Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL 2007)

An Approximate Dynamic Programming Strategy for Responsive Traffic Signal Control

Chen Cai*

Centre for Transport Studies, University College London, London, WC1E 6BT, United Kingdom

***Abstract*—This paper proposes an approximate dynamic programming strategy for responsive traffic signal control. It is the first attempt that optimizes signal control objective dynamically through adaptive approximation of value function. The proposed value function approximation is separable and exogenous factor independent. The algorithm updates the approximated value function progressively in operation, while preserving the structural property of the control problem. The convergence and performance of the algorithm have been tested in a range of experiments. It has been concluded that the new strategy is as good as the best existing control strategies while being efficient and simple in computation. It also has the potential of being extended to multi-phase signal control at isolate junction and to decentralized network operation.**

I. Introduction

Traffic signal governs road user at junctions in an increasingly congested urban traffic environment. The performance of traffic signals therefore largely determines the quality of travel within urban network, and its influence may well extend to other aspects of urban life. This challenging environment has made traffic signal control a testing ground for a variety of optimum control strategies. In this paper, for the first time in responsive traffic signal control at isolated junction, we develop an adaptive control strategy based on approximate dynamic programming (ADP) to provide efficiency in computation and effectiveness in operation. It will show that the control algorithm updates itself progressively in operation, and the performance is as good as the best existing control strategy. As ADP comes out of the latest development in adaptive control theory and reinforced learning, this work also provides an opportunity of bridging intelligent computing to the needs of urban traffic management.

Section II introduces the basic context of traffic signal and its control objectives, followed by the explanatory discussion on traffic signal control strategies. Section III is a general review of related ADP literature and value function approximation methods. The problem specification and formulation are in Section IV, with experiment design in Section V and consequent results in VI. Section VII summarizes this study.

* Tel: +44-020-7679-0467; fax: +44-020-7679-1567; E-mail address: c.cai@ucl.ac.uk (C.Cai).

II. Traffic Signal Control

A. Traffic Signal

Traffic signals are used to manage conflicting requirements for the use of road space by allocating right of way to different sets of mutually compatible traffic movements during distinct time intervals (see Ref. [1]). The objective of signal control will vary in accordance with the prevailing policy of urban traffic management and control. Objectives may include minimizing delays to road users, or reducing vehicle emissions, or improving safety, or providing priority for public usages, or a practical combination of those. The resources that are available for optimization are data from empirical data sets or online information from detectors, and optimization procedures to make use of data to calculate an appropriate plan.

B. Control Strategies

A daily life experience with traffic signal may give the impression of green and red indications, or possibly plus the amber indication. However, to calculate an appropriate signal timing plan, a control strategy may encounter much more variables denoted by specific terminologies. We here define the necessary terminologies to facilitate the discussion that follows.

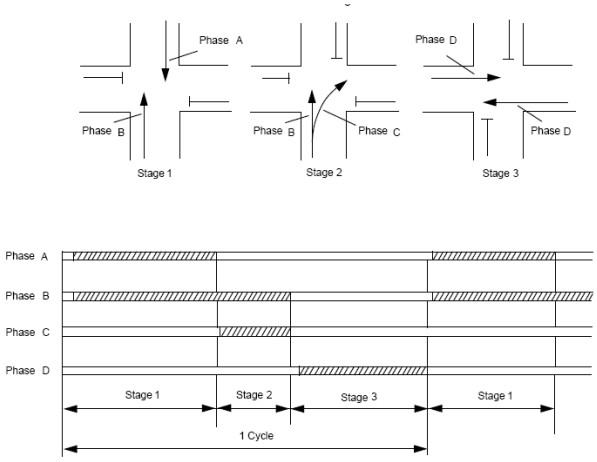

Fig.1. Phase, Stage and Cycle

Link: A group of adjacent lanes on which traffic forms a combined queue.

Phase: A group of one or more traffic or pedestrian links which always receive identical signal light indications.

Stage: A group of one or more traffic and/or pedestrian phases which receive a green signal during a particular period of the cycle.

Intergreen: The period between the end of the green display on one stage and the start of the green display on the next stage.

Cycle: Usually considered to be the time between successive starts of the stage 1 green.

The relationship between phase, stage, intergreen and cycle is illustrated in Fig.1.

The early generation of traffic signal control strategies operated within a fixed cycle (and often with a fixed stage order) and regardless of variation in approaching traffic. Strategies of this nature are referred as fix-time control. Ref. [2] provides pioneer work in this field with its fundamental establishment in delay function and cycle time setting. Ref. [3] optimizes stage-based fix-time control, followed by [4] with a phase-based optimization algorithm. Phase-based strategies yielded substantial benefits over stage-based ones, however, at the expenses of complexity in variables and constraints, and moreover, still not responsive to changing traffic.

Responsive signal control, or vehicle actuated (VA) control, arose in a context that fix-time control may operate unsatisfactorily under changing traffic, and in that if not updated manually, even optimum fix-time plan is aging over time. The pioneer research in responsive control theory is [5]. The first generation of successful responsive systems includes SCOOT (Ref. [6]) which gives responses to real-time traffic through optimizing cycle length, phase split and offset. Similar to SCOOT is SCATS (Ref. [7]). These systems began to outperform the best fix-time control strategies with 6%–20% savings in travel time at network level. On the other hand, MOVA (Microprocessor Optimized Vehicle Actuation, Ref. [8]) system has been successfully implemented at isolated junctions and becomes standard strategy of its kind in UK. Nevertheless, the above systems are all stage-based.

Not long after the establishment of initial responsive control strategies was backward dynamic programming (BDP) recognized as a complete solution. Ref. [9] uses BDP to deduce the analytical benchmark of optimum signal control performance, with a near optimum policy proposed thereafter. This works inspired the evolution of successful dynamic strategies such as OPAC (Ref. [10]) and PRODYN (Ref. [11]). OPAC uses a rolling horizon approach with optimum sequential constrained search (OSCO) algorithm. Also using rolling horizon approach, PRODYN employs a forward dynamic programming (FDP) to optimize performance, and the value function in FDP adopts the empirical formula in [9]. The dynamic feature of the two systems allows them to decentralize network control to local level, thus more flexible than centralized control strategies such as SCOOT and SCATS.

The BDP itself, though powerful in analytical research, has limited rule in dynamic signal control. As in many other fields, it nevertheless under the curse of dimensionality: large state space (more links, phases, and etc.), large outcome space, and large action space. Not only that, in traffic control, BDP requires the knowledge of traffic information for the whole planning period, which is quite impractical. The existing dynamic strategies either use exhaustive search with anticipated traffic information (OPAC) or FDP algorithm depending on empirical value function which is dependent on exogenous factor (traffic flow).

A more adaptive, intelligent and efficient dynamic traffic signal control strategy may be developed if the exhaustive search could be circumvented and the value function itself was adaptive to changing environment.

III. Approximate Dynamic Programming

Approximate dynamic programming (ADP) evolved in the field of dynamic programming to solve the problems that would have been computationally intractable via backward propagation. Given a single traffic junction with eight links and up to 20 vehicles per link, if we define the traffic state as the vehicles in the junction, we will have an excessive large number of states which is 21^8, not to mention that we actually have signal states and optional decisions to consider as well. With such a large dimension, dynamic programming with backward propagation could easily be inefficient or impractical for real time control, for it has to loop over all the states to find the optimal solution. ADP, however, is so designed to overcome the curse of dimensionality that it avoids evaluating through the whole state space by using functional approximations which only require the estimation of a few parameters to approximate the whole value function. The functional approximations may update the value estimates, at each iteration, using the updating function, which calculates discrete derivatives at a single state. ADP of this kind was investigated for multistage resource allocation problems in [12] and [13]. Based on the investigation of structural property, applications of ADP were further extended to stochastic batch service problems in [14] and [15]. Ref. [16] summarized approximation algorithms for discrete stochastic optimization and proposed a provable algorithm which is separable and piecewise linear. A comprehensive introduction to ADP, dimensionality, stepsize and functional approximation can be found in [17].

Assisted by the findings in ADP literature, especially in batch service problems which resembles a certain kind of similarity, we are able to formulate an ADP algorithm for our traffic signal control problem.

IV. Problem Definition

In the section we formulate the dynamic signal control problem and develop the ADP algorithm strategy according the problem specifications.

A. Assumptions

1) The signal control strategy is designed to be stage-based.
2) Time is divided into short intervals of 5 seconds each. Queue lengths are calculated at the end of each interval, neglecting the detail of vehicle behavior during the interval. Signals may only be switched at the boundary between intervals.
3) Approaching traffic is homogenous. The arriving rate of traffic is known at the beginning of each time interval for the next 10s (two intervals). Traffic rate within a single time interval may take an integer value of 0, 1 or 2 vehicles only. It is assumed that the information comes from detectors 100m upstream.
4) Signal phases are composed of effective greens and effective reds only, so that no amber interval needs be considered. The intergreen period, which follows a decision to change the signals from green on one stage to green on the other stage, is of 5 seconds duration. This is to say that the signals will be all red for one interval when a switching decision is made.
5) The saturation flow on all traffic links is 2 vehicles per interval. This is equivalent to 1440 vehicles per hour, a rate that is sufficiently close to the saturation flow of a single traffic lane.
6) There are no constraints on the minimum or maximum duration of a green period. The maximum queue length on a single link is restricted to 20 vehicles.

B. State Variable

A control state in dynamic traffic signal control is defined here as the total number of vehicle queuing in traffic links which receive identical signal indications. The total number of vehicles in queue receiving green signal is represented by the variable q_g, and those receiving red by q_r. These two variables together form the state variable S given by

$$S = \begin{bmatrix} q_g \\ q_r \end{bmatrix}.$$

State variable S can be seen as a product of two subordinate state variables — the queue state variable Q, and the signal state variable G. We define the subordinate state variables as the followings:

$$Q = \begin{bmatrix} q_1 \\ \vdots \\ q_i \end{bmatrix}, \text{ where } q_i \text{ is the queue length on link } i;$$

$$G = \begin{bmatrix} g_1 \\ \vdots \\ g_i \end{bmatrix}, \text{ where } g_i = \begin{cases} [1 \quad 0], \text{ if link receives green,} \\ [0 \quad 1], \text{ if link receives red.} \end{cases}$$

The relationship between the control state variable and its two subordinate variables is simply as:

$$S = G^T Q. \tag{1}$$

Based on this relationship and for convenience, we refer to S in the rest of this paper as primary state variable. The corresponding value function should therefore return the value of being in primary state S.

Another interesting but also critical issue about state variables is whether they are *incomplete*. Ref. [14] and [15] provide a thorough definition of incomplete state variables S^- and its distinction from complete one S. In short, $S_t = S_t^- + W_t$, where W_t is the information vector which becomes available at time t. This is to say that S contains the already arrived information, while S^- does not. In our case, all the state variables are *incomplete*. But for the simplicity in notations, we do not explicitly denote them as in [14] or [15].

C. System Dynamics

Arrival traffic information vector W and decision vector X are introduced as:

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_i \end{bmatrix},$$

and

$$X = \begin{bmatrix} 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \text{ for change signal , or } X = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \text{ for not change .}$$

Given that at time interval t, at state S_t, with arrival traffic information W_t, we make decision X_t. Thus the transfer function for subordinate state variable Q is expressed by:

$$Q_{t+1} = Q_t + W_t - O_t(Q_t, W_t, X_t), \tag{2}$$

where the O is the outflow vector,

$$O = \begin{cases} o_1 \\ \vdots \\ o_i \end{cases}, \text{ and } o_i = \begin{cases} 0 & \text{if on red or change signal} \\ 2 & \text{if on green and } q_i + w_i \geq 2 \\ q_i + w_i & \text{if on green and } q_i + w_i < 2 \end{cases}.$$

The transfer function for subordinate state variable G is given by:

$$G_{t+1} = (G_t + X_t)_{\text{mod } 2}. \tag{3}$$

The primary state is then transferred through (1):
$S_{t+1} = G_{t+1}^T Q_{t+1}$.

D. Delay Functions

Delay functions (or cost function in general terms) can be divided into two parts: the first part represents the one-step

delays, and the second part represents discount delays incurred the future. We define the one-step delay function as:

$$C_t(S_t, W_t, X_t) = \sum_{i=1}^{I} \left[Q_t + W_t - O_t(Q_t, W_t, X_t) \right]. \quad (4)$$

The second part is frequently referred as the value function. In BDP the value function returns directly the expected value (in probabilistic BDP) or exact value (in deterministic BDP) of being in a state. In ADP, however, we usually have to approximate the value function.

Ideally we would assume that the value function could represent the delays that are to be incurred to the vehicles in queue. Ref. [2] established one of the most fundamental formulas to estimate the average delay per vehicle at a signalized junction. Although this algorithm was widely adopted in deriving optimum fixed-time control, its application in dynamic control is limited in that it applies to fixed values of cycle time, green time split, degree of saturation and traffic flow, and moreover, does not associate the primary state variable *S*.

Ref. [9] proposes an approximate function to estimate the total additional delays in 10 minutes caused by non-zero initial queues in the traffic links. The additional delay *D* is expressed by:

$$D = \frac{0.2}{(1-Y)}(q_g + 1.3q_r)^2, \quad (5)$$

where

$$Y = \sum_{\text{Links}} \frac{\text{Arrival flow}}{\text{Saturation flow}}.$$

Equation (5) associates primary state variable *S* to the delay estimating function. It assigns a greater coefficient for queues on red than on green. The purpose of this is to define the structure of signal control problem—leaving the same amount of queue on red will cause more delay than leaving them on green. A near optimum strategy was proposed based on (5). The strategy determines whether to change the traffic signal or not based on the values that sum up the one-step delays and the additional delays from (5) caused by respective decisions. We refer this strategy here as RB (Robertson and Bretherton), and it will be a competing method in experiment to our ADP strategy later proposed in the paper. Strategy RB was later validated in [18] which found that the consequences of a non-zero initial queue persisted only for finite length of time, after which the optimum state sequences merged. In most cases merge happened within 10mins in the latter study.

Equation (5) presents a good basis to construct our approximate value function in ADP. However, it is clear that it has an inseparable quadratic form and also depends on the exogenous factor, i.e. variable *Y*. Further investigation through simulations also demonstrated considerable discrepancies between estimations from (5) and the true values. To apply those already proofed ADP algorithms, and to facilitate the possible introduction of Mote Carlo sampling techniques in cases of high dimensionality, we would prefer an approximate value function of separability as well as independence from exogenous factor. To make the value function also adaptive over time and reinforce the learning, we would further prefer adaptive variable coefficients. In those regards we propose an approximate value function for the signal control problem, which takes the form of:

$$\bar{V}(S) = S^T A = \alpha q_g + \beta q_r, \quad (6)$$

where the coefficient matrix *A* is given by

$$A = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

E. Objective Function

Our objective then is to minimize the total discounted delay within a time horizon of *T* intervals, which can be expressed as:

$$\min \left\{ \sum_{t=0}^{T} \gamma^t C_t(S_t, W_t, X_t) \right\}, \quad (7)$$

through recursively computing the optimality equation:

$$\begin{aligned} \tilde{V}_k(S_t) = \min_{X \in \mathrm{X}} \{ & C_t(S_t, W_t, X_t) + C_{t+1}\left(\left(S_{t+1} \middle| S_t, W_t, X_t \right), W_{t+1}, X_{t+1} \right) \\ & + \gamma \bar{V}_{k-1}\left(S_{t+2} \middle| S_{t+1}, W_{t+1}, X_{t+1} \right) \}, \end{aligned} \quad (8)$$

where *k* indicates the number of iterations.

F. Decision Rule

The decision rule governs the optimum decision to be made at each epoch of time. It first offers three optional decisions for the next 10s, i.e. next two time intervals, as such

$$\chi(S_t) = \begin{cases} 1^{st}\ Option: X_t = \text{ no change}, X_{t+1} = \text{no change} \\ 2^{nd}\ Option: X_t = \text{ change}, X_{t+1} = \text{ no change} \\ 3^{rd}\ Option: X_t = \text{ no change}, X_{t+1} = \text{change} \end{cases}.$$

The decision rule allows the change in signal indication at time *t* if and only if when 2nd option gives lower value than both 1st and 3rd option.

G. Adaptive Approximation

To approximate the value function adaptively through progress, we first obtain the new observation of each element of the coefficient matrix *A* by:

$$\hat{\alpha}^k \simeq \frac{\partial \bar{V}}{\partial q_g} = \frac{\tilde{V}(S_t + e_g) - \tilde{V}(S_t)}{\Delta q_g}, \quad (9)$$

$$\hat{\beta}^k \simeq \frac{\partial \bar{V}}{\partial q_r} = \frac{\tilde{V}(S_t + e_r) - \tilde{V}(S_t)}{\Delta q_r}, \tag{10}$$

where e is a 2-by-1 column vector with Δq in the respective entry and the other zero. We then update current coefficient matrix through:

$$A^k = (1-\theta_k) A^{k-1} + \theta_k \hat{A}^k, \tag{11}$$

where θ is the stepsize and takes the value of $1/k$.

H. Approximate Dynamic Programming Algorithm

The ADP algorithm can now be set as:

Step 1: Initialize A^0, S_0. Set t=0, and k=1.

Step 2: Obtain information vector W_t.

Step 3: Calculate

$$X_t(S_t) = \arg\min_{X \in \chi} \{ C_t(S_t, W_t, X_t) + C_{t+1}((S_{t+1} | S_t, W_t, X_t), W_{t+1}, X_{t+1}) + \gamma \bar{V}_{k-1}(S_{t+2} | S_{t+1}, W_{t+1}, X_{t+1}) \}. \tag{12}$$

Step 4: Obtain new estimations of coefficient matrix $\hat{A}^k$ using (8), (9) and (10), and then update coefficient matrix A^k using (11).

Step 5: Implement the optimum decision $X_t(S_t)$.

Step 6: If $t<T$, set $t=t+1$, $k=k+1$, and then go to step 2; else stop here.

V. Experiment Design

The experiments are designed to test the convergence of value function coefficients and the performance of ADP strategy. Experiments are realized via computer simulation. The testing bed for the experiments is an isolated four-arm junction. Considering the preliminary nature of our study, we prefer a kind of simplicity in the junction design so that it is just enough to fulfill the purpose of the investigation. Therefore, we only consider two traffic links: Link A from East to West, and Link B from North to South. The two links are mutually exclusive and no turning traffic is included. Flows on each link are random and have their constant mean rates in vehicles per hour. Flows are generated by random number generator with binomial distribution. Four optional in-flows are available to Link A , and they are 252 v/h, 396 v/h, 600 v/h, and 678 v/h respectively. Link B has two optional in-flows that are 240 v/h and 432 v/h. In total, there are 8 traffic in-flow combinations with Link A the major link and Link B the minor link. This design represents a typical sample of under-saturation traffic at an isolated junction.

Performance of ADP strategy will be compared with three other competing strategies which are BDP, OSCO (used in OPAC system, Ref. [10]) and RB (Robertson and Bretherton, Ref. [9]), of which BDP will serve as the benchmark. Since the significant advantages of all the three competing opponents over optimum fix-time method have been evidenced (Ref. [9], [10]), we will not include fix-time control in comparison.

We fix the following parameters, T=1200 (100mins in real time), Δq=1, γ=0.95, throughout the experiments. We artificially initialize A^0 at the beginning of each experiment by referring to (5) with the assumption that Y=0.833, which is equivalent to set traffic flow as 600 v/h on both links. This specification is to illustrate adaptive approximation and reinforced learning.

VI. Results

A. Convergence of Value Function Coefficients

The value of coefficients α and β converge in all experiments. As shown in TABLE 1, the converged values of α and β increase in proportion to the increase in traffic, thus the heavier the traffic, the more additional delay per vehicle. In each case, the value of β is higher than α, thus vehicles on red link suffer more delay than vehicles on green. This scenario directly reflects the problem structure of traffic signal control, and it is well preserved in operations. It is worth noticing that the structure preservation is achieved without explicit effort in algorithm development. The absolute difference between the two coefficients, however, is out of proportion to the changes in traffic. It floats between 0.945 and 1.211 over different pairs of traffic. Coefficients in all experiment converge to values significantly different from their initialization at the beginning of experiments.

TABLE 1 Converged Value of Coefficients

Arm A / Arm B	Con-efficient	252 V/h	396 V/h	600 V/h	678 V/h
240 V/h	α	0.437	0.739	1.100	1.217
	β	1.503	1.866	2.274	2.428
	| α - β |	1.066	1.127	1.174	1.211
432 V/h	α	0.789	1.128	1.757	1.941
	β	1.913	2.265	2.767	2.886
	| α - β |	1.124	1.137	1.009	0.945

We take two examples from the experiments to illustrate the process of convergence. The first example is shown in Fig. 2, with 396 v/h on Link A and 240 v/h on Link B. This is a relatively light flow condition. The second example represents a heavier flow condition, with 678 v/h on Link A and 432 v/h on Link B, and is shown in Fig.3. The values converge in both examples in the first half of the time horizon.

So far the convergence of coefficients has indicated the adaptive approximation of the value function. To further justify the robustness of the ADP strategy, we will compare its performance with both the benchmark which is BDP and other competing strategies.

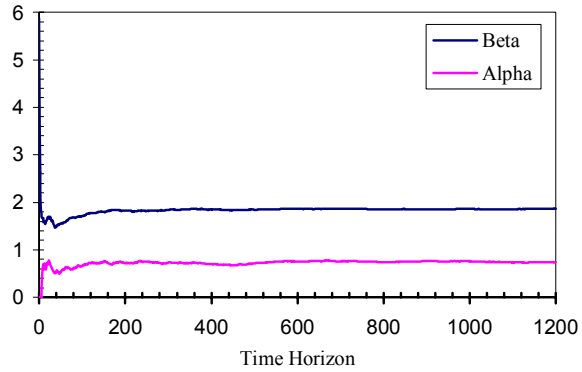

Fig.2. Convergence of coefficients
Link A: 396 V/h, Link B: 240 V/h

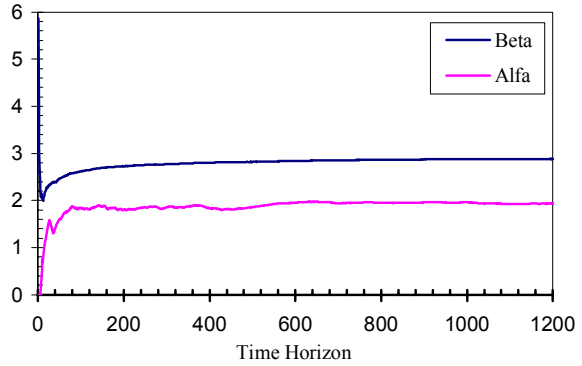

Fig.3. Convergence of coefficients
Link A: 678 V/h, Link B: 432 V/h

B. Performance

Performances of competing strategies in each experiment are grouped into TABLE 2 and TABLE 3. The former summarizes the performances in the first 10 minutes, while the latter summarizes the averaged performances in 10 minutes over the whole time horizon that is 100 minutes in total. The performance of ADP in the first 10 minutes, in comparison with benchmark and other competing strategies, varies in experiments. When approaching traffic flows are light, as shown on the left half of TABLE 2, ADP performs better than RB and OSCO in general. With higher traffic flows, the performance of ADP fluctuates around that of RB, but is constantly better than OSCO. Two factors may contribute to this occurrence: first, the presumed initial coefficient A^0, and second, the effects of transition.

TABLE 3 indicates that the performance of ADP in the long run is as good as RB, and significantly better than OSCO. With higher traffic flows, ADP may produce a slightly higher delay than RB in average, but the difference may owe to the effects of transition. Moreover, because of that in our experiments in-flows have constant hourly rates, the RB strategy does not have to estimate parameter Y over time. This further implies that ADP might be as good as RB in overall performance. Another noticeable feature of RB is that it can only operate with under-saturated traffic because of the structure of the denominator in (5). The ADP strategy is not limited to under-saturation, and a preliminary experiment has shown that it works well with over-saturation, with an acceptable gap from BDP performance.

TABLE 2 Vehicle delay (vehicle-intervals) in the first 10mins

Arm A / Arm B	Method	252 V/h	396 V/h	600 V/h	678 V/h
240 V/h	BDP	62	88	161	200
	RB	90	149	196	251
	OSCO	85	109	254	301
	ADP	84	112	249	288
432 V/h	BDP	122	206	418	486
	RB	157	246	481	542
	OSCO	170	279	497	645
	ADP	156	235	465	544

TABLE 3 Averaged vehicle delay (vehicle-intervals) per 10mins

Arm A / Arm B	Method	252 V/h	396 V/h	600 V/h	678 V/h
240 V/h	BDP	61	105	182	223
	RB	78	137	228	277
	OSCO	83	135	245	303
	ADP	71	123	238	287
432 V/h	BDP	119	208	403	525
	RB	145	252	453	589
	OSCO	158	263	486	641
	ADP	146	255	451	594

The gaps between the performances of ADP and the benchmark BDP shrink in the long run, and the short run differences may well correlate with short-term variation. However, it is still interesting to investigate the difference in performance between the two strategies. In Fig.4 we compare the performance of the two strategies thoroughly by plotting in/out flow profile and the evolution of queue on link A. The comparison consists of two parts. The first part compares the performances in the first 10 minutes (first 120 time intervals), thus representing the transition state. The second part compares the performance in the last 10 minutes, representing the steady-state.

In both time periods, ADP and BDP have the same number of cycles, and what make the difference are the starting time of a cycle and the cycle length. When queues are prominent in the system and short-term traffic is heavy, like in the time period between interval 40 and 80, and that between 1120 and 1160, the

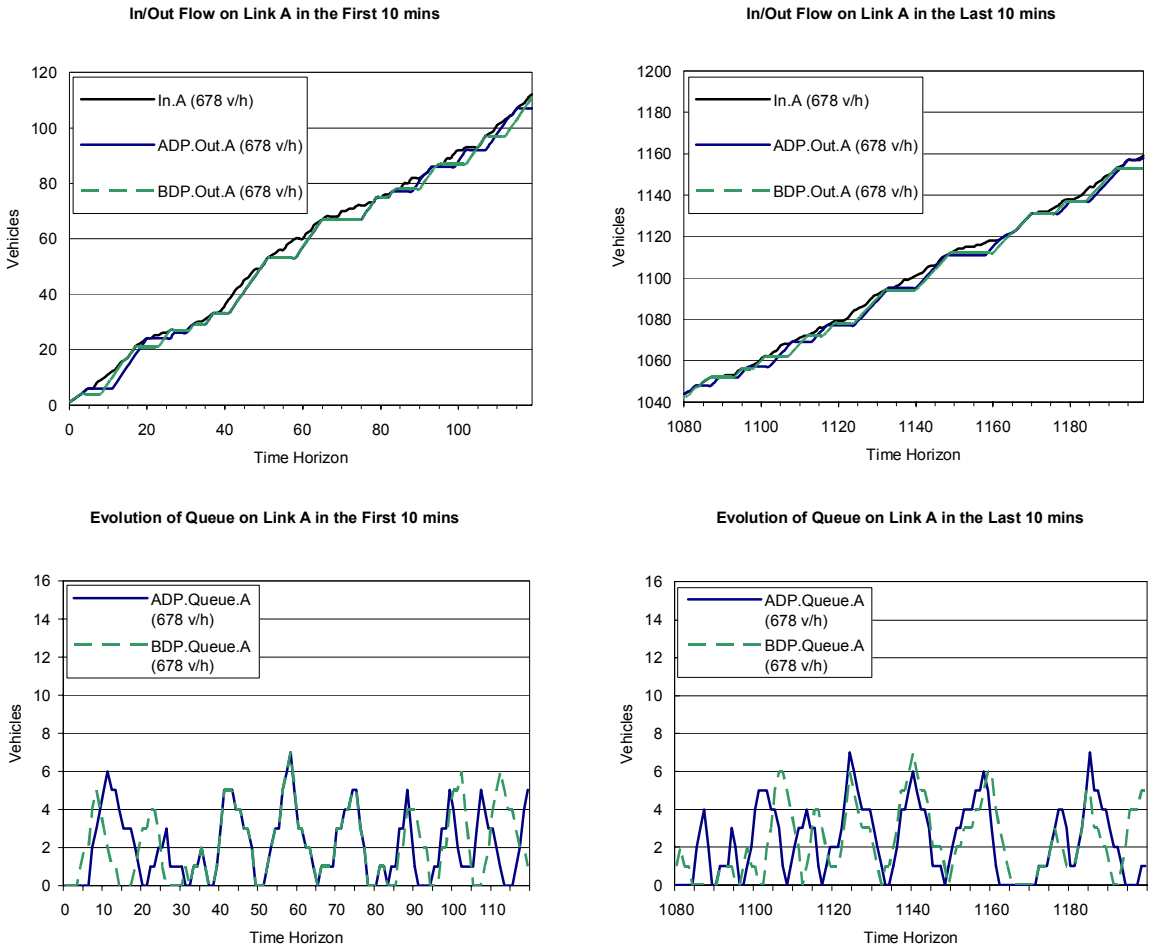

Fig.4. The accumulation of in/out flows and the evolution of queues on Link A under ADP and BDP

two strategies generate similar or even identical signal plans. This can be explained by, when queues in the system are prominent, the optimal solution is to dissipate queues on green link until clearance, or as much as possible. The arriving vehicles become less important to decision making. On the other hand, when queues are rare and traffic is light, BDP becomes more adaptive than ADP. This is because under such condition, a quick response to future arrivals is critical to maintain queue and hence delay at minimum. BDP utilizes the information of the whole time horizon to achieve global optimum, while ADP strategy utilizes the information of next 10 seconds, fairly close to real control situation. In these regards, we have indicated the robustness of the ADP strategy for responsive traffic signal control at an isolated junction with simple geometry.

C. Implication to complex control problems

The experiments in this paper are limited to an isolated junction with only two conflict links, and thus only two phases. There are nevertheless concerns about the possible extension to multi-phase signal control and to coordinate traffic signal control.

To address the concern about multi-phase junction, we need to modify the state variables, and thus the approximate value function to represent the property of each incompatible phase. Given a total number of I incompatible phases, for each phase i, we introduce queue state variable Q_i and signal state variable G_i, where

$$G_i = \begin{cases} [1 \quad 0]^{\mathrm{T}}, \text{ if link receives green,} \\ [0 \quad 1]^{\mathrm{T}}, \text{ if link receives red.} \end{cases}$$

A coefficient vector A_i, where

$$A_i = \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix},$$

is assigned to each phase. Like in the two-phase problem, α_i is a multiplier to queue on link i if it receives green, otherwise β_i applies. An advantage of introducing A_i instead of grouping all the queues to green and red is that the unique property of

each individual phase can be presented. The approximate value function is then given by

$$\bar{V} = \sum_{i=1}^{I} G_i^T A_i Q_i. \quad (13)$$

Equation (13) will be substituted to (12) to represent the multi-phase control problem, and the decision vector X will be further expanded to accommodate the additional options as which phase to change to. Coefficient vector A_i will be updated in the same way as in the two-phase problem. A preliminary experiment on multi-phase control with above specifications shows a good performance that replicates the adaptive property and convergence in two-phase problem. Except for the expansion in optional decisions, there is no significant increase in state space.

If an ADP strategy is able to work with multi-phase junction, it then can be extended to coordinate signal control by adopting the concept of decentralized network control as in [10] and [11]. In simple terms, it means the network controller finds a critical junction in the network at each time epoch to synchronize critical variables like minimum and maximum stage length, whereas performances are optimized by local controller at individual junctions with the subjection to the limits set by network controller. A sophisticated traffic model is indispensable to the network dynamic control, and there are wide choices of them.

VII. Conclusion

This paper, for the first time in dynamic traffic signal control, proposes a traffic responsive, self-adaptive optimizing strategy based on ADP architecture. Instead of an inseparable, exogenous factor dependent value function descended from earlier studies, the proposed strategy incorporates a separable, exogenous factor independent value function approximation. The strategy updates the coefficients of the approximate value function through the iterative estimation of partial derivatives. The experiments, though preliminary in junction geometry, have tested the convergence as well as the performance of the ADP strategy with a range of traffic combinations. The results have not only installed confidence in that the strategy is as good as the best existing control methods but also indicated the significance of the control strategy in real operation and more complex traffic control environment.

The success of ADP in preliminary dynamic traffic signal control opens a new opportunity by which we are able to develop the strategy in further to accommodate complex junctions with more sophisticated control objectives, and to expand to network control. Eventually, it is in complex, high dimensional problems where ADP technique excels. A deliverable ADP strategy in dynamic signal control may offer traffic engineers a powerful tool to manage an increasingly challenging control environment.

Acknowledgment

This work is supported in part by Rees Jeffreys' Road Fund, United Kingdom. The author also would like to thank Prof. B.G. Heydecker, Dr. S. Rana and the anonymous referees for their constructive comments.

References

[1] B. G. Heydecker, "Objectives, Stimulus and Feedback in Signal Control of Road Traffic," Intelligent Transportation Systems, vol. 8, pp. 63–76, 2004.

[2] F. V. Webster, "Traffic Signal Settings", *Road Research Technical Paper* No.39, Road Research Laboratory, 1957.

[3] R. E. Allsop, "Delay minimizing settings for fixed-time traffic signals at a single road junction," *Journal of the Institute of Mathematics and it's applications*, vol.8, pp. 164–185, 1971.

[4] B. G. Heydecker, and I. W. Dudgeon, "Calculation of signal settings to minimize delay at a junction," *Transportation and Traffic Theory*, edited by N. H. Gartner and N. H. M. Wilson, Elsevier, New York, pp. 159–178, 1987.

[5] A. J. Miller, "A computer control system for traffic network," *Proc., 2nd Int. Symp. on Theory of Road Traffic Flow*, London, pp. 201–220, 1963.

[6] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, "SCOOT – a traffic responsive method of coordinating signals." *TRRL Laboratory Report* 1014, 1981.

[7] A. G. Sims, and A. B. Finlay, "SCATS. Splits and offsets simplified (SOS),"*ARRB Proceedings,* vol. 12, No. 4, pp.17–33, 1984.

[8] R. A. Vincent, and J. R. Peirce, "'MOVA': traffic responsive, self-optimizing signal control for isolated intersections", *TRRL Research Report* 170, 1988.

[9] D. I. Robertson, and R. D. Bretherton, "Optimum control of an intersection for any known sequence of vehicular arrivals," *Proceedings of the second IFAC-IFIP-IFORS Symposium on Traffic Control and Transportation Systems*, 1974.

[10] N. H. Gartner, "Demand-responsive Decentralized Urban Traffic Control," *Part I: Single-intersection Policies.* DOT/RSPA/DPB-50/81/24, 1982.

[11] J. J. Henry, "PRODYN tests and future experiments on ZELT," VNIS '89: *Vehicle Navigation and Information Systems, IEEE Conference, Toronto,* 1989.

[12] R. Cheung, and W. B. Powell, "An algorithm for multistage dynamic network with random arc capacities, with an application to dynamic fleet management," *Oper. Res.* Vol. 44, No.6, pp.951-963, 1996.

[13] G. A. Godfrey, and W. B. Powell, "n adaptive, distribution-free approximation for the newsvendor problem with censored demands, with applications to inventory and distribution problems," *Management Sci.* Vol.47, No.8, pp.1101-1112, 2001.

[14] K. Papadaki, W. B. Powell, "A monotone adaptive dynamic programming algorithm for a stochastic batch service problem", *European Journal of Operational Research,* Vol.142, No. 1, pp.108-127, 2002.

[15] K. Papadaki, W. B. Powell, "An adaptive dynamic programming algorithm for a stochastic multiproduct batch dispatch problem," *Naval Research Logistics,* Vol. 50, No. 7, pp.742-769, 2003.

[16] W. B. Powell, A. Ruszczyński, and H. Topaloglu, "Learning algorithms for separable approximations of discrete stochastic optimization problems," *Mathematics of operations research*. Vol. 29, No. 4, pp.814-836, 2004

[17] W. B. Powell, *Approximation Dynamic Programming for Operations Research: Solving the curse of dimensionality.* Princeton University, Princeton, NJ 08544.

[18] B. G. Heydecker, and R. M. Boardman, "Optimization of timings for traffic signals by dynamic programming," *31st UTSG Annual Conference*, University of York, 4-6 January 1999.