Pergamon

Transpn. Res.-B. Vol. 28B, No. 4, pp. 315-331, 1994
Copyright © 1994 Elsevier Science Ltd
Printed in the UK. All rights reserved
0191-2615/94 $6.00 + .00

0191-2615(94)E0008-M

A PROCEDURE FOR REAL-TIME SIGNAL CONTROL THAT CONSIDERS TRANSIT INTERFERENCE AND PRIORITY

SAM YAGAR and BIN HAN*
Department of Civil Engineering, University of Waterloo, Waterloo, Ontario, Canada

(*Received* 18 *May* 1992; *in revised form* 21 *February* 1993)

Abstract — A rule-based procedure for determining real-time signal timings at a signalized intersection is described. It incorporates the effects of the traffic interference caused by on-line loading/unloading of transit vehicles at the intersection. This procedure generates a number of short-term alternative real-time phase sequences for various levels of transit priority, based on a number of decision rules. It then evaluates these signal sequences and selects the one with the least overall cost to all traffic. The procedure is illustrated in terms of a simulated application to a critical intersection in Toronto's Queen Street corridor using real data. The preliminary simulation tests indicate the potential reduction in total delay compared to fixed-time operation, which results largely from selectively ushering transit vehicles to their loading positions at strategic times and serving cross-street traffic while the transit vehicles are loading.

INTRODUCTION

With the rapid development of microprocessors, efforts have been devoted to developing traffic-responsive control methods to meet the challenge of ever-changing traffic demand and more complex situations. Because conventional fixed-time signal control methods are based on the use of historic data, they cannot fully accommodate time-dependent traffic flows especially in cases with large transient demand perturbations. When demands vary and can be monitored in real time, demand-responsive control strategies have the potential to perform better than fixed-time control strategies by employing the information from vehicle detectors. Depending on the level of information and processing power, the appropriate type of real-time control decisions may involve simple rules or an objective optimization process. When the process is further complicated by real-time variations in capacity, as are caused when transit vehicles load on-line, optimization becomes even more tricky.

We describe a process that can generate and select signal plans in real-time using simple generic lists of rules that allocate different priority levels for traffic events. It also allows the user to add other lists and can further be extended to consider more complex strategies. In particular, the technique is sensitive to disruptions and reduced capacity caused by transit vehicles loading near the intersection.

We begin by providing some context for the work and the proposed technique by reviewing some of the representative literature on real-time control, which generally involves optimization techniques and ignores transit effects. We then briefly describe the additional requirements for handling mixed transit operation on the common right-of-way. We then describe a procedure for:

1. Modelling vehicle detection at one or more locations upstream of the intersection on each approach to the intersection;
2. Generating short-term signal phase sequences corresponding to each of several lists of relative priority for the significant events that might occur at each approach to the intersection;

*Present address: Dunn Engineering Associates, 66 Main Street, Westhampton Beach, NY 11978.

3. Evaluating each of these phase sequences in terms of the total cost to drivers and transit patrons based on the respective weights given to transit and private vehicles;
4. Selecting the apparent best signal timings and implementing them for a short roll-forward period.

EARLIER DEVELOPMENTS IN REAL-TIME CONTROL

The development of real-time control methods has followed that of computer hardware, exploiting the increasing speed of computation to evaluate alternative signal timings on line. Earlier methods had to use efficient optimization procedures, and therefore tended to ignore the effects of factors such as on-line transit loading, which were difficult to include in optimization models. Some of these are described to provide a feel for the types of optimization that were considered appropriate when transit is not a factor and to provide some context for the model described herein.

Miller (1963) performed some of the pioneering work on traffic-responsive signal settings. A decision was made every few seconds by comparing the benefit and disbenefit of extending the current green. This was done by examining the gain made by the extra vehicles that could pass through the intersection during the extension and the loss to those vehicles delayed on the red approach because of the extension. Imminent vehicle arrivals were predicted by detectors and arrival rates for the rest of the optimization period were based on exponentially smoothed flows.

Bang (1976) made some modifications to Miller's algorithm. Simulation and field test results showed that this method could give substantial reductions in both average delay and number of stops as compared with conventional fixed-time and vehicle-actuated control.

De Groot (1981) proposed two methods of traffic responsive control: *Delay Profile Method* and *Dynamic Estimate Method.* The Delay Profile method is essentially quite similar to Miller's and Bang's approaches. It differs from the previous methods in that the delay estimation is made by considering each individual vehicle's delay rather than using the approximated average traffic volume to calculate delay on each link. The optimal control policy is one that minimizes the total anticipated delay to vehicles on all approaches as a function of the extension interval. The Dynamic Estimate Method, on the other hand, uses the dynamic programming technique to find the optimal signal settings. The control period is divided into a series of control stages, during which some signal states are defined. Each state is defined by a three dimensional vector containing the queue at each approach (n_1, n_2) and the approach that has the green signal (1 or 2).

Gartner (1983) developed a model called OPAC (Optimization Policies for Adaptive Control). The optimization process is divided into sequential optimization stages of T seconds. During each optimization stage, up to three signal changes are allowed. For any given switching sequence at optimization stage n, a performance index is defined:

$$\phi_n(t_1, t_2, t_3) = \sum_{i=1}^{k} (Q_0 + A_i - D_i(t_1, t_2, t_3))$$

where
Q_0 = initial queue
A_i = arrivals during interval i
D_i = departures during interval i and
(t_1, t_2, t_3) = possible switching times for this optimization stage.

The total delay is evaluated sequentially for all feasible switching sequences. At each iteration, a direct search is made and the switching sequence that gives the minimum delay is taken as the solution for this optimization stage. This process is carried out independently for all the optimization stages. However, because the algorithm requires future arrival information for the entire optimization stage, a rolling horizon method is further introduced. The length of the optimization stage is divided into two parts: the 'head' (length r) and the 'tail' ($k - r$, where k is the stage length). The flow data for the head are obtained from the detectors, and the flow data for the tail are estimated from a model. An optimal policy is then calculated for the entire optimization stage but implemented only for the head section. The projection period is then shifted (rolled) up to r units ahead, new flow data are obtained for the optimization stage, and the process is repeated.

Lin et al. (1988) developed an adaptive control logic called SAST (Stepwise Adjustment of Signal Timing). It divided time into discrete intervals or steps. In each step, a decision is made whether to extend the current green beyond that step. This is based on a four-level decision process. The first three levels are simple decision rules, and the last level involves signal optimization, which requires the evaluation of alternative signal switching sequences to reach a decision. The evaluation criterion is either the total delay to all the vehicles or the total delay to the vehicles in the critical lanes. This approach differs from Gartner's in that there is no prediction for the demands into the future.

Vincent et al. (1986, 1988) introduced the Microprocessor Optimized Vehicle Actuation (MOVA) strategy, which is based on the minimization of total delay to the traffic at an intersection. MOVA's basic approach is similar to Miller in that it compares the benefits and disbenefits of extending the green phase. The MOVA algorithm uses some models of lane behavior to calculate which vehicles will benefit if the current green is extended. It also maintains a count of vehicles currently queuing at red signals around the intersection. From these counts and the measured average arrival flows expected to join these queues in the near future, MOVA estimates the benefits and disbenefits if a green extension is made.

Bell et al. (1990) proposed a model similar to that of Gartner's in that the arrivals are estimated in two parts: the first part comes from the detectors and the second from a predictor. The additional features of this model include the introduction of a filter and a terminal cost function. Heydecker (1990) proposed a continuous optimization formulation using this model to take advantage of the modern group-based signal controllers. The solution method is based on a bilevel approach. At the upper level, the signal stage sequence is specified; at the lower level, the optimization is performed for each specified sequence.

MODELLING AND PRIORITY FOR TRANSIT VEHICLES

The major limitation for the models introduced above is that there is no special provision for transit vehicles, which are effectively treated as normal passenger cars. However, in mixed traffic operation, transit vehicles may hold up other traffic while loading passengers even if the signal is green. When this happens, especially near a signalized intersection, this traffic–transit interaction should be described properly as an input into the signal plan generation procedure. One should also give appropriate priority to transit vehicles, so that the total cost and delay to cars and transit passengers can be minimized. Several research efforts to solve this problem have been reported. These can be traced back to the early 1970's, e.g., Richbell and Van Averbeke (1972), Wattleworth (1977), Cornwell (1986), and Cansult Engineering Limited (1991). In most cases, transit priority is realized by employing simple decision rules.

Basically, transit priority can be achieved by two methods: *passive priority* and *active priority*. The former relies on the historical behavior of the traffic on the different approaches, whereas the latter is based on the selective detection of the transit vehicles. The most frequently used active priority measures are green extension and red truncation.

Because of the interaction between the transit vehicles and private cars, blind priority to transit vehicles based on simple rules without due consideration for private cars can be haphazard. It can result in excessive delays to and cause queues, making it more difficult to accommodate the transit vehicles that arrive later. A procedure is described herein, which develops a traffic responsive signal control method that can provide priority for transit vehicles such as buses and streetcars, but at the same time maintain the overall control performance for the intersection.

A PROCEDURE FOR REPRESENTING MIXED TRAFFIC/TRANSIT OPERATION

Modelling the traffic–transit interaction is described herein in terms of streetcar operation which causes larger and more transient interference to traffic than does bus operation. It was, in fact, the streetcar operation in Toronto which created the need for a real-time procedure, as current fixed-time methods have special difficulties incorporating streetcars.

An event-based approach is used to model the traffic behavior. It relies on the upstream detection of vehicles, which it then projects downstream to estimate time of arrival of the intersection. A set of candidate control strategies is generated to serve the demands. These correspond to a set of lists of priority rules for switching the traffic signal. Each list contains a set of candidate strategic events for switching signal phases. These lists might be developed by experts in traffic control. Key events in the list might include, for example, the arrival of a streetcar or dissipation of a queue. The ordering of events in a list reflects relative priority. This would be provided by a traffic systems analyst. The greater the capability of the processing computer, the greater the number and size of candidate lists that can be considered. This limitation requires a traffic analyst to carefully select and order the important events, minimizing effective redundancy.

The procedure described herein has so far concentrated on two-phase signal operation, which is the usual case in North America for busy downtown intersections. However, the number of approaches having a green in each signal phase is not limited. Each approach can have private vehicles, buses, and streetcars and there is no limit on the number of buses or streetcars on each approach.

System structure

The proposed control system structure is illustrated in Fig. 1. In this system, several vehicle detectors are placed in each traffic approach to gather real-time traffic information which is transmitted to a microcomputer. The computer then uses this information to generate real-time signal plans.

If a vehicle detector is placed at about 150 m upstream of the stopline, it would provide about 10–15 sec of advance information on vehicle arrivals, assuming speeds in the 40–50 km/h range. This detector will be referred to as a nearside detector. If the intersections are far enough apart that another detector can be placed considerably upstream of the nearside detector, then more lead time is available for the prediction of vehicle arrival times. For example, if intersections are 1000 m apart, about 60–90 sec of advance information will be available. In this case, because the traffic information comes from two parts, each of these could be used for generating control strategies. Because the information from the nearside detector is more precise it can be given a higher weight than that from farther upstream. Information from upstream detectors can be used for relatively "longer-range planning" (about 60–90 sec) of the signal sequences. The number of vehicle detectors recommended for an approach to an intersection depends on the distance to the upstream intersection and the average vehicle speed as well as other factors such as the cost of detectors. At the test intersection described later, 3 detectors are used for each approach.

Processing vehicle detection information

When a vehicle is detected by a detector, two messages are recorded: (a) the vehicle type, i.e., whether it is a private vehicle or a transit vehicle; (b) the detection time, i.e., the time at which that vehicle is detected. The vehicle type is used to identify transit vehicles so that the control algorithm can allocate appropriate priorities for them. The time of detection, and to some extent, the vehicle type, are used for projecting the expected time of arrival at the stopline. The expected time of arrival at the stopline is simply the distance between the detector and the stopline divided by the average speed for that type of vehicle.

The real-time control algorithm maintains a list of recently-detected vehicles for each detector, i.e., vehicles that have been detected by the detector but are not yet expected to have arrived at the next downstream detector or the stopline.

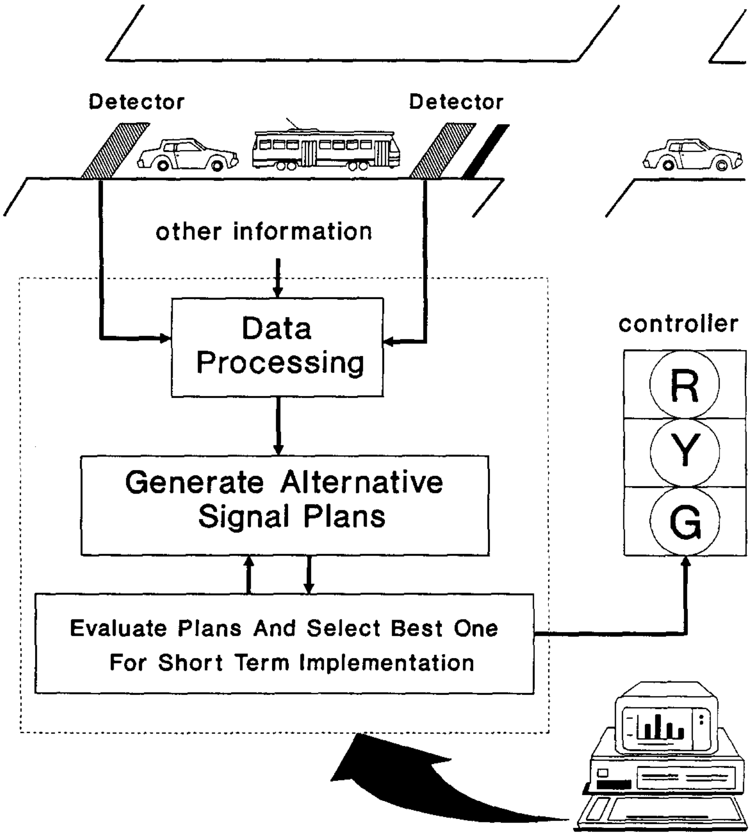
Detector
Detector
other information
Data Processing
Generate Alternative Signal Plans
Evaluate Plans And Select Best One For Short Term Implementation
controller
R
Y
G

Fig. 1. The control structure.

Although the control algorithm keeps adding each newly detected vehicle to the list of vehicles for each detector, it also keeps removing those vehicles that should have arrived at the next downstream detector, if one exists, or otherwise at the stopline, based on the average vehicle speed and the distance to the downstream detector or stopline.

Ideally, each vehicle should only be included in one detector's list of vehicles. However, the simple treatment introduced above can sometimes present problems, due to the imprecision in estimating the time for a vehicle to travel from a detector to the downstream point of interest. A vehicle may be double counted for a short time period if it travels faster than the assumed average speed as it will be detected by the downstream detector earlier than predicted. This would cause it to appear on both the current detector's vehicle list and the downstream detector's vehicle list, until its "expected" time of arrival, at which time it would be removed from the current detector's vehicle list. Similarly, a vehicle may be missing from the system for a short time if it travels at a speed slower than the assumed average speed. In either case, the control algorithm would generate signal timing plans based on somewhat inaccurate traffic information.

These problems are even more serious when there are transit vehicles such as streetcars in the system, because either double counting or missing a streetcar will generally result in poor signal timings. However, because transit vehicles, and especially streetcars, will not turn onto or off the roadway as private vehicles do, it is therefore assumed that:

1. A streetcar detected by a detector should reach its downstream detector within a reasonable time period, which is approximately the estimated travel time plus any loading/unloading time between the two detectors;
2. Within the same approach, two or more streetcars detected sequentially by any detector, should arrive at the downstream detector or intersection in that same sequential order in which they were detected.

Therefore, each streetcar will only be removed from a detector's list of vehicles when it has actually been detected by the downstream detector, rather than at the expected arrival time at the downstream detector. To avoid double counting streetcars, the control algorithm removes a streetcar from a detector's list of vehicles when a streetcar is detected by its downstream detector, even if that removed streetcar is not expected to have reached the downstream detector. On the other hand, to avoid missing streetcars, a streetcar will be kept if by the time of the expected detection by the downstream detector, no streetcar has not yet been actually detected. Meanwhile, its estimated detection time at the downstream detector is updated until the streetcar is detected.

Thus, the control algorithm is able to maintain lists of vehicles for each of the detectors in the system from which it can obtain estimates of arrival times for traffic in real-time. The time period for which future traffic information is available is called the *projection period.*

Modelling vehicle movement

The control algorithm maintains lists of vehicles for the vehicle detectors. An event-based approach is used to model vehicle movement. An event is defined as an occurrence that will change the state of the system, e.g., the arrival or departure of a vehicle at the stopline. By manipulating the expected arrival and departure times of each individual vehicle, the system behavior can be traced, allowing the control strategy to respond to it. The expected arrival time for each vehicle at the stopline can be deduced from the list of vehicles for each detector and its departure time can be estimated using the signal timing information and the service headway for the approach in which the vehicle is travelling.

In the following discussion, we assume a nearside streetcar stop at the stopline.

When the signal is green and there are no other streetcars loading in front of it, an arriving streetcar will move right up to the stop position. However, if there are other streetcars loading in front of it or if the signal is red the driver will normally open the door and start serving the passengers if the streetcar is reasonably proximate to the stop position. This proximity limit is governed by a user-specified parameter. For this initial study, we assumed that a streetcar could load if the streetcar arrived on red to join a queue of not more than three vehicles on the approach. If the signal is green and there is no other streetcar loading passengers in front of it, then this streetcar will only load when it has arrived at the stop position. However, if there is another streetcar loading/unloading in front of it, then this streetcar will behave as if the signal is red.

In the absence of any vehicle-specific information, it is assumed that each streetcar holds up traffic for a fixed amount of time. We have used the average loading time plus acceleration/deceleration time.

GENERATION AND SELECTION OF SIGNAL TIMING SEQUENCES

A sequence of signal timings is obtained for the projection period for which the traffic flow data are available, by developing the mutually dependent vehicle movement and signal switching scenario for each priority list, and selecting the one with minimum cost of stops and delays. This process renews at about 5-sec intervals as is discussed later.

Estimation of total delay

The procedure used for estimating the total delay for a given control strategy is based on the event-based modelling method. For each approach j ($j = 1, 2, \ldots, n$), the projection period T_p can be divided into a series of time intervals of varying lengths T_{j1}, $T_{j2}, \ldots, T_{jk}, \ldots, T_{jI}$, $k = 1, 2, 3, \ldots, I$, where I is the total number of intervals, t_{jk} is the time at which an event occurs (Fig. 2).

The objective function is defined as the total delay during the simulation period, which can be expressed by:

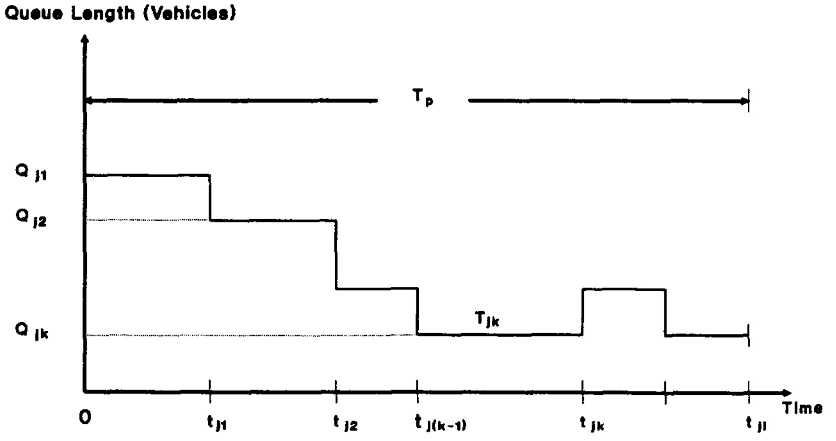

Fig. 2. Time evolution of queue lengths.

$$D = \sum_{j=1}^{n} \sum_{k=1}^{I} \left\{ T_{jk} \sum_{i=1}^{Q_{jk}} W_{ijk} \right\}$$

where

D = total delay for the projection period T_p;

n = total number of approaches;

Q_{jk} = queue length for approach j between $t_{j(k-1)}$ and t_{jk}, $j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, I$.

W_{ijk} = the weighing factor for vehicle i in the queue on approach j during interval T_{jk}, $i = 1, 2, \ldots, Q_{jk}$; j = 1, 2, . . . , n; $k = 1, 2, \ldots, I$. A car typically has a weight of 1, while a streetcar has a higher weight to give it priority in accordance with its occupancy.

Considerations in developing candidate signal plans

A real-time traffic responsive signal optimization routine must have a high execution speed to develop its signal plans on-line in the allotted time. Usually the generation of an optimum control strategy should not take more than 5 sec. This implies that it is generally not practical to search for global optima which can consume considerable computing time. However, it is feasible to consider several candidate event sequences. Therefore, a list of distinct events, that an experienced traffic cop or other expert might consider in switching the signal, is offered for consideration and pre-evaluation. Some important events are: (a) streetcar arrivals/departures; (b) queue requests/serving queues; (c) waiting for platoons.

A queue forms a request if the number of vehicles in that queue is greater than a specified value. While the first two events are relatively straightforward, the latter, "Waiting for platoons," attempts to somehow quantify an expert traffic cop's ability to synthesize upstream traffic patterns and decide that it is worthwhile to keep the signal green even though there is a gap in the traffic, in anticipation of the imminent arrival of a platoon behind this gap. This is described in the following section.

Waiting for a platoon. When a queue dissipates, it might be wise to switch the green phase to the competing approaches. However, due to the lost time involved in the inter-green phase, it might be wise to check the density of traffic just upstream of the tail of the queue, as a traffic cop might do. If there is a platoon approaching, its arrival time is taken as an event. The most critical platoon event is defined as the greatest average traffic density immediately upstream of the tail of the queue.

We define a user specified parameter τ, which might be of the order of 1 sec. This parameter represents a desirable average excess time per vehicle, which is used to identify the end of an extended platoon. At the time, t_o, when an initial queue is cleared, if the n^{th} upstream vehicle satisfies the condition:

$$t_n \leq n(h + \tau) + t_o$$

where h is the saturation flow service headway at the intersection and n is the index of this vehicle, and t_n is the expected arrival time of vehicle n at the stopline, then this vehicle is considered to be in the extended platoon. This consideration allows the routine to cater for platoons arriving during a green phase as well as for vehicles already waiting in queue. It allows upstream vision beyond the end of the first queue and an alternative to simply switching the signal as soon as a queue dissipates. For example, suppose that $h = 2$ sec/veh, and τ is specified as 1 sec. Then we would include at least an additional n vehicles upstream of the tail of the queue, where n is the smallest numbered vehicle within $(2 + 1)n = 3n$ sec of the intersection. This would extend the time of dissipation of the queue by $3n$ sec. Thus, if there were vehicles 4, 6, 7, 12, 18, and 19 sec upstream, we would wait for the first 2, because $4 > 3(1)$ but $6 = 3(2)$. Then we would renew the process and decide whether to wait for additional vehicles, whose new relative arrival times are converted from 7, 12, 18, 19 to 1, 6, 12, 13. We now add one more vehicle to the queue and the remaining relative arrival times become 5, 11, 12, so that no further upstream vehicles are awaited. To investigate even more possibilities, we could make τ a function of n, or platoons of different size could be given different priority levels. For example, a platoon that has a size of at least 5 vehicles can be allocated one level of priority, whereas a platoon whose size is between 1 and 5 can be given a lower priority. In any event, extending the green for upstream platoons will have an additional benefit of reducing vehicle stops. This may also enhance safety.

Defining ordered lists of event priorities

The following sample priority list is one of many possible competing ordered lists of events that might be used to determine which approaches should have a green phase for the example of Fig. 3. In this case, a streetcar in the peak (southbound) direction would always receive a green phase, subject to constraints such as minimum and maximum phase lengths. The next priority would be to continue a green phase if a southbound queue was being served (i.e., Event 2). If neither of Events 1 or 2 was occurring, then a streetcar on the cross street would get the green (i.e., Event 3). In the absence of events 1, 2, or 3, Event 4 would receive the green, and so on through Event 9 on the list. If none of the events is in effect, the signal remains in the current phase until the maximum green time constraint is reached. The sample priority list is as follows:

1. A streetcar on main street—peak direction
2. Serving a queue on main street—peak direction
3. A streetcar on the cross street
4. Serving a queue on the cross street
5. A streetcar in the main street off-peak direction
6. A queue request from main street—peak direction
7. Serving a queue in main street off-peak direction
8. A queue request from the cross street
9. A queue request from main street in off-peak direction

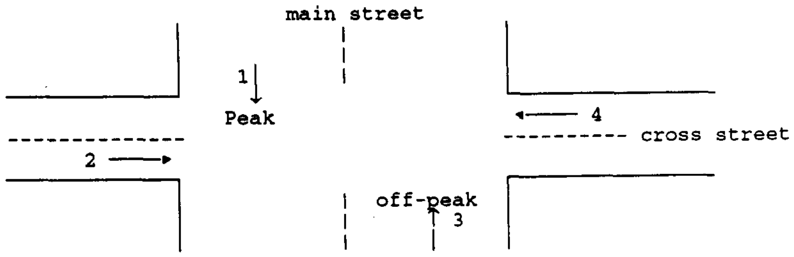

Fig. 3. An example intersection.

If this sample list were the only priority list, the signal would operate simply as traffic actuated according to this list, just as police with a pre-determined set of values would. However, other police, or expert, might prefer to give priority to transit in all directions, while yet another might treat transit vehicles like any other vehicle regardless of vehicle occupancy. Further to this other traffic experts or theorists might argue for strategies which do not "waste" valuable capacity by catering to the off-peak direction, such as northbound in Fig. 3, feeling that it will be taken care of as a by-product along with the peak (southbound) direction in the long run.

Evaluation and selection

Depending on the dynamic nature of the traffic, each of these experts will be right some of the time. The best strategy then, would be to look ahead at the traffic on all approaches when possible, pre-evaluate the effects of each expert's candidate control scheme and select the best at that time.

Signal plan generation

The signal plan corresponding to each priority list is generated for a projection period of typically 30-90 sec, corresponding to the advance traffic information from vehicle detectors, in the following way:

Step 1: Identify all requests for switching phases.

(a) Identify the starting and ending points of the projection period for which a signal plan should be generated based on predicted traffic information.

(b) Within the projection period, Identify the current *feasible switching region* at which the next switching point can be located. For example, for a projection period of 90 sec, if:
– current time is 0 and the signal has been green for 10 sec,
– minimum green time = 20 and maximum green time = 60,
then the current feasible switching region is between 10 sec and 50 sec.

(c) Estimation of queue evolution during the feasible switching region for all approaches. This is achieved by simulating the vehicular movement according to the traffic flow data and the assumed signal display within this time period.

(d) Identify all events within the feasible switching region that may lead to a phase switching. Then calculate the corresponding requests associated with each event.

Note that while an event from a priority list leads to a request for a green phase, the time of occurrence of the request is considerably more complex than the simple detection of an event. The request will usually occur earlier than the corresponding event even though it is caused by the event. For example, if there is a queue in front of a streetcar, that queue must be cleared first before the streetcar can be served. Therefore, the time of the request from the streetcar is converted from the raw expected arrival time for the streetcar, to an earlier time which reflects serving the queue in front of that streetcar, so that the streetcar has a clear path to the intersection or its loading point, i.e., it is a calculated time which reflects the latest time at which the signal could turn green and still serve the obstructing queue, so that the streetcar has a clear path to its loading point at the intersection.

Step 2: Find the switching point.

In the following description, the term *red approaches* refers to those approaches that receive a red signal display and the term *green approaches* stands for the approaches that receive a green signal display.

The procedure processes requests for green from the red approaches sequentially. A switchover from green to red only occurs when the following two conditions are satisfied:

(a) the request has a higher priority than all the existing requests on the green approaches;
(b) the request can be met without violating any subsequent request in the green approaches which has a higher priority; e.g., there is enough time to come back to serve a streetcar on the green approaches. Otherwise, this request will be stored and might be met later when the above conditions are satisfied.

The flow chart for this part of the algorithm is as shown in Fig. 4.

Evaluating the candidate switching plans

After each short-term signal plan has been generated for a given priority list, its corresponding Performance Index (PI) or delay is then evaluated for a time period equal to the duration of the "plan" horizon or projection period. The list with the smallest PI is then selected for implementation as described next.

The dynamic implementation process

Although a signal plan is selected based on flow information for a period of about 30–90 sec, say, only part of this plan is implemented (e.g., ~5 sec). This process is then renewed dynamically based on the latest flow information, as illustrated in Fig. 5. In a

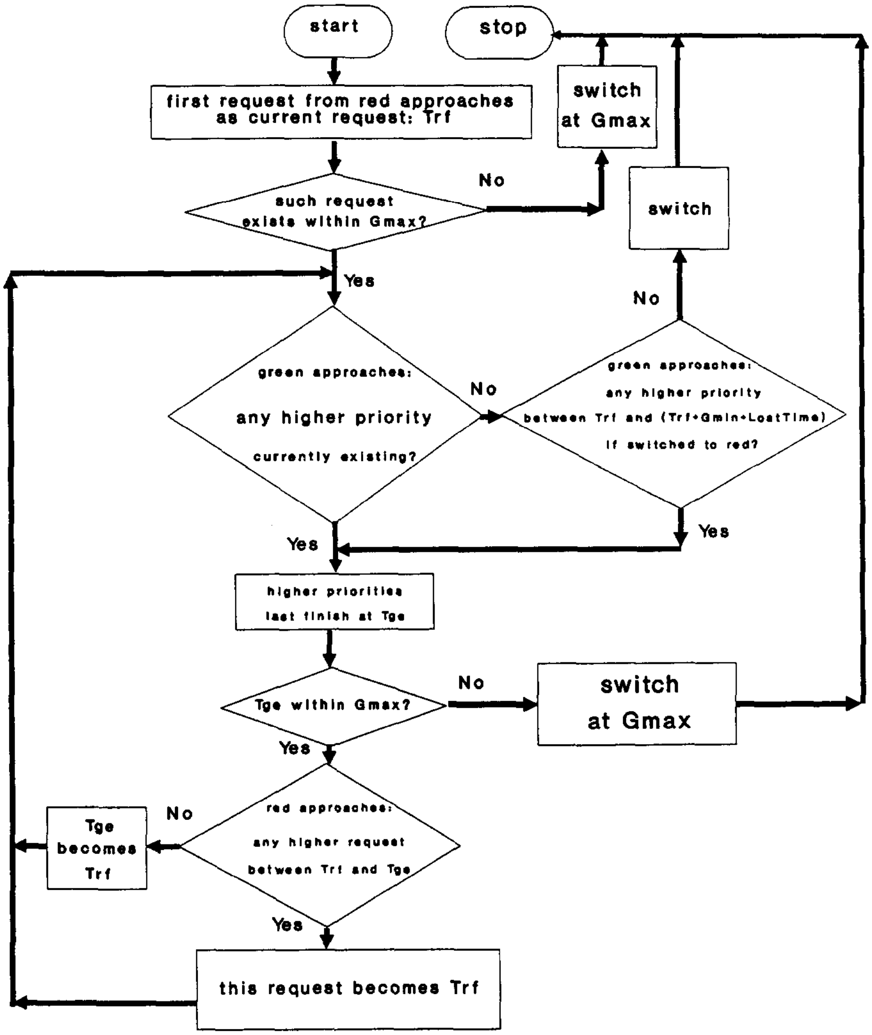
start
stop
first request from red approaches as current request: Trf
switch at Gmax
such request exists within Gmax?
No
switch
Yes
No
green approaches: any higher priority currently existing?
No
green approaches: any higher priority between Trf and (Trf+Gmin+LostTime) if switched to red?
Yes
Yes
higher priorities last finish at Tge
Tge within Gmax?
No
switch at Gmax
Yes
Tge becomes Trf
No
red approaches: any higher request between Trf and Tge
Yes
this request becomes Trf

Fig. 4. Flow chart for the control algorithm.

```
    PROJECTION PERIOD (Demand Detection & Signal Plan)
|---------------------------------------------|

    COMMITTED IMPLEMENTATION OF SIGNAL PLAN (5 seconds, say)
|----------|

                NEW PROJECTION PERIOD (Detection & Plan)
            |---------------------------------------------|

                NEW COMMITTED IMPLEMENTATION
            |----------|
```

Fig. 5. Rollover decision-renewal process.

sense, its "generated plan" plans ahead based on all of the flow information available to it, but the short-term operational decision which commits to only the renewal time span of about 5 sec is based on relatively precise time estimates from the nearside detectors.

THE SIMULATION TEST

The above-described procedure for modelling the mixed-traffic and transit operation at a busy intersection and determining timings for the traffic signal in response to detected private and transit vehicles, was given the name SPPORT (*S*ignal *P*riority *P*rocedure for *O*ptimization in *R*eal-*T*ime) and was programmed in the Borland C++ language. Its ability to develop and evaluate signal timings was tested by simulation for the afternoon peak hour at the intersection of Queen and Bathurst streets in Toronto.

Because this initial version of the model deals only with isolated intersections, it does not measure the precise effect of taking an intersection out of a fixed-time system. However, the comparisons that were made assumed that the existing operation provides reasonable fixed-time offsets, even though fixed-time coordination is difficult to achieve when transit vehicles continually interact with and delay private vehicles. Until recently, most traffic optimization models did not address the interaction of transit and traffic.

We tested SPPORT in two modes of operation to compare it with existing fixed-time operation. This was accomplished by giving it two different sets of priority lists to consider for the respective comparisons. In the first comparison, SPPORT used real-time detection to generate, evaluate, and select strategies but did not give priority to transit vehicles. In the second, SPPORT also considered the effects of real-time priority to each transit vehicle and evaluated the likely outcomes before deciding what the phasing should be.

Test site and traffic data

The intersection of Queen Street and Bathurst streets was chosen as the test site for evaluating (a) SPPORT itself and (b) real-time transit priority, because it is a busy intersection with large volumes of private vehicle traffic and the largest number of streetcars of all the intersections involved in Metropolitan Toronto's transit priority study. Using traffic flow data supplied by the City of Toronto, the daily peak hour from 4:45 pm to 5:45 pm was simulated under the existing fixed-time operation and SPPORT's real-time control, respectively. The layout of this intersection and the corresponding traffic data are as shown in Fig. 6, where the volumes are expressed in the form of private vehicles (cars), buses, and streetcars per hour.

A flow profile traffic generator was used to generate representative traffic flow data for the detectors based on the given volumes. This generator approximates the actual data, where the traffic is filtered through upstream signals to form partial platoons, which are then dispersed to varying degrees in travelling to the next intersection.

Simulation parameters

The following parameters were used for the simulation:

Number of vehicle detectors on each approach: 3;
(For this study, detectors were assumed at 250, 650, 1000 meters upstream of the intersection for all approaches)

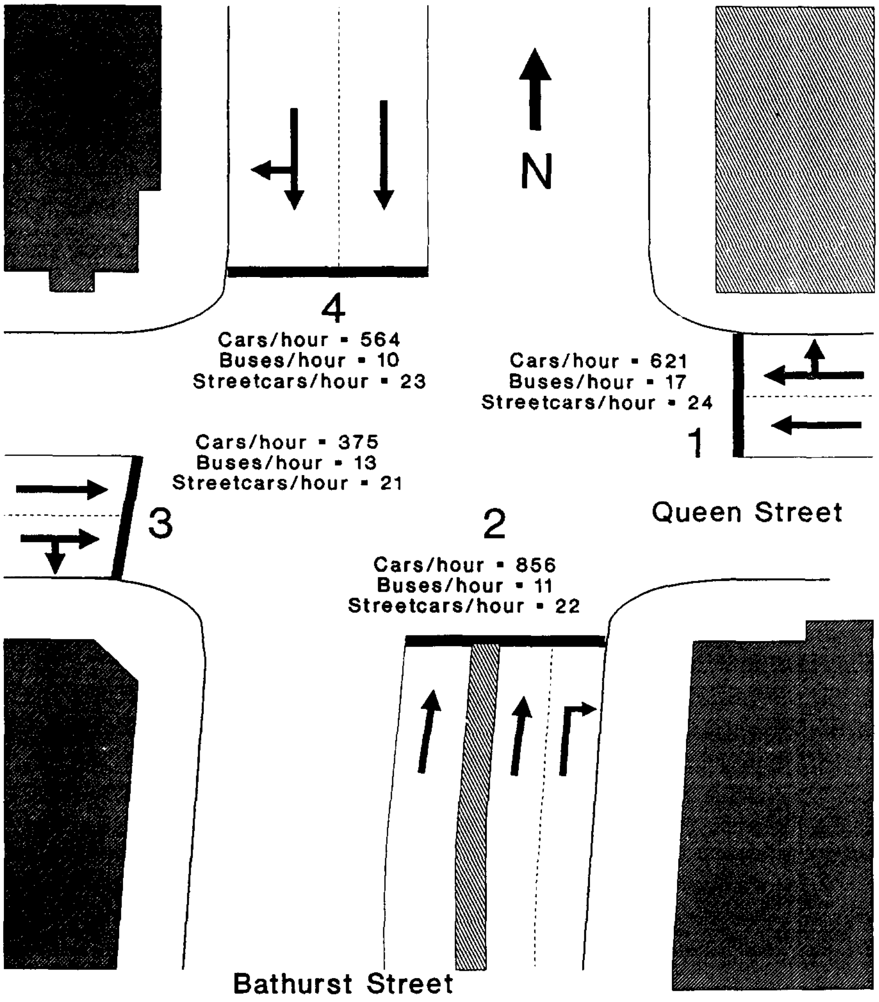

Fig. 6. The intersection of Queen Street and Bathurst Street.

Minimum green time
  Green phase for Queen Street: 22 sec;
  Green phase for Bathurst Street: 15 sec;
Maximum green time
  Green phase for Queen Street: 60 sec;
  Green phase for Bathurst Street: 60 sec;
Maximum extension after maximum green time
  Green phase for Queen Street: 40 sec;
  Green phase for Bathurst Street: 40 sec;
Amber time: 4 sec;
Gain on amber: 3 sec;
All-red period: 2 sec;
Lag time to become effective green: 2 sec;
  (therefore, lost time $= 4 + 2 + 2 - 3 = 5$ sec)
Transit stop position (assume nearside stops)
  Transit stop position on green when
  – no transit vehicles are loading in front: position 1;
  – transit vehicles are loading in front: $\leq$ position 4;
  Transit stop position on red: $\leq$ position 4;
  (in other words, the transit vehicle will open its doors if and only if it is in one of the

first four positions of the approach. When there are two lanes this means position one or two of the streetcar lane or bus lane.)

Car	Occupancy:	1.5 persons;
Streetcar	Occupancy:	60 persons;
Bus	Occupancy:	30 persons;

The existing fixed-time signal settings for the Queen and Bathurst intersection during the afternoon peak period (3:45 p.m.–6:15 p.m.) are:

Cycle time = 70 sec

North/South: green time = 29 sec, amber time = 4 sec, all red = 2 sec
East/West: green time = 29 sec, amber time = 4 sec, all red = 2 sec

We evaluated the intersection of Queen Street and Bathurst Street using this existing fixed-time signal plan, and compared it with SPPORT's timings.

To represent realistic variation in transit passenger loading times, we employed a distribution with loading times between a lower limit and an upper limit. For example, for the streetcars, the loading times are between 10 and 30 sec. Because the actual loading time for each streetcar is not known in advance, we can pre-estimate a representative time, such as the average passenger service time, for example, 20 sec, which is then used to generate the signal timings for each projection period. However, the model does simulate variable loading times in evaluating its performance and can respond to and represent shorter or longer loading times in its decision process, if that information can be relayed by way of an exit detector beyond the stopline or a ready-button on the streetcar, which the driver will use to indicate the time he is going to finish loading in advance; the model uses all available information to estimate when the streetcar will start and finish loading. For the simulated applications in Toronto, we used the following pattern of streetcar loading times for each approach, respectively:

first streetcar:	10 sec;	average loading time = 20 sec
second streetcar:	20 sec;	
third streetcar:	15 sec;	
fourth streetcar:	25 sec;	
fifth streetcar:	20 sec;	
sixth streetcar:	30 sec;	

And the pattern of loading times for buses is:

first bus:	0 sec;	average loading time = 10 sec
second bus:	10 sec;	
third bus:	5 sec;	
fourth bus:	15 sec;	
fifth bus:	10 sec;	
sixth bus:	20 sec;	

These patterns were repeated over and over, rather than selecting loading times at random. The traffic effect of this approximation is felt to be minor, i.e., this sequence will affect traffic to about the same extent as random variation in loading times. However, knowing the sequential pattern of streetcar loading times allows us to relate SPPORT's decisions to the transit arrivals, movement and loading, as we trace through the events in detail to determine whether the model is indeed using appropriate loading times and responding to them accordingly.

Of course SPPORT does not know the actual sequence of loading times in advance. Therefore, if detectors are placed only upstream of the transit loading point, SPPORT does not even have the a posteriori information regarding when the vehicle has finished loading, let alone a priori information of when it will finish, to assist it in planning efficient signal timings. As just discussed, these types of information might be provided by an exit detector immediately downstream of the loading position, or a ready-button on the vehicle.

SPPORT has incorporated routines for simulating and using these types of information if provided, as described next.

Using exit detector or ready-button information

The exit detector. Because in most cases, the actual transit loading times are unknown, SPPORT uses an average value to generate signal plans, although the effects of actual variable loading times are simulated when evaluating the performance index.

While an average dwell time can be used in generating the signal plans, it is useful to monitor the actual departure of transit vehicles. As noted, this can be done by using an extra exit-detector beyond the stopline.

Advance information—ready button. Although an exit-detector may be used to confirm the transit departures, it is also possible to have advance information on the transit loading times through other measures. This can be achieved by installing a ready-button on the transit vehicles so that the driver can inform the system in advance of the time when the loading will be completed. Therefore, the system can plan ahead to avoid unnecessary delays in switching the signal. SPPORT therefore uses a routine to utilize the exit detector information or ready-button information, if one is installed. It updates the estimated transit loading times based on the data available from the relevant device.

Test runs

We conducted test runs for four classes of strategies, which reflect different priority conditions, i.e.,

FT — fixed time;
NP — no transit priority;
TP — transit priority;
EP — transit priority, with exit detectors installed;
RP — transit priority, with ready-buttons installed;

The symbols NP, TP, EP, and RP are also used in Table 1 later to show the results.

Types of events

The same priority rules were used for each of the four tests. For this application 12 priority lists section were used. The lists for the nonpriority cases are obtained by merely taking out the events that give priority to transit. The types of events used in the priority lists are defined as follows:

1. Presence of an emergency vehicle;
2. Queue exceeds the maximum allowed queue;

Table 1. Delays estimated for simulated tests

		Private Vehicle Delay (Vehicle-Hour)	Bus Delay (Bus-Hour)	Streetcar Delay (Streetcar-Hour)	Total Person Delay (Person-Hour)
A	FT	13.33		0.86	71.58
	NP	18.58		0.83	68.54
	TP	14.17		0.75	66.26
	EP	13.74		0.72	64.07
	RP	12.62		0.63	56.83
B	FT	13.77	0.41	0.91	87.30
	NP	12.68	0.40	0.86	82.36
	TP	13.17	0.36	0.85	81.94
	EP	13.25	0.38	0.80	79.45
	RP	12.94	0.37	0.82	79.76

(A) No buses; (B) With buses.

3. Serving a queue at full saturation flow on all green approaches;
4. Serving a queue at full saturation flow on one green approach;
5. Queues on all green approaches, but all being served at reduced rates;
6. A streetcar arrives;
7. A streetcar loading;
8. A streetcar finishes loading;
9. A bus arrives;
10. A bus loading;
11. A bus finishes loading;
12. A queue request;
13. Cutting off the green extension after normal maximum green time.

For example, if its normal maximum green is 60 sec and the maximum allowable extension to this is specified to be 40 sec, then SPPORT will allow the green to extend up to 100 sec, unless no higher event in the priority list is currently in effect, in which case the green phase is truncated, provided that it has been on for at least 60 sec. To give an example, list 1 of the 12 ordered priority lists which use these events is as follows:

1. An emergency vehicle or the queue exceeds the maximum allowed queue;
2. Serving a queue at full saturation flow in at least one direction; serving a queue at a low rate in all directions;
3. A streetcar arrives or finishes loading;
4. A bus arrives or finishes loading;
5. A queue request;
6. A streetcar loading or a bus loading; or enforcing maximum green time.

Results of simulated tests

We assume that the existing fixed-time operation has reasonable coordination. Therefore, to be fair to existing operation, we evaluated several base starting times relative to the upstream signals and chose the best to represent the existing fixed-time signal timings.

The results of the best fixed-time signal timings appear in Table 1, along with SPPORT's real-time results. Case (a) considers the current situation where there are no buses, and Case (b) simulates a hypothetical case where buses are added into the system. Both Case (a) and Case (b) were tested with and without transit priority. The last column in Table 1 gives the total person delay which was calculated from the private vehicle delay and the streetcar delay, considering an average car occupancy of 1.5 persons and an average peak-hour bus occupancy of 30 persons and streetcar occupancy of 60 persons.

Table 1 indicates that SPPORT would reduce delays to transit by about 5% and to private traffic by slightly more if active transit priority was not used (i.e., streetcar delay would be reduced from 0.86 to 0.83 streetcar hours per hour when there are no buses. When there are buses, bus delay is reduced from 0.41 to 0.40 hours, and streetcar delay from 0.91 to 0.86). Total person delay is reduced from 71.58 to 68.54 person-hours per hour when there are no buses, and from 87.30 to 82.36 when there are buses, in each case by about 5%. The figures in Table 1 also indicate that if transit priority was implemented, the delays to transit would be reduced by about another 5%, at the expense of increased delay to private traffic. The total person delay would be further reduced marginally (from 68.54 to 66.26, and from 82.36 to 81.94, respectively) when transit priority was used. SPPORT predicts further savings in delay if an exit detector or advanced notification of the time of completion of loading are incorporated.

The simplest explanations for these savings is that by recognizing streetcars, SPPORT not only weights them to reflect their occupancy but also considers the residual effects of wasted capacity caused by loading during the green phase. Therefore, its evaluation routine has a more accurate representation of actual delays than those of models that do not recognize the effects caused by the on-line transit loading operation.

CONCLUSIONS

A modelling technique for traffic-responsive control with transit priority has been developed. By generating control strategies based on finite priority lists, efficient signal-switching points can be developed and evaluated. Therefore, practical decisions can be made within a reasonable time interval.

The process can give absolute priority to transit, no priority at all, or appropriate consideration based on weights given to the transit vehicles and its own pre-evaluation of various scenarios.

The ability to represent the traffic impacts of on-line transit loading also allows the model to serve as an evaluation tool for comparing different decision policies. Moreover, its ability to use upstream information offers the potential for communication among neighboring intersections, as it is otherwise difficult to achieve coordination when streetcars are present.

The major difference between SPPORT and true optimization models such as OPAC is that the SPPORT approach has deliberately sacrificed the feasibility of using a structured optimization procedure such as dynamic programming to accurately represent major discrete noncyclical effects such as the on-line transit loading process. Other major differences from OPAC are: (a) rather than using smoothed data (obtained from a traffic model) for the projection period beyond the lead time of the nearside detector, SPPORT relies on an upstream detector; (b) instead of dividing the projection period into a finite number of steps (e.g., a 90-sec projection period may be divided into 18 successive 5-sec steps), SPPORT models the discrete event times directly.

FURTHER WORK

The lists of ordered important event types provided by experts can be improved by evaluating and learning from such events over time. For example, SPPORT can learn which orders of event priority work best in certain situations in a stochastic environment. Because transit vehicles have a random loading time which is not known in advance, it may be best to plan for other than an average loading time. For example, it may be best to switch to or from green for the transit vehicles at some time during the loading process rather than risk delaying a loaded transit vehicle. SPPORT can then "remember" such types of strategic events and can adapt from them to search for other important event types.

Finally, it may be beneficial to test different values of the parameters and to modify the performance measure used for evaluating different control strategies. For example, SPPORT can consider both stops and delays.

Acknowledgements—We thank the constructive suggestions of the anonymous referees. The basic research was financially supported by the Natural Science and Engineering Research Council and the application by the Metropolitan Toronto Transportation Department.

REFERENCES

Bang, K. L. (1976). Optimal control of isolated traffic signals. *Traffic Engineering & Control*, **17**(7), 288–292.

Bell, M. G. H., Cowell, M. P. H., and Heydecker, B. G. (1990). Traffic-responsive signal control at isolated junctions. In S. Yagar & E. Rowe, eds., *Traffic control methods*. New York: Engineering Foundation Press; pp. 273–294.

Cansult Engineering Limited. (1991). *Mainline traffic signal priority study. Phase V-Demonstration Project*. Toronto, Ontario, Canada.

Cornwell, P. R. (1986). Tram priority in SCATS. *Traffic Engineering & Control*, **27**(11), 561–574.

De Groot, P. (1981). *Advanced traffic responsive intersection control strategies*. MASc Thesis, Department of Civil Engineering, University of Waterloo, Waterloo, Ontario.

Gartner, N. H. (1983). OPAC: A demand-responsive strategy for traffic signal control. *Transp. Res. Record*, **906**, 75–81.

Heydecker, B. G. (1990, July). *A continuous-time formulation for traffic-responsive signal control*. Paper presented to the 11th International Symposium on Transportation and Traffic Theory, Yokohama, Japan.

Lin, F. B., Wang, N., and Vijayakumar, S. (1988). Development of an intelligent adaptive signal logic. In S. Yagar, ed., *Management and control of urban traffic systems*. New York: Engineering Foundation Press; pp. 257–279.

Miller, A. J. (1963). *A computer control system for traffic network*. Proceedings of the Second International Symposium on the Theory of Road Traffic Flow (Ed. J. Almond), OECD Paris, pp. 200–220.

Richbell, L. E. and Van Averbeke, B. A. (1972). Bus priorities at traffic control signals. *Traffic Engineering & Control*, **13**(6), 70–75.

Vincent, R. A. and Peirce, J. R. (1988). MOVA: Traffic responsive, self-optimising signal control for isolated intersections. *TRRL Report RR*, **170**, Crowthorne, England.

Vincent, R. A. and Young, C. P. (1986). Self-optimising traffic signal control using microprocessors – the TRRL 'MOVA' strategy for isolated intersections. *Traffic Engineering & Control*, **27**(7–8), 385–387.

Wattleworth, J. A. (1977). Evaluation of bus-priority strategies on Northwest Seventh Avenue in Miami. *Transpn. Res. Record*, **626**, pp. 32–35.