Pergamon

Transpn Res.-C, Vol. 5, No. 3/4, pp. 153–163, 1997

0968-090X/97 $17.00+0.00

PII: S0968-090X(97)00012-0

TRAFFIC-RESPONSIVE SIGNAL TIMING FOR SYSTEM-WIDE TRAFFIC CONTROL

JAMES C. SPALL and DANIEL C. CHIN

The Johns Hopkins University, Applied Physics Laboratory, Laurel, Maryland 20723-6099, U.S.A.

(Received 9 August 1996; in revised form 15 July 1997)

Abstract—A long-standing problem in traffic engineering is to optimize the flow of vehicles through a given road network. Improving the timing of the traffic signals at intersections in the network is generally the most powerful and cost-effective means of achieving this goal. However, because of the many complex aspects of a traffic system—human behavioral considerations, vehicle flow interactions within the network, weather effects, traffic accidents, long-term (e.g. seasonal) variation, etc.—it has been notoriously difficult to determine the optimal signal timing. This is especially the case on a system-wide (multiple intersection) basis. Much of this difficulty has stemmed from the need to build extremely complex models of the traffic dynamics as a component of the control strategy. This paper presents a fundamentally different approach for optimal signal timing that eliminates the need for such complex models. The approach is based on a neural network (or other function approximator) serving as the basis for the control law, with the weight estimation occurring in closed-loop mode via the simultaneous perturbation stochastic approximation (SPSA) algorithm. The neural network function uses current traffic information to solve the current (instantaneous) traffic problem on a system-wide basis through an optimal signal timing strategy. The approach is illustrated by a realistic simulation of a nine-intersection network within the central business district of Manhattan, New York. © 1997 Elsevier Science Ltd. All rights reserved.

Keywords: adaptive control, transportation systems, traffic signal timing, simultaneous perturbation stochastic approximation (SPSA), neural networks.

1. INTRODUCTION

A major component of advanced traffic management for complex road systems is the timing strategy for the signalized intersections. This is an extremely challenging control problem at a system (network)-wide level. For present purposes:

> *System-wide control* is the means for real-time (demand-responsive) adjustment of the timings of all signals in a traffic network to achieve a reduction in overall congestion consistent with the chosen system-wide measure of effectiveness (MOE). This real-time control is responsive to instantaneous changes in traffic conditions, including changes due to accidents or other traffic incidents. Further, the timings should change automatically to adapt to long-term changes in the system (e.g. street reconfiguration or seasonal variations). To achieve true system-wide optimality, the timings at different signals will not generally have a predetermined relationship to one another.*

To the best knowledge of the authors, no existing or planned approach achieves such system-wide control. This paper presents an approach—S-TRAC (System-wide Traffic-Adaptive Control)—for treating this challenging problem.

All attempts known to us for real-time demand responsive control either are optimized only on a per-intersection basis or make simplifying assumptions to treat the multiple-intersection problem. An example of the former is OPAC (Gartner *et al.*, 1991) while examples of the latter include SCOOT (Hunt *et al.* 1981; Martin and Hockaday, 1995) and REALBAND (Dell'Olmo and Mirchandani, 1995). SCOOT's system-wide (i.e. multiple, interconnecting artery) approach is limited to broad strategy choices from one traffic corridor to another rather than a co-ordinated set of signal parameter selections for the entire network. Hence, although SCOOT may be

*One notable exception to this would be for those signals along one or more arteries within the system to synchronize the timings, where it is desirable.

implemented on a full traffic system, it is not a true system-wide controller in the sense considered here. "[SCOOT's] regional boundaries are satisfactory for zoned control, but fail to offer wide-spread strategic control" (Martin and Hockaday, 1995). The other multiple intersection technique mentioned above, REALBAND, provides a way to improve platoon progression, which the other techniques apparently lack. However, REALBAND is limited in its application to types of traffic patterns for which vehicle platoons are easily identifiable and, thus, may not perform well in heavily congested conditions with no readily identifiable platoons. Note that none of these techniques incorporate a method to automatically self-tune over a period of weeks or months. In addition, most approaches to traffic control have been developed independent of modern techniques in nonlinear stochastic control (notable exceptions to this for *freeway* traffic control are Messmer and Papageorgiou (1994) and Papageorigiou *et al.* (1995)).

The essential ingredient in these and other modern attempts to provide optimal signal timings for single or multiple intersections is a model for the traffic behavior. However, the problem of fully modeling traffic at a system-wide level is daunting: "To develop a 'general theory' for the stochastic behavior of a traffic system is out of the question. Even if it were possible such a theory would be so complex as to be of no practical value." (Newell, 1989, p. 258). In the OPAC, SCOOT, and REALBAND approaches discussed above, the models used are in the form of traditional equation-based relationships, but it is also possible to use other model representations such as a neural network (Nataksuji and Kaku, 1991), fuzzy associative memory matrix (Kelsey and Bisset, 1993), or rules base for an expert system (Ritchie, 1990). The signal timings are then based on relationships (algebraic or otherwise) *derived from* the assumed model of the traffic dynamics. For real-time (demand-responsive) approaches, this relationship (or 'control function') takes as input information about current traffic conditions and produces as output the timings for the signals. However, to the extent that the traffic dynamics model is flawed or oversimplified, the signal timings will be suboptimal.

The unique aspect of the S-TRAC control strategy here is that it *does not* require a system-wide traffic-dynamics model (this model avoidance is possible through use of a powerful method in stochastic optimization, as discussed in Sections 2–4 below). S-TRAC is based on a neural network or other function approximator for use *in the control function*; no model (e.g. set of differential equations or a second neural network) is needed for the traffic dynamics. Thus, in S-TRAC, there are no requirements to build equations describing critical traffic elements such as complex flow interactions among the arteries in the presence of traffic congestion, weather-related changes in driving patterns, flow changes as a result of variable message signs or radio announcements, etc. The extreme difficulty in mathematically describing such critical elements of the traffic system will inherently limit any control strategy that requires a model of the traffic dynamics, which is the implication of the Newell (1989) quote above. Related to this is the non-robustness of system model-based controls to operational traffic situations that differ significantly from situations represented in the data used to build the system model (this non-robustness can sometimes lead to unstable system behavior). Further, even if a reliable system model could be built, a change to the scenario or measure-of-effectiveness (MOE) would typically entail many complex calculations to modify the model and requisite optimization process.

In addition to the above considerations, system-wide control (as defined in the first paragraph) requires that the controller automatically adapt to the inevitable long-term (say, month-to-month) changes in the system. This is a formidable requirement for the current model-based controllers as these long-term changes encompass difficult-to-model aspects such as seasonal variations in flow patterns on all links in the system, long-term construction blockages or lane reconfigurations, changes in the number of residences and/or businesses in the system, etc. In fact, in the context of the Los Angeles traffic system, Rowe (1991) notes that the difficulty in adapting to long-term system changes is a major limitation of current traffic control strategies. By avoiding the need for a system model, however, S-TRAC is able to produce a controller that generates optimal instantaneous (minute-to-minute) signal timings while automatically adapting to long-term (month-to-month) system changes.

Central to S-TRAC is the use of the simultaneous perturbation stochastic approximation (SPSA) algorithm (Spall, 1992). SPSA provides a highly efficient means for estimating parameters without the need for the gradient of the underlying performance measure (with respect to the parameters being estimated). In the context of control problems, requiring the gradient vector is tantamount to requiring a model of process (see, e.g. Spall and Cristion, 1994, 1995, 1997 for further discussion).

The remainder of this paper is organized as follows: Section 2 presents an overview of the S-TRAC approach, including the relationship between the demand-responsive instantaneous traffic controller and the long-term SPSA training process, and some of the practical issues associated with algorithm initialization and calculation of the measurement of effects. Section 3 discusses the SPSA algorithm for traffic control and Section 4 translates the principles in Section 2 and Section 3 into a step-by-step implementation guide. Section 5 illustrates S-TRAC for a nine-intersection network in mid-town Manhattan and Section 6 offers some concluding remarks.

2. OVERVIEW OF S-TRAC CONTROL STRATEGY

2.1. Summary

S-TRAC is based on developing a mathematical function, say $u(\bullet)$, that takes current information on the state of the traffic conditions and produces the timings for all signals in the networks to optimize the performance of the system. (A dot shown here and throughout the rest of this paper as an argument in a mathematical function represents all relevant variables entering the function.) The inputs may include sensor readings from throughout the traffic system and other relevant information such as weather and time-of-day. The output values for each of the signals in the network may be any of the usual timing quantities: e.g. red-green splits, offsets, and cycle times.

The traffic control function $u(\bullet)$ in S-TRAC is implemented by a neural network (NN) for which the internal NN connection weights are estimated and refined by an *on-line* training process. These weights will fully define a function that takes sensor information on current traffic conditions and produce the optimal system-wide timings.* It is within these weights that information about the optimal control strategy is embedded. To reflect reality, it is important that the weights contain information to facilitate a response to traffic conditions (including accidents or other incidents). The weights are able to evolve in the long-term (say, month-to-month) in accordance with the inevitable changes in the transportation system. Hence, the values of the weights are absolutely critical to this framework.

Figure 1 illustrates the overall operation of S-TRAC. The lower loop provides the real-time feedback on traffic conditions for use by the NN controller (with specified weights) in providing real-time signal commands. The upper loop is the weight estimation path that refines the real-time control. This loop operates on a day-to-day basis and can be turned on and off as needed to build the NN controller and to self-tune the controller to long-term changes in the system. At the heart of the upper-loop weight training is the SPSA algorithm†, which provides a highly efficient and

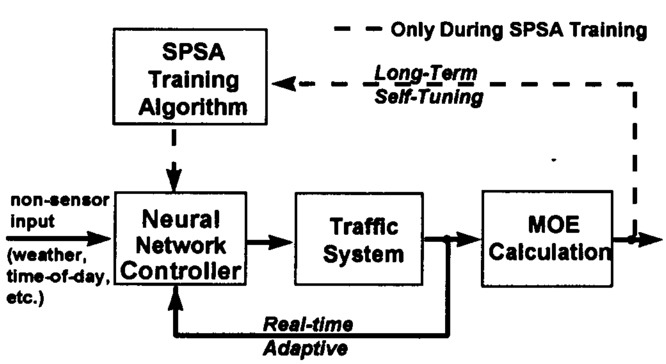

Fig. 1. S-TRAC configuration: relationship between traffic system, controller, and training algorithm.

*Theory given in, say, Funahashi (1989) shows that any reasonable mathematical function can be approximated to a high level of accuracy by a NN *if* (and only if) the weights are properly estimated. In our case, the NN is being used to

†Note that SPSA is fundamentally different from infinitesimal perturbation analysis (IPA) (or other PA approaches) although the algorithms share one word in their names. SPSA uses only loss function evaluations in its optimization while IPA uses the gradient of the loss function. For traffic control problems, requiring the gradient is equivalent to requiring a network-wide model of the system; evaluating the loss function alone does not require a model. The lack of a gradient also precludes the use of such standard NN training algorithms as backpropagation.

relatively easy-to-implement means of estimating the NN weights θ on an on-line basis. The use of SPSA in the day-to-day training will be presented in detail in Sections 3 and 4.

In implementing S-TRAC, a different specific NN structure (number of inputs/outputs, number of weights, etc.) may be chosen to produce controls during each of several periods within a 24 h time-frame. The periods should be chosen so that system-wide traffic patterns are *roughly* consistent over the period. For example, a 24 h time-frame may be divided into five periods: 5:30 AM–9:30 AM, 9:30 AM–3:30 PM, 3:30 PM–7:30 PM, 7:30PM–11:30 PM, and 11:30 PM–5:30 AM, each of which will have a separate NN controller. Hence, the controller illustrated in Fig. 1 pertains to one time period of interest. In principle, it would be possible to have one NN for a full 24 h period, but such a NN may be excessively complex due to the wide variety of traffic conditions over a full day (and further a fixed timing plan may be sufficient for the time periods 9:30 AM–3:30 PM and 11:30 PM–5:30 AM).

2.2. *Some practical issues*

The upper loop (weight training process) in Fig. 1 will continue as long as needed to achieve effective convergence of the weight estimate; convergence is obtained when the MOE has been optimized subject to constraints on road capacity, minimum signal phase length, etc. While the SPSA training is occurring, only minor controller-imposed variations in traffic flow (from what would have occurred based on the previous day's timing strategy) will be seen, which should be unnoticed by most drivers. After training is complete for a given period, there will be a control function $u(\bullet)$ (based on a converged value of weights θ) that provides optimal signal timings for any specific time within the period given the current traffic conditions.* Note that the training is based on adjacent days having similar *mean* traffic behavior within the time period of interest (the *actual* traffic conditions are allowed to vary significantly day-to-day in line with the usual stochastic effects); so, for example, there may be a recursion for weekdays (perhaps with a special 'tag' for Friday evenings to accommodate the extra flow if that was significant) and another corresponding recursion (and associated NN) for weekends/holidays.

As part of the training process, an initial set of values (prior to running SPSA) must be chosen for the NN weights (these yield the control strategy on 'day 0' of the training process). It will generally be desirable to initialize the weights to produce a NN control with the same timing strategy as the traffic system had in place prior to the implementation of S-TRAC (this allows S-TRAC to take advantage of the 'tuning' and prior information embedded in the prior strategy). For a fixed time-of-day strategy, this is straightforward, though the specification of 'bias weights' on the NN output (with other weights, except those linking time-of-day if that input is used, zeroed out). For a demand-responsive prior strategy, one could use current and recent-past data on traffic flow and corresponding (flow dependent) signal timings in conjunction with standard ('off the shelf') back-propagation-type software. This will generate a NN controller that is able to reproduce the timing strategy embedded in these data. Then the SPSA optimization process will begin with that strategy and improve from there. We must emphasize that this off-line analysis is done *only* to initialize the weights in the algorithm. Alternatively (or supplementarily) 'pseudo historical' data could be generated by running traffic simulations (say, based on the well-known U. S. Federal Highway Administration-sponsored TRAF software collection) together with corresponding 'reasonable' (flow-dependent) signal timings. These pseudo historical data could then be used with back-propagation (as with the real historical data) to generate the initial weights.

One appealing feature in using simulations for initialization is that it is possible to introduce 'incidents' (accidents, break-downs, special events, etc.) that may not have been encountered in other initialization information (e.g. historical data); having this incident information embedded in

*We must emphasize that although there is a fixed value of θ after training is complete, the signal timings given by $u(\bullet)$ will generally change throughout the period—possibly on a cycle-to-cycle basis—to adapt to instantaneous fluctuations in traffic conditions, i.e. the *function* $u(\bullet)$ is the same during the time period of interest, but the specific output *values* of $u(\bullet)$ will change during the period as the traffic conditions change. If necessary, this idea can perhaps be made clearer by viewing the NN control $u(\bullet)$ with specified weights as analogous to a polynomial function with specified coefficients. For a *fixed* set of coefficients, the value of the polynomial will change as the value of the independent variable changes. In contrast, a change in the coefficient values represents a change in the polynomial function itself. The former case is analogous to what happens in producing instantaneous controls for a fixed weight vector (the lower loop in Fig. 1) and the latter case is analogous to what happens as the NN undergoes its day-to-day training (the upper loop in Fig. 1).

the initial weights may help the real-time NN controller cope with similar incidents in real operations after day 0. It is not required that all possible incident scenarios be introduced in the simulation since the NN (in principle, at least) can interpolate to unencountered incidents if the initialization information contains a reasonable variety of plausible incidents.

Periodically, after effective convergence for θ has been achieved (and the controller is operating without the use of SPSA, i.e. the upper loop in Fig. 1 is disconnected), the training should be turned 'on' in order to adapt the weights to the inevitable long-term changes in the traffic system and flow patterns. (The reason that it is not recommended to run training continuously day-to-day is that when the training is operative, the weight values θ used in the controller are slightly perturbed from those that the algorithm has currently found to be optimal.) This updating can be done relatively easily without the need to do the expensive and time-consuming off-line modeling that is required for standard model-based approaches to traffic control (e.g. in the context of the Los Angeles traffic system, Rowe (1991) points out that the adaptation to long-term changes is not done as frequently as necessary because of the high costs and extreme difficulty involved). Whether the training in SPSA is 'on' or 'off' should be invisible to most drivers.

3. THE MATHEMATICAL ALGORITHM: SPSA-BASED TRAINING

The above discussion outlines how NN functions for real-time traffic control can be constructed by setting up a recursion that iterates on a day-to-day basis for a fixed time period. The discussion here will provide the mathematical form of the recursion. Given the set of weights to be determined, we let $\hat{\theta}_k$ denote the estimate of θ at the kth iteration of the SPSA algorithm. The aim of the SPSA algorithm is to find that set of weight values that minimizes some 'loss function', which is directly related to optimizing the MOE. Mathematically, this is equivalent to finding a weight value such that the gradient of the loss function with respect to the weights is zero. However, since we are *not* assuming a model for the traffic dynamics, it is not possible to compute this gradient for use in standard NN optimization procedures such as backpropagation.

The SPSA algorithm is based on forming a succession of highly efficient approximations to the uncomputable gradient of the loss function in the process of finding the optimal weights. The SP gradient approximation used in SPSA only requires observed values of the system (e.g. loop detector counts, traffic queues, wait times, pollutant emission readings, etc.). The theoretical and numerical properties of the SPSA algorithm are thoroughly described in Spall (1992). The high efficiency of SPSA relative to competing (gradient-free) SA algorithms is established in Spall (1992) and Chin (1993, 1997). The application of SPSA to NN controller design has been considered in Spall and Cristion (1994, 1995, 1997). (The theoretical properties related to algorithm convergence in Spall (1992) provide a guarantee that SPSA will work properly in a wide variety of practical conditions; this contrasts with many other algorithms proposed for adaptive traffic control, which are ad hoc and have only been demonstrated on a limited set of test cases.) The SPSA algorithm for estimating θ has the form:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k(\hat{\theta}_k) \tag{1}$$

where a_k is a scalar gain coefficient and $\hat{g}_k(\hat{\theta}_k)$ is the SP gradient estimate at $\theta = \hat{\theta}_k$. Note that eqn (1) states that the new estimate of θ is equal to the previous estimate plus an adjustment that is proportional to the negative of the gradient estimate. The initial value $\hat{\theta}_0$ may be chosen according to the discussion of subsection 2.2.

To calculate the most critical part of eqn (1)—i.e. the gradient approximation $\hat{g}_k(\theta)$ for any θ—we must define an underlying loss function $L(\theta)$. This loss function is directly related to the MOE, and mathematically expresses the MOE criteria. The form of $L(\theta)$ reflects the particular system aspects to be optimized and/or the relative importance to put on optimizing several criteria at once (e.g. mean queue length or wait times at intersections, traffic flow along certain arteries, pollutant emissions, etc.). Because of the variety of MOE criteria considered in practice, the specific form of $L(\theta)$ will be allowed to be flexible in this paper. An *example* loss function might be a standard quadratic measure such as

$$L(\theta) = E[x^T x \mid \theta] \tag{2}$$

where

- $E(\bullet \mid \theta)$ denotes an expected value conditional on the set of controls with weights θ;
- x represents the system state vector, e.g. vector of mean queue lengths or mean vehicle wait times at all intersections within the time period of interest (the state depends on θ through the fact that the control used in affecting the state x depends on θ).

Given a definition of the loss function (as derived from the MOE), the critical step in implementing the SPSA algorithm in eqn (1) is to determine the gradient estimate $\hat{g}_k(\theta)$ of any value of θ. This embodies the key and unique technical contribution of our approach since $\hat{g}_k(\theta)$ does not require a complete model for the system-wide traffic dynamics. Assuming that θ is p-dimensional, the gradient estimate at any θ has the form

$$\hat{g}_k(\theta) = \begin{bmatrix} \frac{\hat{L}(\theta+c_k\Delta_k)-\hat{L}(\theta-c_k\Delta_k)}{2c_k\Delta_{k1}} \\ \bullet \\ \bullet \\ \bullet \\ \frac{\hat{L}(\theta+c_k\Delta_k)-\hat{L}(\theta-c_k\Delta_k)}{2c_k\Delta_{kp}} \end{bmatrix} \tag{3}$$

where $\hat{L}(\bullet)$ denotes an observed (sample) value of $L(\bullet)$, $\Delta_k = (\Delta_{ki}, \Delta_{k2}, \ldots, \Delta_{kp})$ is a user-generated vector of random variables that satisfy certain important regularity conditions, Spall (1992), Spall and Cristion (1994, 1995, 1997); having $\Delta_{ki} = \pm 1\ \forall k, i$ with probability 1/2 of each outcome satisfies these conditions and is used in the study of Section 5 below), and c_k is a small positive number. Note that the numerators in the p components of $\hat{g}_k(\theta)$ are identical; only the denominators change. Hence, to compute $\hat{g}_k(\theta)$, one only needs *two* values of $\hat{L}(\bullet)$ *independent* of the dimension p. Note also that SPSA (as a *stochastic* approximation algorithm) is designed specifically to deal with day-to-day stochastic variations in traffic conditions. The mathematical manifestation of this property is that SPSA will converge even though $\hat{L}(\bullet) \neq L(\bullet)$ in general.

The SPSA approach is in contrast to the standard approach for approximating gradients (the 'finite-difference' method), which requires $2p$ values of $\hat{L}(\bullet)$, each representing a positive or negative perturbation of one element of θ with all other elements held fixed. In the context of traffic control, each value of $\hat{L}(\bullet)$ represents data collected during one time period (within one 24 h period). For traffic control, the dimension p is *at least* as large as the total number of factors to be controlled within the traffic system (e.g. in a system with 100 signals and an average of four control factors per light, $p \geq 400$). Hence, the SPSA method is easily two to three orders of magnitude more efficient than the standard finite-difference method in finding the optimal weights for most realistic traffic settings. Theory in Spall (1992) and Chin (1993, 1997) rigorously justifies this gain in efficiency. (In particular, it is shown that the SPSA method and the finite-difference method achieve a given level of accuracy in estimating θ in the same number of iterations, which translates into a p-fold total savings in $\hat{L}(\bullet)$ evaluations since each iteration of SPSA requires only $1/p$ the number of $\hat{L}(\bullet)$ evaluations as finite-difference.)

4. STEP-BY-STEP IMPLEMENTATION OF SPSA TRAINING ALGORITHM FOR S-TRAC

Let us now present a step-by-step summary of how the SPSA algorithm in eqns (1) and (3) would be implemented to achieve optimal traffic control in the system-wide setting. This summary pertains to building up the controller (i.e. estimating a θ) for one time period, as illustrated in Fig. 1 above. Obviously, the same procedure would apply in the other periods. Starting with some $\hat{\theta}_0$ (see the discussion in subsection 2.2) the step-by-step procedure for updating $\hat{\theta}_k$ to $\hat{\theta}_{k+1}$ is:

1. Given the current weight vector estimate $\hat{\theta}_k$, change all values to $\hat{\theta}_k + c_k\Delta_k$ where c_k and Δ_k satisfy conditions in Spall (1992) or Spall and Cristion (1994, 1995, 1997).
2. Throughout the given time period, use a NN control $u(\theta, \bullet)$ with weights $\theta = \hat{\theta}_k + c_k\Delta_k$. Inputs to $u(\theta, \bullet)$ at any time within the period include current and recent past state information (e.g. queues at intersections), previous controls (signal parameter settings), time-of-day, weather, etc.

3. Monitor system throughout time period (and possibly slightly thereafter) and form sample loss function $\hat{L}(\hat{\theta}_k + c_k\Delta_k)$ based on observed system behavior. For example, with the loss function in eqn (2), we have

$$\hat{L}(\hat{\theta}_k + c_k\Delta_k) = x^T x$$

where the state values are based on the controls $u(\hat{\theta}_k + c_k\Delta_k, \bullet)$ used throughout the period (a possible state vector might include the queues of all intersections over a set of sampling time in the overall time periods).
4. During the same time period on following like day (e.g. weekday after weekday), repeat steps 1–3 with $\hat{\theta}_k - c_k\Delta_k$ replacing $\hat{\theta}_k + c_k\Delta_k$. Form $\hat{L}(\hat{\theta}_k - c_k\Delta_k)$.
5. With the quantities computed in steps 3 and 4, $\hat{L}(\hat{\theta}_k + c_k\Delta_k)$ and $\hat{L}(\hat{\theta}_k - c_k\Delta_k)$, form the SP gradient estimate in eqn (3) and then take one iteration of the SPSA algorithm in eqn (1) to update the value of $\hat{\theta}_k$ to $\hat{\theta}_{k+1}$.
6. (Optional) During same period on following like day, use a NN control with updated weights $\theta = \hat{\theta}_{k+1}$. This provides information on performance with current updated weight estimates (no perturbation); this information, is not explicitly used in the SPSA updating algorithm.
7. Repeat steps 1–6 with the new value $\hat{\theta}_{k+1}$ replacing $\hat{\theta}_k$ until traffic flow is approximately optimized (or at least sufficiently improved) based on the chosen MOE.

There are several practical aspects of the above procedure that are worth noting. By initializing the weight vector at a value $\hat{\theta}_0$ that is able to produce the initial signal timings actually in the system (see Section 3), the algorithm will tend to produce signal timings that are between the initial and improved timings while it is in the training phase. Hence, there will likely be no significant control-induced disruption in the traffic system during the training phase. After the weight estimates have effectively converged (so we have a controller that produces improved signal timings for given traffic conditions), the algorithm may be turned 'on' or 'off' relatively easily without the need to perform detailed off-line modeling. It would, of course, be desirable to turn the algorithm 'on' periodically in order to adapt to the inevitable long-term changes in the underlying traffic flow patterns. A further point to note in using SPSA is that there will be some coupling between traffic flows in adjacent time periods within a 24 h time-frame. This is automatically accounted for by the fact that inputs to $u(\bullet)$ include previous states and controls (even if they are from the previous period). Hence, even though there are separate SPSA recursions (and neural networks) for each of the time periods, information is passed across periods to ensure true optimal performance.

5. EXAMPLE OF S-TRAC IMPLEMENTATION IN MANHATTAN

5.1. Introduction

This section illustrates by simulation an application of the S-TRAC approach to a nine-intersection network in mid-town Manhattan, NY. The small-scale realistic example here is intended to be illustrative of the ability of S-TRAC to address larger-scale traffic systems and is not entirely trivial as it considers a congested (saturated) traffic network and includes nonlinear, stochastic effects. The simulation was calibrated based an actual Manhattan traffic data, as discussed in subsection 5.2.

We are considering control for one 4 h time period and are estimating, across days, the NN weights for the collective set of traffic signal responses to instantaneous traffic conditions during this 4 h period. The software used here is described in detail in Chin and Smith (1994); the simulation was conducted on a Pentium-based PC using C+ +. The traffic dynamics were simulated using state-space flow equations similar to those in Papageorgiou (1990) or Nataksuji and Kaku (1991) with Poisson-distributed vehicle arrivals at input nodes into the network. Of course, consistent with the fundamental S-TRAC approach as it would be applied in a real system, the controller does *not* have knowledge of the equations being used to generate the simulated traffic flows. The traffic simulation here is being applied as a surrogate for the real traffic system; SPSA on-line training in a real system would not require a traffic simulation. The controller is constructed via SPSA by the efficient use of small system changes and observation of resulting system performance. Recall that SPSA is explicitly designed to account for stochastic variations in the traffic flow in creating the NN weight estimates. This simulation will illustrate this capability.

5.2. The simulated traffic flow and form for NN controller

Two studies were conducted for a simulated 90-day period: one with constant mean Poisson distributed arrival rates over the total period, and another with a 10% step increase in all mean arrival rates into the network (not including the internal egress discussed below) at day 10 during the total period. In both studies, the simulated traffic network runs between 55th and 57th Streets (North and South) and from 6th Avenue to Madison Avenue (East and West) and therefore includes nine intersections with 5th Avenue as the central artery. Figure 2 depicts the scenario. The time of control covers the 4 h period, from 3:30 PM to 7:30 PM, which represents evening rush time. The technique could obviously be applied to any other period during the day as well. In the 4 h control period several streets have their traffic levels gradually rising and then falling. Their traffic arrival rates increase linearly from non-rush hour rates starting at 3:30 PM The rates peak at 5:30 PM to a rush hour saturated flow condition and then subside linearly until 7:30 PM Back-up occurs during some of the 4 h period in the sense that queues do not totally deplete during a green cycle. Nonlinear, flow-dependent driver behavioral aspects are embedded in the simulation. (e.g. the probabilities of turns of intersections are dependent on the congestion levels of the through street and cross street). Some streets have unchanging traffic statistics during the total time period while others have inflow rates from garage-generated egress at the end of office hours from 4:30 PM to 5:30 PM. The simulation and baseline fixed time controller have been extensively tested to ensure that they produce traffic volumes that correspond to actual recorded data for the Manhattan traffic sector as given in Rathi (1988). [A complete discussion of the development and testing of the baseline simulation and the details of its operation are given in Chin and Smith (1994).]

For S-TRAC, we used a two-hidden-layer, feed-forward NN with 42 input nodes. The 42 NN inputs were (i) the queue levels* at each cycle termination for the 21 traffic queues in the simulation, (ii) the per-cycle vehicle arrivals at the 11 external nodes in the system, (iii) the time from the start of the simulation, and (iv) the nine outputs from the previous control solution. The output layer had nine nodes, one for each signal's green/red split. The two hidden layers had 12 and 10 nodes, respectively. For this NN, there were a total of 745 NN weights that must be estimated.

In response to current traffic conditions, the controller determines the green/red split for the succeeding cycle of each of the nine signals in the traffic network. Each signal operates on a fixed 90 s cycle as discussed in Rathi (1988) (in a full implementation of S-TRAC, cycle length for each signal could also be a control variable). The controller operates in a real-time adaptive mode in which its cycle-by-cycle responses to traffic fluctuations are gradually improved, over a period of several days or weeks, based on an MOE (i.e. loss function) consisting of the summed square values of the cycle-traffic-wait time at each intersection over the daily 4 h period. Note that since the underlying MOE for the NN controller weight estimation is based on system-wide traffic data (i.e. data downstream from each traffic signal as well as upstream) *over a several-hour time period,*

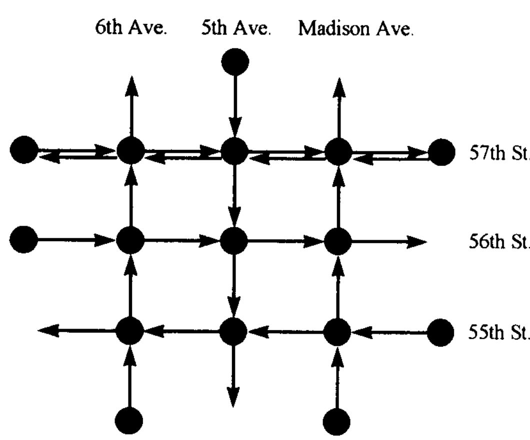

Fig. 2. Traffic simulation area (mid-Manhattan).

*The traffic queues were approximated from the assumed travel time, the upstream and downstream loop-counts, the downstream traffic signal phases, and the depletion process. Also, a queue represents the total number of cars on a road sector at each intersection without being further divided into lane counts.

the effect of signal settings, turning movements, etc. on the future accumulation of traffic at internal queues is factored into the formation of the controller function. (This is an example of how a true system-wide solution would differ from a solution based on combining individual intersection, artery, or zoned solutions on a network-wide basis as done e.g. in SCOOT.)

5.3. Results

The results of our simulation study of the system-wide traffic control algorithm are presented in Fig. 3 (mean arrival rate into the network over the 90 day period does not change) and Fig. 4 (step increased mean arrival rates on day 10 for all artery points into network). The 'prior' fixed-time control assumed a green-time/total-cycle-time value of 0.55 for all signals along N–S arteries. This was in the specified range of prior strategies in-place in the Manhattan sector during the recording of actual data (Rathi, 1988). In order to show true learning effects (and not just random chance as from a single realization) the curves in Figs 3 and 4 are based on an average of 100 statistically independent simulations. Every third day for S-TRAC in both figures represented an optional 'evaluation day' (step 6 of implementation in Section 4) to demonstrate improved values of the MOE. However, only data from the other 60 'training days' were used in the SPSA algorithm; thus, the adaptive training period could have been reduced to 60 days.

In Fig. 3, S-TRAC resulted in a net improvement of approximately 10% relative to the fixed-strategy-controlled system. This reduction in total wait time represents a reasonably large saving with a relatively small investment, particularly for high traffic density sectors. In comparison, major construction changes to achieve a net improvement in traffic flow of 10% in a well-developed area, such as for the traffic system in mid-Manhattan, would be enormously expensive. The large drop on the first day follows from the introduction of real-time (demand-responsive) control (vs the initial fixed-time strategy). Confidence bonds around the indicated curves that captured 90% of the daily variation were $\pm 2.8\,h$ for the prior control strategy and $\pm 5.2\,h$ for S-TRAC. Note that these bounds do not overlap after the first day, indicating the significance of the improvement offered by S-TRAC.

In the step increase case, Fig. 4 shows a corresponding step increase in total system wait time under the fixed-time (prior) strategy. Under S-TRAC, a step increase also occurred in total system wait time on day 10, but the wait time continued to decrease without any transient behavior

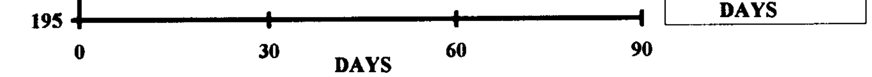

Fig. 3. System-wide mean wait time for 3:30 PM–7:30 PM period with constant mean arrival rates over 90 days.

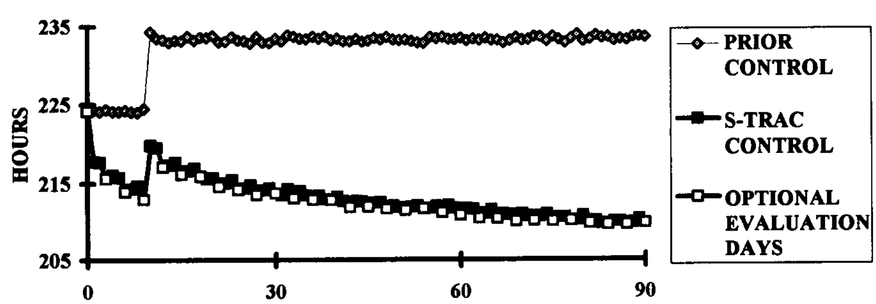

Fig. 4. System-wide mean wait time for 3:30 PM–7:30 PM period with increase in mean arrival rates on day 10.

subsequent to this phenomenon, and an approximate 11% improvement is evident after the 90-day test period.

6. CONCLUDING REMARKS

This paper has discussed S-TRAC for system-wide signal timing. It provides timings in response to instantaneous flow conditions while accounting for the inherent stochastic variations in traffic flow through a powerful stochastic optimization technique. The SPSA optimization technique (Spall, 1992) is critical to the feasibility of the approach since it efficiently provides the values of weight parameters in the neural network for control of signal timings in one of the periods within a 24 h time-frame. S-TRAC makes signal timing adjustments to accommodate to short-term conditions such as congestion, accidents, brief construction blockages, adverse weather, etc. Through SPSA, S-TRAC also has the ability to automatically accommodate to long-term system changes (such as seasonal traffic variations, new residences or businesses, long-term construction projects, etc.) without the cumbersome and expensive off-line remodeling process that has been customary in traffic control. The SPSA training process may be turned 'on' or 'off' as necessary to adapt to these long-term changes in a manner that would be essentially invisible to the drivers in the system.

A major issue in modern traffic control is practical implementation and maintainability. In practice, it has been found that most modern computer-based systems are not achieving their full potential as a result of inadequate understanding or commitment on the part of municipal authorities and the associated difficulties in implementation [see, e.g. DeSanto (1996)], which mentions that only two of 24 systems recently surveyed by the U. S. Department of Transportation were operating at their full capability. Approaches currently under development (e.g. OPAC) are even more complex than those currently implemented. On the other hand, S-TRAC avoids much of the complex modeling associated with other modern traffic control approaches (the main practical challenges in S-TRAC are the initialization of the search process and the choice of the NN structure for the controller). Further, S-TRAC may work with any existing sensor implementation provided there is some means of transmitting information between intersections and a central control facility; this contrasts with known model-based approaches (e.g. SCOOT) where additional sensors must be installed. Hence, S-TRAC has the potential to deliver real-time system-wide signal timings in a practically feasible manner.

Acknowledgements—The authors are grateful to Dr Richard H. Smith of the Johns Hopkins University, Applied Physics Laboratory for his help and knowledge of current traffic control systems. This work was supported by a JHU/APL Independent Research and Development Grant and U.S. Navy Contract N00039-95-C-0002.

REFERENCES

Chin, D. C. (1993) Performance of several stochastic approximation algorithms in the multivariate Kiefer-Wolfowitz setting. *Proc. of the 25th Symp. on the Interface: Computing Science and Statistics*, pp. 289–295.

Chin, D. C. (1997) Comparative study of several stochastic approximation algorithms for system optimization based on gradient approximations. *IEEE Transactions on Systems, Man, and Cybernetics* **27**, 244–249.

Chin, D. C. and Smith, R. H. (1994) A traffic simulation for mid-manhattan with model-free adaptive signal control. *Proc. of the 1994 Summer Computer Simulation Conf.*, pp. 296–301. San Diego, CA.

Dell'Olmo, P. and Mirchandani, P. (1995) An approach for real-time coordination of traffic flows on networks. Transportation Research Board Annual Meeting, Paper no. 950837. Washington, DC.

DeSanto, R. (1996) Operations and Maintenance of Traffic Control Systems. Technical report: RDS Assoc., Rocky Hill, CT 06067, U.S.A.

Funahashi, K. I. (1989) On the approximate realization of continuous mapping by neural networks. *Neural Nets.* **2**, 183–192.

Gartner, N. H., Tarnoff, P. J. and Andrews, C. M. (1991) Evaluation of optimized policies for adaptive control strategy. *Transportation Research Record 1324*, pp. 105–114.

Hunt, P. B., Robertson, D. I., Bretherton, R. D. and Winton, R. I. (1981) SCOOT—a traffic responsive method of coordinating signals. Transport and Road Research Lab., Crowthorne, U.K.

Kelsey, R. L. and Bisset, K. R. (1993) Simulation of traffic flow and control using fuzzy and conventional methods. *Fuzzy Logic and Control*, eds M. Jamshidi *et al.*, Ch. 12. Prentice Hall, Englewood Cliffs, NJ.

Martin, P. J. and Hockaday, S. L. (1995) SCOOT—an update. *ITE Journal*, pp. 44–48.

Messmer, A. and Papageorgiou, M. (1994) Automatic control methods applied to freeway network traffic. *Automatica* **30**, 691–702.

Nataksuji, T. and Kaku, T. (1991) Development of a self-organizing traffic control system using neural network models. *Transportation Research Record 1324*, pp. 137–145.

Newell, G. F. (1989) *Theory of Highway Traffic Signals.* Institute of Transportation Studies, University of California, Berkeley.

Papageorgiou, M. (1990) Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research-B* **24B**, 471–495.

Papageorgiou, M., Messmer, A., Azema, J. and Drewauz, D. (1995) A neural network approach to freeway network traffic control. *Control Engineering Practice* **3**, 1719–1726.

Rathi, A. K. (1988) A control scheme for high traffic density sectors. *Transportation Research-B* **22B**, 88–101.

Ritchie, S. G. (1990) A knowledge-based decision support architecture for advanced traffic management. *Transportation Research-A* **24A**, 27–37.

Rowe, E. (1991) The Los Angeles automatic traffic surveillance and control system. *IEEE Transactions on Vehicular Tech.* **40**, 16–20.

Santiago, A. J. and Smith, S. E. (1991) Evaluation of the highway capacity manual procedure for signal design. *ITE 1991 Compendium of Technical Papers*, pp. 239–243.

Spall, J. C. (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control* **37**, 332–341.

Spall, J. C. and Cristion, J. A. (1994) Nonlinear adaptive control using neural networks: estimation with a smoothed simultaneous perturbation gradient approximation. *Statistica Sinica* **4**, 1–27.

Spall, J. C. and Cristion, J. A. (1995) Model-free control of nonlinear stochastic systems in discrete time. *Proc. of the IEEE Conf. on Decision and Control*, pp. 2199–2204.

Spall, J. C. and Cristion, J. A. (1997) A neural network controller for systems with unmodeled dynamics with applications to wastewater treatment. *IEEE Transactions on Systems, Man, and Cybemetics-B* **27**, 369–375.

U.S. Dept. of Transportation (1991) *Transyt-7F User's Guide* (Methodology for Optimizing Signal Timing, MOST Vol. 4).

U.S. Dept. of Transportation (1995) *Intelligent Transportation System (ITS) Projects*, Publication no. FHWA-JPO-95-001.