/rt-tracs.html), Sept. 2000.

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

Shelby, S., 2004. Single Intersection Evaluation of Real-Time Adaptive Traffic Signal Control Algorithms. TRB Annual Meeting, Washington DC, January 2004

(http://www.trb.org/am/ip/).