Accounting for Real-World Uncertainty in Real-Time Adaptive Traffic Control

Xiao-Feng Xie, G. Barlow, S. Smith, and Z. Rubinstein
The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
Email: xfxie@alumni.cmu.edu

Technical Report
August 1, 2012

Abstract – Advanced traffic signal control systems require high-frequency predictions of traffic flows for effective real-time operations. However, in urban road networks, various sources of uncertainty can significantly degrade the accuracy of short-term flow predictions. In this paper, we study how to strengthen *schedule-driven traffic control*, a decentralized approach to real-time traffic control that has been shown capable of achieving scalable traffic network optimization in simulation, to enable its effective and stable operation in the dynamic real world. Strengthening strategies are proposed to mitigate several types of uncertainties in urban environments, based on a set of features extracted from limited vehicle detection information. Their effectiveness is then demonstrated in the context of a pilot implementation.

INTRODUCTION

Traffic congestion results in substantial costs for drivers and negative impacts on environmental conditions (1). As an important topic in implementing intelligent transportation systems (ITS), it is generally recognized that adaptive traffic control systems (ATCS) that adjust signal timing sequences based on the current traffic conditions offers the biggest payoff for reducing congestion.

Various ATCSs have been proposed in past years (2, 3). Some of them are based on *parametric optimization* that adjusts three major types of signal timings, i.e., *cycle lengths*, *green splits*, and *offsets*. Typical examples are SCOOT, SCATS, and ACS-Lite (4). Basically, each intersection operates through its sequence of phases with calculated green splits in a cycle that is offset from those of its neighbors. These ATCSs rely heavily on historical average volumes as the prediction of traffic flows. Thus, it is difficult for these methods to exploit the significance of high-frequency (e.g., second-by-second) predictions of actual traffic flows (5).

Many other ATCSs use *model-based control* (including model predictive control) methods. Typical examples are OPAC (6), CRONOS (7), RHODES (8), and some others (9, 10, 11). For intersection control, these methods proceed according to a rolling horizon and attempt to produce a good signal sequence over a specified planning horizon.The basic challenge for existing methods is computational efficiency, as they are not real-time feasible for realistic planning horizons (2) and are often forced to plan at coarse time resolutions (e.g., 5 seconds) or less effective approximations, due to the inefficiency of searching in an exponential planning search space.

Schedule-driven traffic control (12, 13) is an integration of traffic flow theory and artificial intelligence (AI) research that deals with multi-actor interactions, to address the challenge of achieving scalable traffic network optimization using high-resolution flow data and taking advantage of rapid online planning. At its core is a SCHedule-driven approach to Intersection Control (SchIC) (12) that is applied in a totally decentralized manner. Equipped with compatible interfaces to the inputs and outputs assumed by traditional model-based control approaches, SchIC defines a *scheduling search space* that enables efficient computation of near-optimal solutions, and it has been shown to outperform other state-of-the-art approaches (e.g., COP (14)) on optimality and efficiency. In a road network, the local scheduler of each intersection operates asynchronously and requests scheduled output flows from its direct upstream neighbors to increase its visibility of future incoming traffic from direct and indirect neighbors. If necessary, additional coordination mechanisms are applied to adjust the local schedules of each intersection to compensate for mis-coordination with its neighbors.

However, to be effective in practice, a real-time traffic control method requires quite accurate knowledge of traffic flows (3, 5). Given that prediction of local traffic flows must be accomplished with a limited number of sensors (e.g., induction loops and video detection) (5, 15, 16), various uncertainty issues can degrade the accuracy of flow prediction in real-world urban traffic control. The quality of vehicle detectors might be highly dependent on proper installation. For video detection, the performance is also influenced by dynamic environmental factors, e.g., weather, lighting, and traffic conditions (17). Mis-counting or over-counting might also be caused by arbitrary lane-changing behavior of human drivers. In a urban environment, passenger cars, trucks, and buses share the right of way. Temporary (partial) lane blockages might be caused by turning trucks, stopping buses, illegal parking, and on-street parking process, etc. Some roads might have side streets to contribute hidden flows that are not covered by any detectors. In general, the flow prediction process contains a predictable part and an unpredictable part (5), and in the real world, these sources of uncertainty can make the unpredictable part very significant. These

nontrivial perturbations, however, are often ignored in existing work (5, 12, 13, 18).

Recently, the decentralized, schedule-driven traffic control approach of (12, 13) has been incorporated into the SURTRAC system (Scalable Urban TRAffic Control) for pilot testing on a nine-intersection road network in the East Liberty area of Pittsburgh, PA. One basic challenge is coping with the extent and diversity of real-world uncertainties that can be found in this urban road network. In this paper, we propose several strengthening strategies aimed at achieving effective and stable operation in the face of these uncertainties, and evaluate their effectiveness. Before discussing these strategies, we first summarize our basic schedule-driven approach to real-time traffic control in a road network with peer-to-peer communication between neighboring intersections.

SCHEDULE-DRIVEN TRAFFIC CONTROL

Schedule-driven traffic control (12, 13) is an integration of traffic flow theory and artificial intelligence (AI) research in decentralized multi-agent planning and coordination. As shown in Figure 1, schedule-driven traffic control can be described in terms of intersection and network layers.

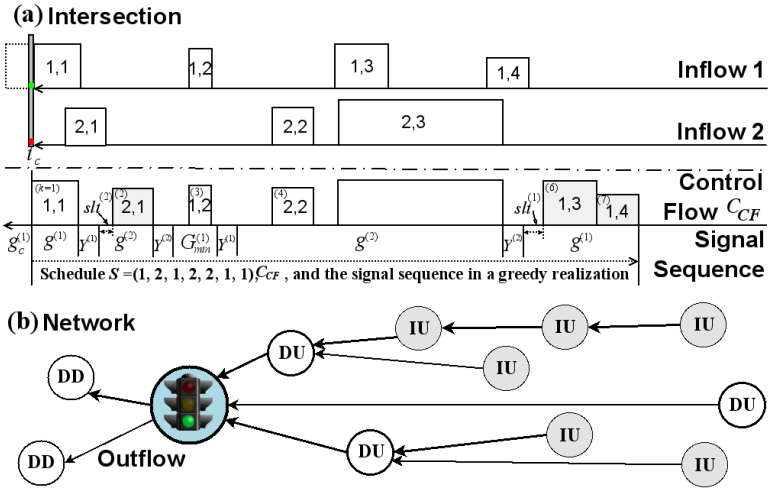

FIGURE 1 Schedule-driven traffic signal control: (a) Intersection level (Scheduling search space); (b) Network level (Non-local impacts for an intersection: DD = direct downstream; DU = direct upstream; IU = indirect upstream).

We focus on a road network of signalized intersections. For each intersection, the traffic light cycles through a set of *phases* I, and each phase $i \in I$ governs the right of way for a set of compatible movements from entry to exit *links* (roads). For traffic signal control, a *signal sequence* (*SS*) contains a sequence of phases and associated durations, where basic safety and fairness can be guaranteed using some *signal timing constraints*: For a pre-specified phase sequence, the yellow light after each phase i runs for a fixed duration (Y_i), while each phase i has a variable duration (g_i) that can range between a minimum (G_i^{min}) and maximum (G_i^{max}).

Each intersection is controlled by a local *scheduler* that holds a private signal sequence SS_{TL} for a finite future time, and makes decisions to extend SS_{TL} according to a *rolling horizon*. The system objective is to minimize the total delay of vehicles traveling through the network.

The intersection layer (Fig. 1a) is used for considering the significance of short-term (e.g., second-by-second) variability of traffic by searching in a prediction horizon H. To attain both the efficiency and optimality, a schedule-driven intersection control (SchIC) strategy is used (12).

The vehicles in a given traffic flow are characterized as a *job sequence* $C = (c_1, \cdots, c_{|C|})$, where $|C|$ is the number of jobs in C. Each *job* c is defined according to the number of vehicles in c and the expected arrival (departure) time at the intersection respectively for the first (last) vehicle in c. Two simple techniques, i.e., anticipated queue and threshold-based clustering (12, 19), are used to aggregate queuing and arriving vehicles.

At each decision time, the vehicles in the current prediction horizon H are organized into a set of *inflows* $(C_{IF,1}, \cdots, C_{IF,|I|})$, where each $|C_{IF,i}|$ is a job sequence that has the right of way in a given phase $i \in I$. In the *scheduling search space*, each full *schedule* contains a sequence of all jobs that pass the intersection at the shortest time, subjected to the given timing constraints. The corresponding *control flow* (C_{CF}) contains the results of applying a signal sequence (which can be extracted from each schedule) that clears all jobs in the inflows. The objective is to find a schedule of jobs with the minimal delay.

A forward recursion, dynamic programming process is then used to efficiently solve the scheduling problem. From a constructive view, the state space can be organized as a decision tree: Each schedule is built from the root node, and a job is added at each stage. At the same depth in the tree, states are grouped if they have the same jobs (with different orders) and the same last job (referring to the same last phase). A greedy state elimination strategy is then applied in each group, where only the state reached with the minimum delay is kept while all other states are eliminated. Thus, most branches are pruned in the early stages. The total process has at most $|I|^2 \cdot \prod_{i=1}^{|I|} (|C_{IF,i}| + 1)$ state updates, where $|C_{IF,i}| \leq H$ is the number of jobs in the ith inflow, and each state update can be executed in constant time. Thus, the time complexity is polynomial in the prediction horizon H since $|I|$ is limited for each intersection in the real world. SchIC can utilize high-resolution flow data in a long prediction horizon (12).

If an intersection has sufficiently long views, SchIC can efficiently achieve near optimal solutions. However, when operating in a larger road network, the local scheduler might be susceptible to myopic decisions that look good locally but not globally, and thus must incorporate non-local impacts from their neighbors to extend the visibility.

For each intersection, the network layer (Fig. 1b) is used for incorporating non-local impacts. For scalability, the communication is limited to direct neighbors.

The basic protocol is that each intersection only send out its scheduled *outflows* to its downstream intersections. For each intersection, each schedule describes a control flow, i.e., the results of applying the corresponding signal sequence that clears all jobs in the current inflows. With the auxiliary information of a road-ratio function (13) of the inflows and turning movement proportions at the intersection, the outflows to the exit roads can be obtained. Intuitively, for an intersection, the outflows of upstream intersections can form optimistic non-local inflows. The joint local and non-local inflows essentially increase the look-ahead horizon for an intersection. Due to the chain effect, a sufficiently long horizon extension can incorporate non-local impacts from indirect upstream neighbors. The optimistic assumption that is made is that direct and indirect neighbors are trying to follow their schedules. Normally, the optimization capability of SchIC results in schedules that are quite stable, given enough jobs in the local observation and large jobs (platoons) in the local and non-local observation. It is also the case that minor changes in the schedules of neighbors can often be absorbed, if there are sufficient slack time between successive jobs.

In the original SchIC, maximum green constraints are no included in the optimization process. This simplification does not present a problem for an isolated intersection since this will be

repaired during the execution. However, if operating in a network, such repairs might cause disruptive changes due to the difference between planned and actual outflows. To ensure the feasibility, the schedule is iteratively checked from the start time (13): If there is a violation time point, a cut is performed on the job at the time point, and all jobs after the time point are rescheduled by SchIC.

Additionally, coordination mechanisms are included for handling nontrivial mis-coordinated situations in the network. One common situation is the spillover that can occur due to insufficient capacity on an road that will block traffic flow from its upstream neighbor. Severe spillover that starts from some links can spread to a large area of a road network and eventually lead to "grid lock". For each intersection, a *spillover prevention mechanism* (13) is used to prevent a spillover to its upstream neighbor in the next phase by deciding if the current phase should be terminated earlier than planned, after evaluating the impact difference between the expected residual queue in the current phase and the expected spillover vehicles in the next phase. If conditions warrant, the downstream intersection sacrifices its own interest for the sake of its upstream neighbors.

SHORT-TERM TRAFFIC FLOW PREDICTION

For each intersection, an accurate view of the local inflows is essential to construct the scheduling search space, since the non-local inflows are communicated from its neighbors.

Ideally, the local inflows can be directly obtained if the phase flow profiles are known, as assumed in some exiting algorithms, e.g., COP (14). In a more practical setting where the requesting phases of vehicles are not completely known, the local flows can be obtained through a road-to-phase (RtoP) mapping (13), given the local *link flow profiles* and *turning movement proportions* at intersection, where the latter can be estimated using classical models (8, 20).

Each local link flow profile describes the state of queuing and arriving vehicles in a high-resolution prediction horizon (15). In practice, the main challenge is to achieve accurate short-term estimation through limited detection available in the real world (5, 16).

Figure 2 shows the placement of detectors in a typical installation. Detector groups are used for reporting two fundamental measures: traffic counts (passage) and occupancy time (presence) of vehicles. For each entry link, a group of stop-bar detectors is placed near the intersection, and a group of advance detectors is placed sufficiently far away from the intersection. For each exit link, a group of exit detectors is placed near the intersection. In a network, the advance detectors are normally adopted from the exit detectors of the upstream intersection, if they exist.

These detectors are compatible with the usages in existing traffic control systems (3).

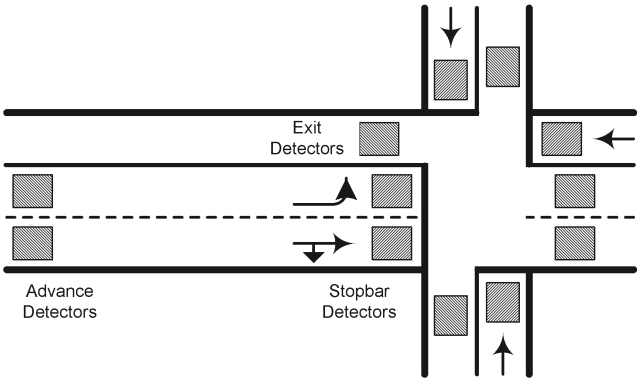

FIGURE 2 The placement of detectors in a typical installation.

Stop-bar detectors are used in vehicle-actuated operations, ACS-Lite, and SCATS. Far-side advance detectors (upstream exit detectors) are used by SCOOT, OPAC, ACS-Lite, and optionally by RHODES.

For video detection, all of these detectors can be defined with no additional cost than would be required to support the detection requirement for vehicle-actuated logic and other traffic-responsive control techniques. For simplicity, we will not consider special cases where some detectors might be removed (e.g., some exit counts might be estimated from stop-bar counts if the corresponding turning proportions are known for some intersection types).

A short-term flow prediction normally combines sensed data and basic traffic models (5, 15, 18). On each road, all vehicles are moving at constant free-flow speed (v_f) when they are not stopped. On each lane, a queue of vehicles is discharged in a green phase at the *saturation headway* (*shw*) after the start-up lost time. These model parameters can be easily estimated from historical flow data (8, 15).

The basic algorithm for obtaining a link flow profile is an input-output technique (15). The predicted arrival time of each vehicle is shifted by the horizon L/v_f, where L is the location of the advance detectors. From time t_0 to t_1, the queue q is changed by the difference of the number of predicted arrivals a and the number of departures d (8):

$$q(t_1) = q(t_0) + \sum_{t=t_0}^{t_1} a(t) - \sum_{t=t_0}^{t_1} d(t). \tag{1}$$

This technique cannot detect queues beyond the advance detectors (5, 18, 21). In theory, this does not cause a problem in schedule-driven traffic control, since non-local inflows can be communicated from upstream neighbors. For boundary nodes (i.e., intersections without any upstream neighbors), longer queues might be estimated using shockwave theory (18). However, as real-world uncertainties arise, this might not be effective since hidden information in a link flow profile can be highly distorted from the actual.

STRENGTHENING IN THE REAL WORLD

It is natural to combine data from multiple sensors and prediction models to achieve improved accuracies (22). In our context, we first extract *features* in current data, and then apply strengthening strategies for special situations that are identified according to (combined) features.

Features

Features service as interfaces between raw information and robust control techniques, since different features can be used for identifying traffic flow states. In RHODES (8), stop-bar presence information is used to confirm if there is a queue. In (23), phase status, detector occupancy, flow rate, and elapsed green time are considered as input features for classifying three basic stop-bar states. In (15), the difference between numbers of vehicle arrivals and departures is used for adjusting the queue, and a threshold headway is used for obtaining the queue clearance state. In (18), the occupancy data at the advance detectors are used for identifying changes in traffic states.

For each entry link n, we have the link flow rates fr_n^{arr} and fr_n^{dep} respectively from the groups of advance and stop-bar detectors. We also obtain the flow rates $fr_{n,i}$ from the *lane group* of the stop-bar detectors for each phase i. Each average flow rate is updated every 300 seconds.

For each entry link n, the *arrival/departure ratio* is

$$ADRatio_n = fr_n^{arr} / fr_n^{dep}. \tag{2}$$

For phase i, the *flow ratio for the critical lane group* is

$$clg_i = \max_n(fr_{n,i} \cdot shw/l_{n,i}), \tag{3}$$

where $l_{n,i}$ is the number of involved lanes on the entry link n that are serviced in phase i.

The *exit* and *advance spillover* features SO_m^{exit} and SO_n^{adv} at each exit link m and entry link n are respectively identified if the occupancy ratio on any exit and advance detectors is larger than 80% for t_{SO} seconds. For each entry link n in phase i, the *queue stalling* and *queue clearance* features $QS_{n,i}$ and $QC_{n,i}$ are respectively obtained if the occupancy ratios of all stop-bar detectors in the corresponding lane group are 80% for t_{QS} seconds and 0% for t_{QC} seconds. Here By default, $t_{SO} = t_{QS} = 3 \cdot shw$, and $t_{QC} = 2 \cdot shw$.

Strengthening Strategies

For schedule-driven traffic control, strengthening strategies are proposed for handling real-world situations, based on the features extracted from detection information.

The basic queue length calibration is realized as follows. For each entry link, the predicted queue size q is a typical hidden state. A basic adjustment is $q = \max(0, q)$, and $q = \min(N_{SC}, q)$, where N_{SC} is the link storage capacity (15). The queue size is also adjusted from the measured features: $q = N_{SC}$ if SO^{adv} and t_{QS} are identified, as the link is not serviced; and $q = 0$ if QC is obtained, and the link is serviced. The adjustment by spillover detection is especially important for video detection under a high traffic volumes, where the count accuracy may deteriorate substantially due to the difficulty in sensing gaps between vehicles.

Some properties can be obtained from these basic adjustments. If q is over-estimated, an ideal full clearance will adjust q to 0. For each intersection, it costs extra t_{QC} seconds. For its neighbors, the estimation error imposes some uncertainty on weights and execution durations of jobs in non-local inflows, and might cause some disturbance of the coordination between intersections. From the viewpoint of robust scheduling, any over-estimation might be considered as a buffer insertion (24), and SchIC with the rolling horizon scheme is essentially doing on-line rescheduling that can effectively respond to such unexpected dynamics (25) and provide good stability guarantees (26). If q is under-estimated, significant delay might occur from residue queues, and these residue queues are not seen in the following one or more cycles. The situation might be significantly worse if the queue starts to spill back to upstream intersections. Thus under-estimation should normally be avoided, although arriving vehicles in the look-ahead horizon might alleviate most negative impacts on pure queue clearance, if the current phase might be further extended according to arriving platoons in the prediction horizon.

The *arrival-adjusting* strategy is used to take account of the detection inaccuracy. As in (15), we assume that the group of stop-bar detectors can yield a quite accurate estimation of departure vehicles. If $ADRatio < 1$, some arriving vehicles are missed, and the numbers of queuing and arriving vehicles are under-estimated. Thus, when vehicles are detected at the advance detectors, the count is divided by $ADRatio$ to reclaim those missing vehicles in the link arrival profile.

The *queue clearance management* strategy contains two parts: "elastic" and "tolerance". The "elastic" part concerns the queue clearing time t_{QC}. If t_{QC} is too small, a queue might be truncated. If t_{QC} is too large, green time is regularly wasted. Thus, for $t_{QC} = r_{QC}^{ela} \cdot shw$, the elastic ratio r_{QC}^{ela} is defined as a sigmoid function on the queue size q:

$$r_{QC}^{ela} = r_L + (r_U - r_L)/(1 + \exp(-1.5 \cdot q/l - 6), \tag{4}$$

where r_L and r_U are lower and upper bounds. By default, $r_L = 1.5$ and $r_U = 3.5$. A long queue size will have a large r_{QC}^{ela}, and thus is hardly be truncated.

Due to some real-world uncertainty, such as mid-road bus stopping, on-street parking, or stop-bar mis-counting, a long queue might still be unexpectedly truncated and become a residue queue. The "tolerance" part of *queue clearance management* is applied to avoid under-estimation. The current queue is stored as q', and is derived using Eq. 1 in the same way as q. Then q' is retrieved as q in the following N_{TOL} cycles, where $N_{TOL} = 1$ is the default tolerance size.

In a urban road network, spillover happens on some links from time to time. The *spillover reaction* strategy is then applied. For an intersection at phase i, the spillover status might be identified from $QS_{n,i}$ on some entry links and SO^{exit} on some exit links. For the current phase, the possible turning movements are identified. If the total flow proportion toward these exit roads exceeds a threshold r_{SO}, the signal will immediately switch to the next phase, and SchIC is applied to recalculated the new schedule. By default, r_{SO}=50% for having a reasonable reaction.

The *major-flow management* strategy determines if the major phase should be actually terminated after SchIC makes the decision. The major phase is defined as the phase $\hat{i}$ that services the maximum flow $\sum_n fr_{n,\hat{i}}$. The main reason is to mitigate unexpected disturbance from pedestrian calls, which are unknown to the current scheduler. For a minor phase, an unexpected pedestrian call might significantly increase the minimal green time (e.g., 20 seconds), which might lead to nontrivial inefficiency. Basically, there is a significant waste of green time allocation for the minor phase. Furthermore, the traffic signal cannot switch back to the major phase on the scheduled time. For an intersection with short entry links, this disturbance might cause spillover, and might interrupt the scheduled queue clearance process by triggering the spillover reaction strategy of its upstream intersections. In this strengthening strategy, the major phase will be continued, if each entry link that are serviced in the other phases has the queue size lower than q_{TH}. For responding to the current major flow, q_{TH} is defined as

$$q_{TH} = \text{ceil}(clg_{\hat{i}}/0.05) * l_{\hat{i}}. \tag{5}$$

For example, for the main phase $\hat{i}$, if the critical lane group has 2 lanes, and the flow ratio $clg_{\hat{i}}$ is 0.15, then $q_{TH} = 6$.

EXPERIMENTS

The performance of strengthening strategies for schedule-driven intersection control was evaluated on a nine-intersection road network in the East Liberty neighborhood of Pittsburgh, PA, as shown in Fig. 3. Although the total scale is not large, this road network has some interesting characteristics. First, instead of a single arterial (as studied in most traditional systems (27, 28)), this network contains a triangle, where three major streets cross (Penn Circle, Penn Avenue, and Highland Avenue), with changing traffic flows throughout the day. Second, the road lengths are quite short (ranging from 40.5 to 174.1 meters, as considering the speed limit is 11.18 meters/second), which impose a nontrivial challenge for achieving effective coordination in a totally decentralized traffic control system. Third, there are various uncertainty elements in an urban environment, e.g., bus lines (on Penn Avenue and Highland Avenue), on-street parking (on all three major streets), hidden flows from/to mid-block side streets (Baum Boulevard for Highland Avenue, Sheridan Avenue for Penn Avenue and Penn Circle), and pedestrian calls (all intersections except for the intersection of Penn Avenue and Highland Avenue). Finally, video detection is currently installed for sensing

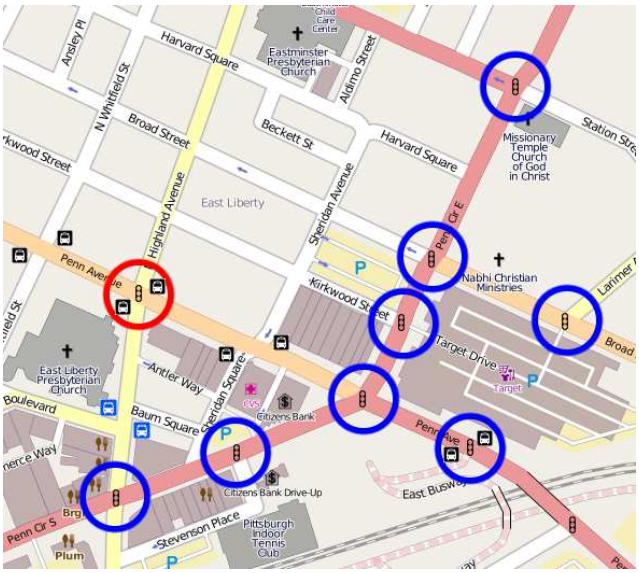

FIGURE 3 The nine-intersection pilot test site in the East Liberty neighborhood of Pittsburgh, PA.

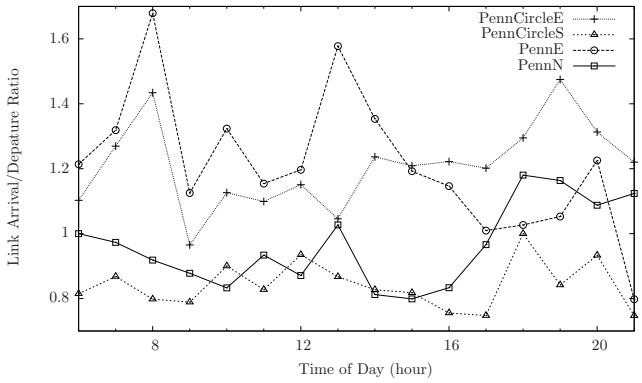

FIGURE 4 Hourly link arrival/departure ratios at the intersection of Penn Avenue and Penn Circle.

vehicles. Fig. 4 gives the actual hourly *ADRatio* data in a day (from 6:00am to 22:00pm, Jul-2-2012) for all entry links of the intersection of Penn Avenue and Penn Circle. This shows that *DARatio* varied in a large range, where the fluctuations indicate a nontrivial unpredictable part.

The evaluation results are reported on simulation and actual pilot tests. We use a microscopic road traffic simulator, Simulation of Urban Mobility (SUMO)[1]. Simulation is useful to demonstrate trends for different parameter settings, but it is difficult to reproduce some kinds of real-world uncertainties in SUMO. For pilot tests, the schedule-driven traffic control approach has been integrated into SURTRAC (Scalable Urban TRAffic Control), which is currently running in the road network. The travel information can be recorded using the Global Positioning System (GPS) through floating vehicle runs. Pilot tests are more expensive than simulation.

For main parameters, the saturation headway is $shw = 2$ seconds/vehicle, the detection

[1]http://sumo.sourceforge.net

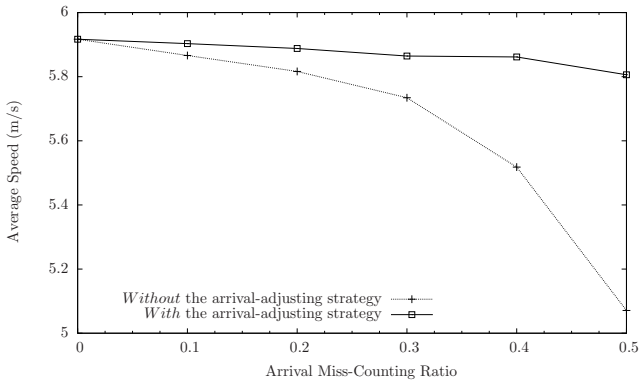

FIGURE 5 Before and after results for the arrival-adjusting strategy.

TABLE 1 Pilot test results during the PM rush period

	BASELINE	SURTRAC0		SURTRAC1	
Speed (m/s)	3.53	3.70	4.98%	4.48	26.94%
Travel Time (s)	144.38	140.66	2.58%	112.56	22.04%
Wait Time (s)	85.87	82.19	4.29%	55.65	35.19%

frequency is one second, and the minimal phase extension time is one second.

Fig. 5 gives the average vehicle speed results in simulation with and without the arrival-adjusting strategy on different mis-counting ratios at the advance detectors, using the flow data in the AM rush period. For each instance, we calculate the mean of 100 independent runs. Without the arrival-adjusting strategy, the performance drops significant for increasing ratios of missing counts. With the strengthening strategy, the performance can be restored to near optimal.

In the pilot tests, we compared three systems during the PM rush period, which is the most congested time of day in this region. BASELINE is the original coordinated-actuated signal control with timings optimized by Synchro 7.0 (a state-of-practice package based on traditional offset calculation). SURTRAC0 and SURTRAC1 are two versions of our system; the former does not include the queue clearance and major-flow management strategies.

All tests were performed between 5pm and 6pm. Each system was evaluated in three runs, where each run contains twelve dominant routes. The relative volumes along different routes were used to determine weights. BASELINE and SURTRAC runs were conducted in March and June of 2012, respectively. An analysis of traffic data shows that volumes were essentially the same, though slightly higher in June.

Table 1 gives the average performance of pilot test results and the percentage improvements over BASELINE. SURTRAC0 achieved better results than BASELINE, and SURTRAC1 achieved much better results. The improvements clearly demonstrate the advantage and necessity of using strengthening strategies for schedule-driven traffic control, which might significantly enhance the progress of large platoons crossing intersections in the road network.

CONCLUSIONS

In this paper, we have presented some strengthening strategies for schedule-driven traffic control. These strategies are aimed at handling real-world uncertainties, based on the features extracted from detection information. The effectiveness of these strategies was evaluated in simulations and

in the context of a real-world pilot test. The overall results demonstrate that use of these techniques does in fact enable effective real-time adaptive traffic control in urban road networks and indicates the potential of decentralized, schedule-driven traffic control. Given the decentralized nature of the approach, it is inherently scalable and expansion to incorporate additional intersections is straightforward.

There are several aspects of the current system that warrant further study. First, the system might gain more precise control given knowledge about bus arrivals and pedestrian calls. Second, multi-sensor data fusion techniques, e.g., Kalman filter and Dempster-Shafer theory of evidence, might be integrated for achieving more accurate flow prediction.

REFERENCES

[1] Schrank, D., T. Lomax, and S. Turner. *Annual Urban Mobility Report*. Tech. rep., Texas Transportation Institute, Texas A&M University System, TX, 2011.

[2] Papageorgiou, M., C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, Vol. 91, No. 12, 2003, pp. 2043–2067.

[3] Stevanovic, A. *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*. Tech. Rep. Synthesis 403, NCHRP, 2010.

[4] Luyanda, F., D. Gettman, L. Head, S. Shelby, D. Bullock, and P. Mirchandani. ACS-Lite algorithmic architecture: Applying adaptive control system technology to closed-loop traffic signal control systems. *Transportation Research Record*, Vol. 1856, 2003, pp. 175–184.

[5] Head, K. L. Event-based short-term traffic flow prediction model. *Transportation Research Record*, Vol. 1510, 1995, pp. 45–52.

[6] Gartner, N., F. Pooran, and C. Andrews. Optimized policies for adaptive control strategy in real-time traffic adaptive control systems - Implementation and field testing. *Transportation Research Record*, Vol. 1811, 2002, pp. 148–156.

[7] Boillot, F., S. Midenet, and J. Pierrelee. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies*, Vol. 14, No. 1, 2006, pp. 18–38.

[8] Mirchandani, P. and L. Head. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, Vol. 9, No. 6, 2001, pp. 415–432.

[9] Porche, I. and S. Lafortune. Adaptive look-ahead optimization of traffic signals. *ITS Journal*, Vol. 4, No. 3-4, 1999, pp. 209–254.

[10] Prashanth, L. A. and S. Bhatnagar. Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 2, 2011, pp. 412–421.

[11] Lin, S., B. D. Schutter, Y. Xi, and H. Hellendoorn. Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C: Emerging Technologies*, Vol. 24, 2012, pp. 122–140.

[12] Xie, X.-F., S. F. Smith, L. Lu, and G. J. Barlow. Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies*, Vol. 24, 2012, pp. 168–189.

[13] Xie, X.-F., S. F. Smith, and G. J. Barlow. Schedule-driven coordination for real-time traffic network control. In *International Conference on Automated Planning and Scheduling (ICAPS)*, Sao Paulo, Brazil, 2012, pp. 323–331.

[14] Sen, S. and K. Head. Controlled optimization of phases at an intersection. *Transportation Science*, Vol. 31, No. 1, 1997, pp. 5–17.

[15] Sharma, A., D. Bullock, and J. Bonneson. Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. *Transportation Research Record*, Vol. 2035, 2007, pp. 69–80.

[16] Gentili, M. and P. Mirchandani. Locating sensors on traffic networks: Models, challenges and research opportunities. *Transportation Research Part C: Emerging Technologies*, Vol. 24, 2012, pp. 227–255.

[17] Rhodes, A., D. M. Bullock, J. Sturdevant, Z. Clark, and D. G. Candey. Evaluation of the Accuracy of Stop Bar Video Vehicle Detection at Signalized Intersections. *Transportation Research Record*, Vol. 1925, 2005, pp. 134Ð–145.

[18] Liu, H. X., X. Wu, W. Ma, and H. Hu. Real-time queue length estimation for congested signalized intersections. *Transportation Research Part C: Emerging Technologies*, Vol. 17, No. 4, 2009, pp. 412–427.

[19] Xie, X.-F., G. J. Barlow, S. F. Smith, and Z. B. Rubinstein. Platoon-based self-scheduling for real-time traffic signal control. In *IEEE International Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011, pp. 879–884.

[20] Lan, C. and G. Davis. Real-time estimation of turning movement proportions from partial counts on urban networks. *Transportation Research Part C: Emerging Technologies*, Vol. 7, No. 5, 1999, pp. 305–327.

[21] Cai, C., B. Hengst, G. Ye, E. Huang, Y. Wang, C. Aydos, and G. Geers. On the performance of adaptive traffic signal control. In *International Workshop on Computational Transportation Science*, Seattle, WA, 2009, pp. 37–42.

[22] Hall, D. L. and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, Vol. 85, No. 1, 1997, pp. 6–23.

[23] Perez-Montesinos, J., M. P. Dixon, and M. Kyte. Detection of stop bar traffic flow state. *Transportation Research Record*, Vol. 2259, 2011, pp. 132–140.

[24] Herroelen, W. and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, Vol. 165, No. 2, 2005, pp. 289–306.

[25] Hiatt, L. M., T. L. Zimmerman, S. F. Smith, and R. Simmons. Strengthening schedules through uncertainty analysis. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Pasadena, CA, 2009, pp. 175–180.

[26] Terekhov, D., T. T. Tran, D. G. Down, and J. C. Beck. Long-run stability in dynamic scheduling. In *International Conference on Automated Planning and Scheduling (ICAPS)*, Sao Paulo, Brazil, 2012, pp. 261–269.

[27] Day, C. M., T. M. Brennan, H. Premachandra, J. R. Sturdevant, and D. M. Bullock. Analysis of Peer Data on Intersections for Decisions About Coordination of Arterial Traffic Signal. *Transportation Research Record*, Vol. 2259, 2011, pp. 23–36.

[28] Liu, Y. and G.-L. Chang. An arterial signal optimization model for intersections experiencing queue spillback and lane blockage. *Transportation Research Part C: Emerging Technologies*, Vol. 19, No. 1, 2011, pp. 130–144.