Coordinated Road-Junction Traffic Control by Dynamic Programming

Tsin Hing Heung, Tin Kin Ho, *Member, IEEE*, and Yu Fai Fung, *Member, IEEE*

***Abstract*—This paper presents a novel approach to road-traffic control for interconnected junctions. With a local fuzzy-logic controller (FLC) installed at each junction, a dynamic-programming (DP) technique is proposed to derive the green time for each phase in a traffic-light cycle. Coordination parameters from the adjacent junctions are also taken into consideration so that organized control is extended beyond a single junction. Instead of pursuing the absolute optimization of traffic delay, this study examines a practical approach to enable the simple implementation of coordination among junctions, while attempting to reduce delays, if possible. The simulation results show that the delay per vehicle can be substantially reduced, particularly when the traffic demand reaches the junction capacity. The implementation of this controller does not require complicated or demanding hardware, and such simplicity makes it a useful tool for offline studies or real-time control purposes.**

***Index Terms*—Coordination, dynamic programming (DP), fuzzy-logic control (FLC), road-traffic control.**

I. Introduction

ROAD-JUNCTION congestion is now part of daily life in urban cities. When the population grows, traffic congestion only worsens, and constructing more roads does not always help matters much. In devising new measures or resources to accommodate the ever-increasing road-traffic demand, not much room is left for the city planners to maneuver. On the other hand, maximizing the capacity of the existing infrastructure remains one of the feasible means of relieving the congestion. Traffic control at a road junction involves assigning green time to each phase of the traffic. In addition to the basic function of ensuring the safe passage of vehicles, traffic control is also an essential tool for minimizing delay and maximizing junction utilization. Indeed, environmental awareness prompts drivers to eliminate unnecessary stops and shorten journey time. Effective and coordinated road-junction control can certainly help this cause.

Traffic control can be imposed on the level of an area, rather than on an isolated junction, in order to maximize the throughput of the available infrastructure. Control at junctions should be relatively simple with the assignments of green time to each traffic phase. However, with more traffic phases approaching the junction, the introduction of traffic-responsive strategies, and the consideration of uncertainties, such as turning traffic percentages and the relationship between queue length and waiting time, the control problem may turn out to be substantially complicated. Besides, junction control is effective only when coordination with the adjacent junctions is in place. Coordination is an ongoing process, while the junction controllers concerned are formulating their own control actions. The decision at one stage determines the state of the problem at the other. To achieve optimality in terms of delay reduction, through coordination within a traffic network, involves numerous time-varying parameters. It is technically feasible, but it may not always be practical in implementation, particularly for real-time control. An approach, which allows simple implementation and nonexcessive data communication, while pointing toward the general direction of delay reduction, provides a pragmatic solution.

We propose a decentralized approach on road-junction traffic control in this study. Each junction is equipped with a local controller and the projected traffic flow from the adjacent junction is the coordination parameter confining the possible control space. The assignment of green time to each phase of a traffic cycle is considered as a multistage control problem with a finite number of possible control actions at each stage. Dynamic programming (DP) is then adopted to facilitate coordination.

In this paper, Section II reviews the road-traffic control strategies and methodologies. A local fuzzy-logic controller (FLC) has been developed for individual junctions and it provides the basis for this coordinated control. The details of the junction controller and its learning capability are discussed in Section III. In Section IV, the application of DP on coordinated control among junctions is presented. The performance of the coordinated control is evaluated through computer simulation, and presented in Section V. Section VI concludes this paper with a summary of work.

Manuscript received March 25, 2004; revised October 14, 2004, January 25, 2005, and April 4, 2005. This work was supported by the Research Committee of the Hong Kong Polytechnic University, Hong Kong. The Associate Editor for this paper was W. Scherer.

T. H. Heung was with the Department of Electrical Engineering, Hong Kong Polytechnic University, Hong Kong. He is now with the Hong Kong Special Administrative Region Government, Hong Kong (e-mail: thheung@ogcio.gov.hk).

T. K. Ho and Y. F. Fung are with the Department of Electrical Engineering, Hong Kong Polytechnic University, Hong Kong (e-mail: eetkho@polyu.edu.hk; eeyffung@polyu.edu.hk).

Digital Object Identifier 10.1109/TITS.2005.853713

II. Road-Traffic Control

A. Single-Junction Control

Single-junction signal control mainly falls into two categories, fixed-time and traffic-responsive control. The former is a static optimization based on the statistics of the past traffic-flow patterns. Since similar flow patterns repeat every day, traffic demand within a day can be divided into a number of sections representing the morning and evening peak hours, afternoon and nightfall offpeaks. A timing plan for each section is then formulated and incorporated into the controller, which switches the plans according to the time of day. These timing plans are determined offline and cannot be adjusted frequently.

A fixed-time control system is simple in structure and does not require any vehicle detection in the vicinity of the junction. The lack of input of the real-time traffic condition is, however, a major drawback of a fixed-time controller, as it is incapable of responding to any unexpected changes in traffic demand. Hence, a fixed-time controller needs to be reevaluated and recalibrated regularly. Updates of control plans are based on collection and projection of traffic data. Since it is not practical to collect the data on a daily basis, a simple model of traffic growth, such as linear increment, is usually assumed. This is to ensure that the traffic plan is sufficient to deal with traffic growth over a period of time.

SIGSET has been one of the commonly used systems for fixed-time control plans. A signal cycle is divided into a number of phases that allow traffic from different directions to pass through with specific green time. SIGSET determines the optimal green time at each phase and the cycle time, while the sequence of the phases is fixed. Extensions to allow different phase combinations are possible [1], which leads to more complexity in the optimization.

Traffic-responsive control is an online process that takes real-time traffic data as input and optimizes the green time at each traffic light, according to the current traffic conditions. At the expense of extra hardware cost, mainly on detectors, traffic-responsive control provides the ability to adapt to real-time traffic variations. The availability of real-time data also enables the control to be extended to meet different performance criteria, such as minimization of stops, fuel consumption, or exhaust emission rate for the whole road network or subareas within the network [2].

Early research in traffic-responsive control [3], [4] focused on traffic flow or demand prediction. The philosophy was to predict what the demand would be in the next few traffic-light cycles and to optimize signal settings according to the predicted demand. However, difficulties in reliable flow prediction resulted in the failure of these early approaches. Another traffic-responsive approach requires the calculation of control parameters according to prevailing traffic conditions and the dynamical adjustment of cycle time and phase split [5].

B. Coordinated Control With Connected Junctions

Junction coordination involves organizing the timing of the traffic signals at adjacent junctions, in such a way that when a vehicle reaches a junction from another, the driver encounters a succession of green signals and need not to stop or slow down. The traveling time from one junction to another is, therefore, the optimal signal offset between the two junctions, if they have the same cycle time. Effective coordination can, of course, shorten queues and reduce delays. It has been proven that effective signal coordination reduced the average queues by four to ten vehicles with an onsite experiment [6]. Coordination of traffic signals in a road network should take into account the cycle length, phasing split, and offset of all signals in a road network, hence leading to a reduction in the average delay or the number of stops. However, determining the signal offset is not always simple in reality.

Fixed-time control is also available with connected junctions [7], [8] and it attempts to prescribe the signal settings in such a way that the traffic, particularly on a main or arterial road, proceeds without stopping at any signal, as much as possible. Again, having relied on historical data to formulate the signal settings, the fixed-time approach is not relevant to traffic dynamics, but it is applicable to unsaturated traffic.

On the other hand, the split, cycle, and offset optimization technique (SCOOT) is regarded as the leading example of the traffic-responsive approach, and is now in daily operation around the world. It has been pointed out in a study [9] that SCOOT has shown better performance than the fixed-time control does generally, but not overwhelmingly. Further, its performance deteriorates with saturated traffic conditions.

The traffic-responsive approach follows the concept of decentralized control, in which each single junction is regarded as an individual unit and looked after by a local controller. Every junction controller is responsible for coordination with its adjacent junctions by exchanging information like current phase, green time, and queue lengths. After gathering all the information from adjacent junctions, the controller makes decisions on the extension of the green-time period. Heydecker [10] provided a systematic method for constructing decentralized control, starting from the design of the signal sequence. Findler and colleagues [11], [12] demonstrated the idea of decentralized approach on solving traffic problems with a self-learning expert system installed at each junction. One of the advantages of decentralized control is that the computational demand is more manageable, because of the distribution of computation to local controllers.

C. Artificial-Intelligence (AI) Techniques

Numerous junction traffic controllers have been developed and tested with different extents of success. Because of the lack of an analytical model relating green time to queue length and the wide variation of driver behavior, AI techniques have become more popular with traffic control in recent years.

Pappis and Mamdani [13] are the pioneers of applying fuzzy logic to junction traffic control. Chiu and Chand [14] extended Pappis' study by introducing more complex rules into the controller. Ho [15] then showed that further reduction of the average delay can be obtained when fuzzy rules are adapted to current traffic conditions. The concept of the artificial neural network was also employed in junction control to predict flow rates and, hence, determine proper control actions for the junction [16], [17]. Findler and Stapp [11] implemented a self-learning knowledge-based traffic control system. With the exception of the last study, the above works only investigate simple junctions with no turning traffic. These studies showed a certain degree of success, particularly when the traffic is not heavy and the turning percentage is small. However, when traffic is heavy, more phases have to be introduced to the traffic-light cycle to cater to the turning traffic. An FLC was developed in a previous study, and it has been used as the local

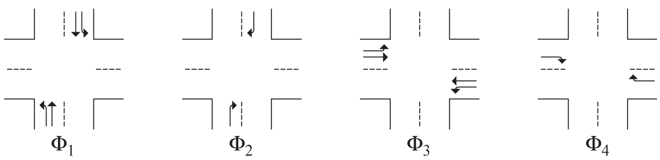

Fig. 1. Phase setting of a junction with right turnings.

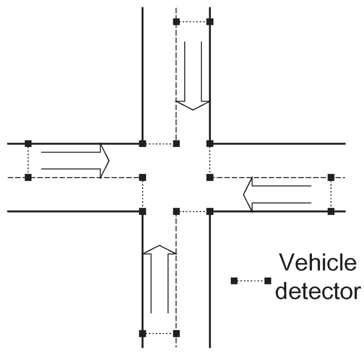

Fig. 2. Junction setup with vehicle detectors.

controller in this study [18]. It is capable of handling junctions with turning traffic, with a significant reduction on the size of the rule set. Advanced applications of AI techniques in traffic control is evident [19], but not very common. AI techniques in transportation systems, however, have found quite a number of applications [20]–[22].

III. Local Controller

A. Fuzzy Logic Controller (FLC)

In order to accommodate turning traffic at a single cross junction, a complete control cycle should consist of a minimum of four phases, as shown in Fig. 1, allowing traffic from the four phases $(\Phi_i, i = 1, 2, 3, 4)$ to pass through the junction in turn.

The following assumptions are made in this junction setup.

1) All vehicles are right-hand driven, hence left-turning traffic does not block the traffic of the opposite direction.
2) The interdependencies of the flow rates among the traffic streams are assumed to be minor and, hence, not accounted for in this study.
3) Vehicle arrivals are stochastic in nature with uniform distribution and the mean equal to the flow rate.
4) Saturation flow for each arm at the junction is d vehicles per second.
5) Each phase has a minimum of 10 s of effective green time.
6) The order of the traffic phases within a cycle is well defined and fixed.
7) The number of vehicles that the junction can hold at its arms is not bounded in this study.
8) The traffic at one arm is collectively regarded as one stream or lane, and the discharge rate of vehicles is the same as the saturation flow.

The objective of the controller is to determine the effective green times (t_{g}) for each phase, in such a way that the total delay at the junction is minimized. To gain the dynamic traffic data for the controller, vehicle detectors must be installed at the junction. The current queue length and flow rate in each traffic stream are the most important information to the controller. To count the number of vehicles entering the junction for flow-rate prediction, a vehicle detector must be installed at a distance from the junction on each arm. It is therefore possible to establish the queue length and the flow rate of each traffic phase. Fig. 2 shows the junction setup with vehicle-detector locations. Miscounting a long truck as two vehicles may happen, but the length of a truck is similar to that of two vehicles. Hence, the impact on the actual length of the queue is not significant.

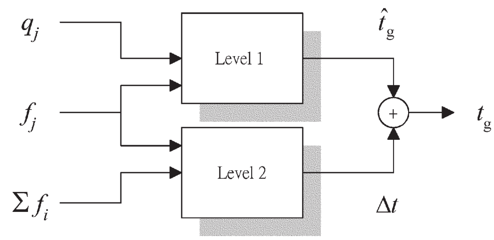

Fig. 3. Block diagram of FLC.

Assume that traffic phase Φ_j is given the right-of-way, the queue lengths at the traffic phases Φ_i are

$$q'_i = \begin{cases} q_i + f_i T_{\mathrm{lost}} + f_i t_{\mathrm{g}} - d t_{\mathrm{g}}, & i = j \quad \text{(1a)} \\ q_i + f_i T_{\mathrm{lost}} + f_i t_{\mathrm{g}}, & i \neq j \quad \text{(1b)} \end{cases}$$

where q_i and q'_i are the queue lengths at traffic phases Φ_i before and after t_{g}, the green time given to Φ_j, respectively. As there are two queues in opposite directions in a traffic stream, the longer queue between the two is taken as q_i or q'_i. f_i is the flow rate of Φ_i. T_{lost} is the lost time including the short red-amber time before the green period and the time elapsed before saturation flow is reached.

The two equations determine t_{g} in different ways, as they involve different parameters of the traffic conditions. Any attempt to attain t_{g} directly requires substantial computational time because of the number of inputs to the control problem. In order to simplify the control for real-time applications, a two-level FLC has been developed, in which an estimate of the green time $\hat{t}_{\mathrm{g}}$ is first made according to (1a), supplemented by a correction Δt with (1b), so that $t_{\mathrm{g}} = \hat{t}_{\mathrm{g}} + \Delta t$. The structure of the controller is illustrated in Fig. 3.

The first level of the FLC is a two-input, single-output controller, in which q_j and f_j are the input. This level provides an estimation of the green time required $\hat{t}_{\mathrm{g}}$ to clear the queue accumulated, just before the green signal is applied. When traffic is not saturated $(d > f_j)$, from (1a), q'_j will eventually become zero. In other words, vehicles on Φ_j can be discharged if t_{g} is large enough. Suppose $q'_j = 0$, from (1a), the first estimate of t_{g} is based on the minimum green time required to clear the queue and is as given as below:

$$\begin{aligned} \hat{t}_{\mathrm{g}} &= \frac{q_j + f_j T_{\mathrm{lost}}}{d - f_j} \\ &= (q_j + f_j T_{\mathrm{lost}}) + \frac{(q_j + f_j T_{\mathrm{lost}})\,(1 - (d - f_j))}{d - f_j}. \end{aligned} \quad (2)$$

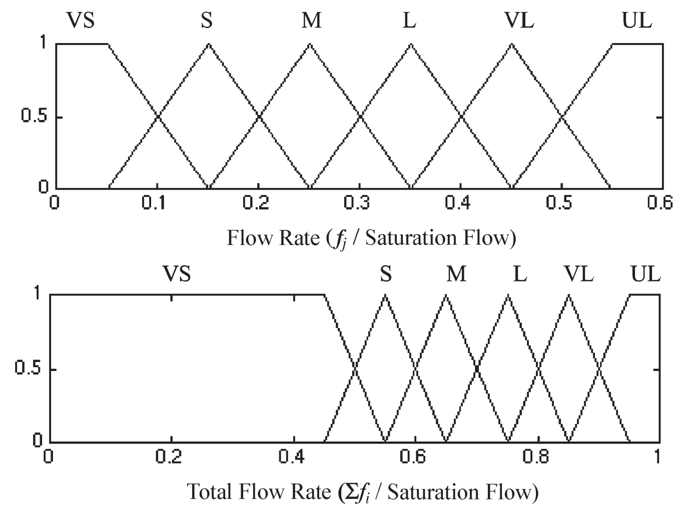

Fig. 4. Membership function of f_j and $\sum_{i=1}^{4} f_i$. VS: Very Small; S: Small; M: Medium; L: Large; VL: Very Large; UL: Ultra Large.

TABLE I
RULE TABLE FOR THE SECOND LEVEL OF THE FLC

		Total Flow Rate					
		VS	S	M	L	VL	UL
Flow Rate	VS	NS	NS	NM	NM	NL	NL
	S	NE	NE	NS	NS	NS	NM
	M	NE	NE	NE	NS	NS	NS
	L	NE	NE	NE	NE	NS	NS
	VL	PS	PS	PS	PM	PM	PL
	UL	PS	PS	PS	PM	PL	PL

NS: Negative Small NM: Negative Medium
NL: Negative Large NE: Negligible
PS: Positive Small PM: Positive Medium
PL: Positive Large

If $q_j + f_j T_{\text{lost}} \geq 1$, i.e., there is at least one vehicle at Φ_j initially, (2) becomes

$$\hat{t}_{\text{g}} \geq (q_j + f_j T_{\text{lost}}) + \left(\frac{1}{d - f_j} - 1\right). \quad (3)$$

$\hat{t}_{\text{g}}$ defines the minimum effective green time to be allocated, in order to clear all vehicles at Φ_j. It should be noted that if $q_j + f_j T_{\text{lost}} < 1$, the minimum effective green time of 10 s is given without evoking the controller.

On the other hand, t_{g} is restricted by traffic at other arms, according to (1b). The total queue length on phases Φ_is: $i \neq j$ is the sum of (1b) for these phases

$$\sum_{\substack{i=1\\i\neq j}}^{4} q'_i = \sum_{\substack{i=1\\i\neq j}}^{4} q_i + (T_{\text{lost}} + t_{\text{g}}) \sum_{\substack{i=1\\i\neq j}}^{4} f_i. \quad (4)$$

The second level of the controller determines an adjustment Δt according to the relationship of the flow rates among the traffic phases (i.e., f_j and $\sum_{i=1}^{4} f_i$).

The structure divides a three-input $(q_j, f_j, \sum f_i)$, one-output (t_{g}) FLC into two levels of control, with a two-input, one-output FLC at each level. Fig. 4 illustrates the membership functions of the two-input variables of the second level of the FLC, and Table I shows the rule table. An example of the fuzzy rules reads "If the flow rate is 'VS' and the total flow rate is 'M,' Δt is 'NM'." With the two FLCs, the total number of rules in the controller is reduced from N^3 to $2N^2$, where N is the number of fuzzy labels of each input. The center-of-area method has been adopted for defuzzification.

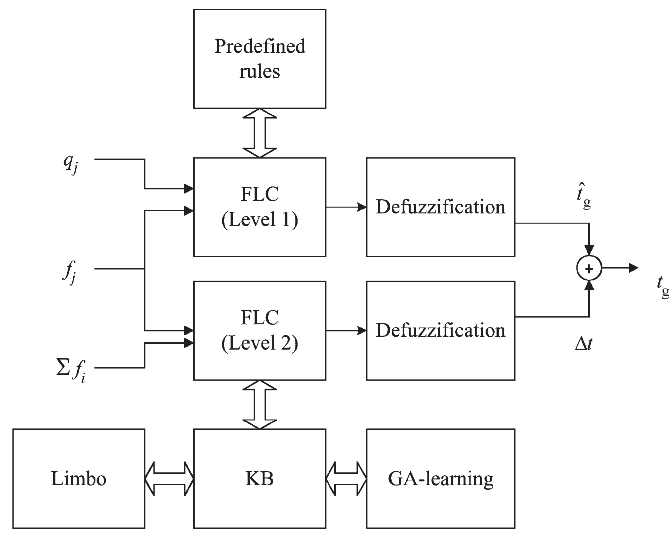

Fig. 5. Block diagram of the FLC with GA learning.

From the results of a previous study [18], this controller enables a significant delay reduction, in general. However, when the rate of change of traffic flow is high under time-varying traffic conditions, the transient behavior is even inferior to that of a fixed-time traffic controller [15]. It indicates that the FLC, particularly the second level in which the time-varying property of the traffic demand and the coordination among traffic streams are handled, does not always offer the best possible control actions. It is not at all unexpected because the rule set has been built and calibrated primarily by trial and error. A more systematic and application-oriented way to obtain the rules is necessary.

B. Genetic-Algorithm (GA) Learning

GA is one of the advanced searching techniques [23] and its capability of leading the search out of the local optimum and searching through multidimensional spaces has found numerous applications in function optimization and machine learning. GA has been widely employed for automatic fuzzy-rules generation or fine tuning [24], and even for applications in transportation systems [25]. The chromosomes in GA are the fuzzy rules (i.e., the fuzzy-relation matrix between the linguistic descriptions of inputs and outputs) and they are evaluated by certain fitness functions, according to the effectiveness of the rules in control action. The chromosomes are then evolved through genetic operators to produce fitter offspring and, hence, better rules.

In this study, GA learning is incorporated into the second level of the FLC to derive the rule set. Fig. 5 shows the block diagram of the revised FLC. A fuzzy rule in the second level of the FLC is in the form of

$$\text{If } f_j \text{ is } A, \text{ and } \sum f_i \text{ is } B, \text{ then } \Delta t \text{ is } T$$

where $A, B,$ and T are fuzzy values and each is given six possible labels. Each rule is, thus, encoded by a 12-bit binary number with 4 bits for each fuzzy value, and defined as a chromosome in the GA learning system.

There are two chromosome tables in the GA learning system, knowledge base (KB) and limbo. The chromosomes in the former will not be removed once they are retained, unless contradicting ones are found. The limbo provides temporary storage for the newly generated offsprings, which are then labeled with "age," "number of fires," and "fitness." The age is the number of generations a chromosome survives and it is increased by one with each generation. The maximum age of 20 is chosen as the termination criteria of the learning process. "Number of fires" indicates how active a chromosome is. In each generation, the candidates in the limbo are selected one at a time and subject to specific arrival patterns of predefined flow rates, together with the rules in the KB. The candidate is set "fired" when it achieves higher fitness than when it is not used. Its "number of fires" is then incremented by one. If the "number of fires" of a chromosome is higher than ten, that is the firing percentage $((\text{number of fires/age}) \times 100\%)$ exceeds 50%, it will be promoted to the KB. This percentage is set such that only highly active chromosomes are promoted.

When the size of the limbo reaches its maximum limit, the oldest chromosome is discarded from the limbo. The maximum size of the limbo is not critical to the performance of the FLC. A larger size presents a better chance of locating the best chromosomes, while the learning process may take longer. To balance between optimality of the chromosomes and learning time, a limbo size of 20 is taken.

The chromosome fitness is defined by the function $F = C/D$, where C is the number of vehicles leaving the junction and D (junction utilization) is the average delay per vehicle (delay). Only the chromosomes in the KB and those with best fitness in the limbo take part in the generation evolution. At each generation evolution, six chromosomes are selected. Roulette wheel selection, despite its slower convergence rate, instead of tournament selection, is used, so that as many rules as possible can be examined. The six selected chromosomes are randomly grouped into three pairs. Single-point crossover is employed to generate new rules, and mutation is applied with a small probability of 0.5%.

The offline learning process starts with an empty KB, and 20 chromosomes are randomly selected and placed in the limbo. At low flow-rate condition, the FLC, supported by the KB and one rule in the limbo in turn, produces green-time extension Δt over a period of time. After all chromosomes in the limbo have been evaluated, the one with the highest "number of fires" is promoted to the KB, and the next generation is then produced. The generation evolution goes on until no chromosome is promoted to KB or a maximum number of generations is reached. The same process repeats for a full range of flow-rate combinations of the four traffic phases so that the rules are trained for various traffic conditions.

As a result, the GA learning process produces a total of 23 fuzzy rules, which is a notable reduction when compared with 36 rules in the second level of the FLC. Through simulation, it has been shown that, with GA learning, the FLC enables more delay reduction when the junction is subject to constant flow-rate conditions. The most significant delay reduction occurs while the junction capacity is within 60–85%. When the traffic is light, most drivers experience very little delay and, hence, near-optimal journeys. There is not much room for further optimization anyway. On the other hand, when the traffic demand approaches the junction capacity, the queue starts growing, regardless of the traffic-control methodology.

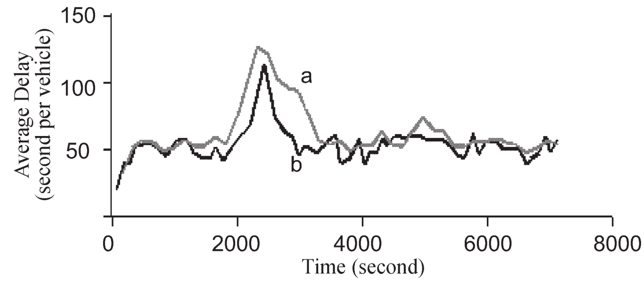

Fig. 6. Average delay of FLC with and without GA learning. (a) FLC without GA learning and (b) FLC with GA learning.

In order to test the FLC to the extreme of time-varying traffic demand, a step change of traffic demand is imposed on the FLC, and the junction is oversaturated for a short period of time, so that queues build up. Fig. 6 shows the average delay variations with and without GA learning, when a sudden increase of flow rate occurs at 1800 s, and it returns to a lower level at 2000 s. It is clear that the FLC produces less delay in general; delay overshoot due to queues piling up at the intersection is lower; and the recovery time from the extreme disturbance is shorter.

IV. Junction Coordination

As the junctions in a road network are interconnected, the decisions made at one junction inevitably affect those made at the adjacent junctions. Coordination among junctions is for organizing the control actions from adjacent junctions, in such a way that the traffic encounters a minimum of red signals throughout its journey. To facilitate coordination, what a local controller needs is the information on upstream traffic. The information to be exchanged between the local controllers at two adjacent junctions is, thus, their latest timings of the green-time allocations on the traffic stream connecting these two junctions and the corresponding expected-traffic volume, which are used to "adjust" the next green-time allocations derived within the two local junction controllers. The quantity of the exchanged information is very much minimal, imposing simple bandwidth requirement on the communication links between junctions.

At a junction controller, after an allocated green-time period on a traffic phase has started, it is not practical to make any amendment, despite possible available update on traffic information from the controller upstream. In other words, a succession of green-time allocations at a junction, in which each allocation may be subject to adjustment due to the junction coordination, has to be determined prior to the commencement of the first allocation. As different green-time adjustments at one allocation take the traffic conditions at a junction to different states (i.e., number of vehicles to be cleared at the green-time stream and queues building up at other traffic streams), the subsequent allocations may become different problems. Incorporating coordination into the local controllers is therefore equivalent to a multistage sequential decision-making problem.

DP is a long-established technique for solving multistage decision problems [26] and has found numerous successful engineering applications [27]–[29]. It converts a multistage decision process, containing many interdependent decision variables, into a series of single-stage problems, each containing only a few variables. It examines only a small subset of possible solutions, guaranteed, under the right conditions, to contain the optimal solution, and it relies heavily on the clear definition of stages and their state space, and how the system evolves from one stage to another. DP, therefore, provides an intelligent search for the solution to this coordination problem.

A. Setup

Under the notion of DP, a traffic cycle at a junction is considered as a number of consecutive stages. At each stage, the traffic condition at the junction is described collectively by a state. As there are four phases in one traffic-light cycle at a junction, the decision process is divided into four stages. Decisions must be made for the traffic at the junction to move on from one stage to another (stage transformation) and the possible sets of decisions depend on the stage and state the junction is in. The FLC described in the previous section is adopted as local junction controller and it determines an effective green time t_{g} for each stage. Junction coordination introduces an adjustment parameter t_{c}, which can be either positive or negative. The resulting sum $t_{\mathrm{e}} = t_{\mathrm{g}} + t_{\mathrm{c}}$ is the green-time allocation (the stage decision) at one stage.

A state is a set of variables describing the current traffic condition at the junction. The state variable $\mathbf{X}_s$ consists of $\mathbf{Q}_s$ and t_s. $\mathbf{Q}_s$ is a vector containing the queue lengths of all traffic phases at the state s, and they are available from the vehicle detectors. t_s indicates the accumulated time of the traffic cycle, and it obviously depends on how the state s is reached.

t_{e} is one of the two factors that enable the junction to proceed from one stage to the next, and also determine the state the junction should be in at the next stage. Another factor is, of course, the coordination parameters from the adjacent junctions, which is denoted as the vector $\mathbf{F}_s$, the projected number of vehicles approaching the junctions from all the traffic phases within the time interval $[\,t_s, t_s + t_{\mathrm{p}}\,]$. t_{p} is the time within which $\mathbf{F}_s$ can be reliably provided, and it should be larger than t_{e}. As the accuracy of $\mathbf{F}_s$ may diminish if t_{p} becomes too long, t_{p} indeed imposes a limitation on the range of t_{e}.

Assume $q_{s,i}$ and $f_{s,i}$ are elements of $\mathbf{Q}_s$ and $\mathbf{F}_s$ at traffic phase i, respectively, and the traffic phase j is to be given the green time. The stage transformation is governed by (5a) and (5b), which are similar to (1a) and (1b). The resultant queue lengths then determine the state which the junction is in at the next stage.

$$q_{s+1,i} = \begin{cases} q_{s,i} + f_{s,i}T_{\mathrm{lost}} \\ \quad + f_{s,i}(t_{\mathrm{g}} + t_{\mathrm{c}}) - d(t_{\mathrm{g}} + t_{\mathrm{c}}), & i = j \qquad (5a) \\ q_{s,i} + f_{s,i}T_{\mathrm{lost}} + f_{s,i}(t_{\mathrm{g}} + t_{\mathrm{c}}), & i \neq j. \qquad (5b) \end{cases}$$

The state space at a stage has to be finite to enable the comparisons at each state transformation. The decision set for possible values of t_{c} should also be finite, so that it only leads to a finite state space in the next stage. To attain a balance between a reasonable size of control space and a manageable computation demand, t_{c} is confined to be one of the elements of the following set

$$t_{\mathrm{c}} \in \{-20, -15, -10, -5, 0, 5, 10, 15, 20\}\ (\mathrm{s}).$$

In subsequent tests, it was found out that t_{c} rarely goes down to -15 s or even -20 s. The t_{c} set is, thus, cut down, in order to reduce the size of the control space, and it becomes $t_{\mathrm{c}} \in \{-10, -5, 0, 5, 10, 15, 20\}$ (s).

B. State Merging

Even with a finite set of t_{c}, the total number of possible states the junctions may be in throughout a traffic cycle is still enormous $(7^4 = 2041)$. In order to bring the number of possible states further down without compromising the control space, state merging is allowed when two or more states are equivalent. Two states are defined as equivalent when they satisfy the following conditions.

1) The same traffic stream is given the green time (i.e., state at the same stage).
2) t_s is the same.
3) Queue lengths at the traffic streams are similar.

Similar queue length means that the corresponding elements of $\mathbf{Q}_s$ do not differ by more than five vehicles, which equals to half of the width of the membership function for a fuzzy label in the FLC. More than 50% of reduction on the number of states, particularly in the last two stages, can be achieved by state merging in most of the test cases carried out in this study.

C. Cost Function

The objective of junction coordination is to minimize the average delay on the vehicles and the number of stops. The cost function of the DP process is defined as the weighted sum of delay and stops, and it is to be evaluated at every state transformation. The cost function z is

$$z = W_D D + W_S S \qquad (6)$$

where D is the average delay per vehicle at the junction, S is the average number of stops per vehicle at the junction, and W_D and W_S are the respective weighting factors.

In each stage transformation, for the vehicles queuing on the traffic phase given the green time, only the waiting times before they clear the junction contribute to D, whilst for other traffic phases, the vehicles are not allowed to move throughout t_{e}, which is, therefore, the additional delay imposed on each vehicle. The number of stops of the vehicles arriving at traffic streams with the red signal is incremented by one.

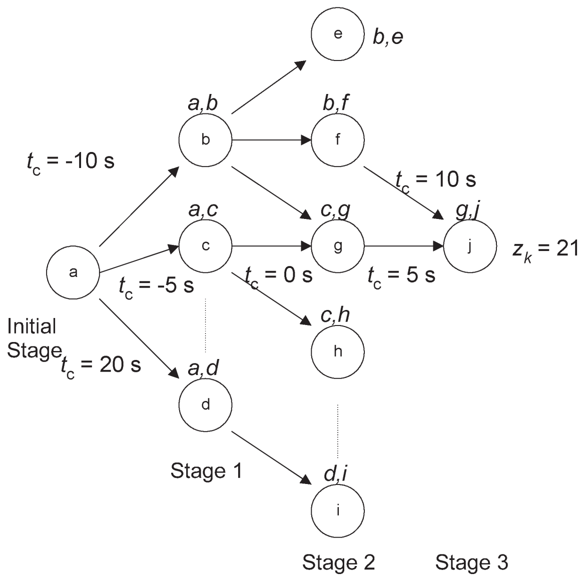

Fig. 7. Example of a state diagram with DP.

D. Overview

With the stages, states, state merging, and cost functions clearly defined, DP with forward recursion can be employed directly to obtain the sequence of decisions within a traffic cycle with reduced delays.

Fig. 7 is an example of a state diagram of this DP model, showing the progress from one stage to the next. There are four stages in a traffic cycle to represent the four phases, and the stage evolution starts from the initial stage. At each stage, the minimum cost and its corresponding path to reach each possible state is calculated and recorded. At the final stage (stage 3), the state with the overall minimum return (state j) is identified. The states on the optimal path are returned (a, c, g, j), and the decisions (t_c) to enable the move along this path are, thus, obtained.

The flowchart of this DP coordination system is illustrated in Fig. 8. It was implemented in C programming language and integrated with the local FLCs. The simulation results are discussed in the next section.

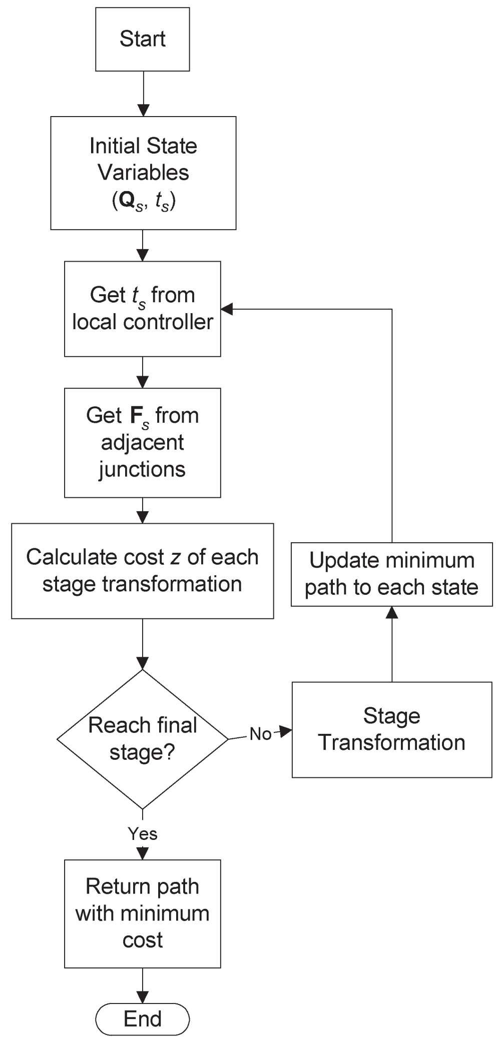

Fig. 8. Flow chart of the DP coordination.

V. Simulation Results

To demonstrate the performance of this decentralized traffic control with DP junction coordination, it is applied on a simple two-junction network which is subject to various traffic demands, and the FLCs presented are adopted as the local junction controllers. The performance is compared with that attained by two local FLCs with fixed offsets only. As the simulation focuses on the delays of individual vehicles, simple data representation of each vehicle is preferred over the sophisticated macroscopic or abstract level of vehicle-movement simulation [30], [31].

A. Junction Setup

A simple network of two connecting junctions is used as the test bed, and its layout is shown in Fig. 9. Vehicle detectors are installed in front of and in the vicinity of the junctions. They are shared between the two junctions. The connecting link between the two junctions can hold up to 50 vehicles in each direction, and the vehicles are moving in platoon, and no dispersion is assumed. It should be noted that this assumption is for simplicity in simulation. Dispersions are likely to occur in long arteries and space has to be inserted in the traffic-flow model to represent the substantial room between vehicles. The optimal offset of junctions having common cycle length is the traveling time of the vehicle in the common link. The turning percentages (both left and right) are set to 10%. Both weighting factors in the cost function W_D and W_S are set to one in the tests.

B. Constant Traffic Demands

The network is first tested with constant traffic flow rates feeding the incoming roads of the network. For comparison, two local FLCs are employed at the two junctions, while they are coordinated by a fixed offset. The offset is set at the average travel time between the two junctions, and the value is kept unchanged. As the green times, and hence the cycle times,

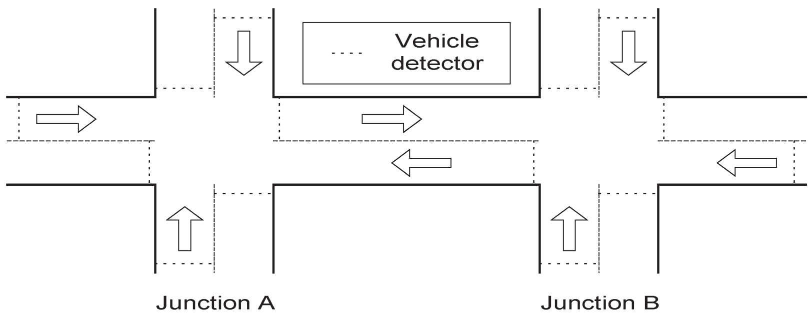

Fig. 9. Junction setup.

TABLE II
PERCENTAGE-DELAY IMPROVEMENT BY THE COORDINATED TRAFFIC CONTROL, WITH RESPECT TO THE FIXED-OFFSET COORDINATION

0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	Junction A Junction B
-18.5	23.1	14.17	13.64	3.2	-9.8	5.4	5.7	27.3	21.1	55.4	0.45
	-30.6	19.0	9.0	-8.6	-10.6	-1.2	-3.1	41.3	43.3	36.9	0.5
		-51.7	22.7	47.5	36.0	36.9	80.2	85.2	85.3	89.1	0.55
			-65.6	16.31	18.0	65.2	36.3	80.6	80.2	83.4	0.6
				-58.3	5.3	57.9	31.5	41.8	72.4	78.7	0.65
					-65.9	8.9	60.8	70.2	44.2	54.9	0.7
						-33.6	21.5	-10.4	56.4	64.2	0.75
							-158	-34.3	-69.6	-18	0.8
								-95.2	37.3	49.5	0.85
									-27.5	49.5	0.9
										-40.5	0.95

of a junction are derived by the local controller according to its local flow rates, different flow rates at the two junctions lead to different cycle times. If the cycle times at the two junctions are equal, common green time with optimal offset is achieved, which means zero stop time at the junctions. However, the offset between the two junctions varies with junction cycle times, which in turn change with flow rates. Hence, the fixed-offset coordination produces near-optimal results only when the flow rates of two junctions are similar. Indeed, fixed offset is a reasonable coordination, and it certainly provides the yardstick for preliminary performance evaluation on the DP coordination as the effect of the change on the offset enabled by the DP coordination, if any, can be easily singled out in this comparison.

Different combinations of flow rates (in vehicles per second) are imposed on junctions A and B. The saturation flow at the two junctions is 1 vehicle/s. The percentage reduction in delays under different traffic demands by the coordinated control is summarized in Table II.

In this study, the DP coordinated control outperforms the fixed-offset coordination in 48 out of 66 traffic conditions (more than 72%), particularly when the traffic demands on the two junctions are different. The DP coordinated control foresees the possible arrival of vehicles from the upstream and the subsequent buildup of queues, so that the vehicles on the common link are allowed to discharge more quickly before the link becomes saturated. It is more apparent when there are imbalanced traffic demands on the two junctions.

When the two junctions have equal flow rates (i.e., in the case of both junctions having common cycle time), the fixed-offset coordinator performs better than the DP one does, because the former is actually producing the optimal control actions. The vehicles traveling through the two junctions do not suffer from any delays. However, when the flow rates at the two junctions are different, common cycle time cannot be maintained. The fixed-offset coordination loses its edge and usually leads to long queues. It has also been found that the fixed-offset coordination is quite sensitive to the changes of traffic demands. A slight increase of flow rate will lead to a significant increase in delay. It explains why a number of junctions are usually grouped together and forced to share a common cycle time in practice. Incompatibility of the cycle time over the boundary of junction groups usually degrades the traffic flow, and congestion always occurs at the boundaries of junction groups.

C. Time-Varying Traffic Demands

Under this test, the same two-junction network is subject to a 24-h flow-rate profile as shown in Fig. 10, which resembles the traffic demand throughout a day at a typical road junction in a busy city with four time zones, namely Nighttime "NT," Morning Peak "MP," Daytime "DT," and Evening Peak "EP."

Junctions A and B can be regarded as the boundary of two junction groups. Junction A is a main entrance to the city center, while junction B is connected to a suburban area. During "MP," the majority of vehicles go from junction B to A, and the flows reverse in "EP." The flow rates of the two junctions under "NT" and "DT" are set equal, while there is a slight difference at the two peak periods. Within "MP," the flow rate at junction B is greater than that at junction A, and vice versa during "EP."

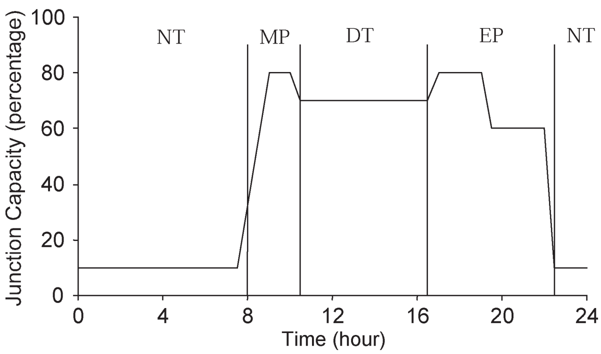

Fig. 10. Flow-rate pattern for the continuous flow-rate changing test.

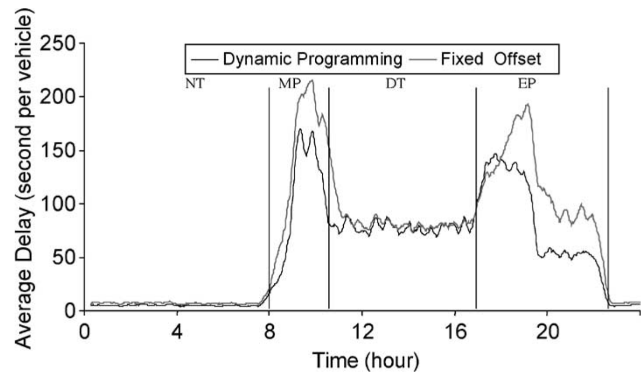

Fig. 11. Average delays by DP coordination control and fixed-offset coordination control under time-varying traffic demands.

Turning percentage is set to 5% during "NT" and "DT," and 10% during the two peaks. The delay variation is given in Fig. 11 and the average delays in the four time zones are summarized in Table III.

From the results, the DP coordinated control performs better in time-varying conditions, especially under the peak periods. It achieves more than 20% reduction in the average delay and the response to changes of flow rates, such as overshoot and transient, is significantly superior. On the other hand, the fixed-offset coordination performs slightly better than the DP coordination during NT and DT. It is because under these time periods, the flow rates of the two junctions are similar, which means the two local controllers have a common cycle time.

In general, the DP coordination provides a flexible means to deal with the imbalanced and time-varying traffic demands on the junctions, which usually happens in real life. It significantly reduces delays on average, and alleviates the momentary buildup of queues.

It has to be pointed out that DP does not always produce the optimal control actions even though it improves the traffic control performance in general. The discrete steps of 5 s in control action t_c and the necessary accuracy on the projection of the number of vehicles approaching the junction $\mathbf{F}_s$ are the possible causes of DP not having the right conditions to attain the optimal solution. Finer resolution on t_c may lead to better control, but the computational demand increases as a result. Reliable prediction of traffic with advanced detection and estimation techniques always play important roles in traffic control. Indeed, state merging in DP may, though not very likely, eliminate states which could have been on the optimal path.

On computation time required, DP coordination takes nearly double of the time for fixed-offset coordination. DP is not particularly efficient in large systems. In this study, the state space for DP is refrained by the number of stages (i.e., four stages for four phases) and states, and the resolution of control actions in order to keep reasonable efficiency.

VI. Conclusion

The development of a generic decentralized traffic-junction controller is presented here. The controller is simple in structure and does not require any geographic knowledge of the road network. The local controllers installed at the junctions are physically and functionally independent from each other. They employ fuzzy logic and genetic algorithms (GA) to handle the local control and the learning process, respectively. The coordination among junctions is introduced to the local controllers in the shape of the projected numbers of vehicles arriving through the connecting traffic links. The local controllers then allocate green time to the four phases of a traffic-light cycle by DP. Unlike the commonly used centralized control methods, the proposed coordination does not limit the number of junctions in the network. The computational effort is distributed to the local controllers, which is more computationally efficient, and the communication among the controllers is kept to a minimum. As the traffic demands do not usually change drastically, the local controllers should be able to cope with the traffic, even in the case of brief loss of communication.

The performance of this decentralized control has been examined under various traffic conditions and the results are satisfactory. It should be noted that the traditional traffic control methods and the fixed-offset coordination offer reasonably good control actions, particularly under normal and unsaturated traffic conditions. The proposed control and coordination supplement well for imbalanced and short-term oversaturated traffic. The investigation of integrated control, in which the most appropriate control is selected and applied according to the current traffic conditions, is one of the directions of further development.

Further work will also address the control of a more complex junction than a simple cross junction with a similar approach. When the number of phases in a traffic-light cycle is inevitably increased and the cycle time gets longer, it is not always practical to provide an accurate projection of vehicles coming upstream over a long period of time, as the local controller upstream might not have come up with its decisions yet. Reliable traffic prediction is, thus, an important part of the planned further work.

Acknowledgment

The authors also wish to thank the referees and the editor for the detailed review on the technical correctness and presentation quality. Their comments certainly helped improve the paper substantially.

TABLE III
AVERAGE DELAY (IN SECONDS) OF DECENTRALIZED AND FIXED-OFFSET CONTROL AT DIFFERENT TIME ZONES

	Fixed Offset	Dynamic Programming	Reduction (percentage)
Nighttime (NT)	25.49	27.69	-8.63
Morning Peak (MP)	147.30	106.12	27.96
Daytime (DT)	77.95	83.79	-7.49
Evening Peak (EP)	115.46	84.70	26.64
Overall	65.25	59.89	8.21

REFERENCES

[1] G. Improta and G. E. Cantarella, "Control systems design for an individual signalized junction," *Transp. Res., B*, vol. 18, no. 2, pp. 147–167, 1984.
[2] A. J. Al-Khalili, "Urban traffic control—A general approach," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, no. 2, pp. 260–271, Mar. 1985.
[3] D. W. Ross, E. E. Humphrey, R. C. Mahoney, R. C. Sandys, G. L. Williams, R. Whonoutka, P. J. Wong, and H. M. Leidler, "Improved control logic for use with computer-controlled traffic," Stanford Research Inst., Stanford, CA, NCHRP Project 3-18(1) Rep., 1977.
[4] P. J. Tarnoff, "The results of FHWA urban traffic control research: An interim report," *Traffic Eng.*, vol. 45, no. 4, pp. 27–35, 1975.
[5] R. A. Vincent and C. P. Young, "Self-optimizing traffic signal control using microprocessor: The TRRL "MOVA" strategy for isolated intersections," *Traffic Eng. Control*, vol. 27, no. 7–8, pp. 385–387, 1986.
[6] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time—The SCOOT method," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 11–15, Feb. 1991.
[7] J. D. C. Little, M. D. Kelson, and N. H. Gartner, "MAXBAND: A program for setting signals on arteries and triangular network," in *Transportation Research Record 795*. Washington, DC: U.S. Dept. Transp., 1981.
[8] M. R. Crabtree, R. A. Vincent, and S. Harrison, "TRANSYT/10 user guide," in *TRL Application Guide 28*. Crowthorne, Berks., U.K.: Transp. Res. Lab., 1996.
[9] J. Y. K. Luk, "Two traffic-responsive area traffic control methods: SCAT and SCOOT," *Traffic Eng. Control*, vol. 25, no. 1, pp. 14–20, 1984.
[10] B. G. Heydecker, "A decomposition approach for signal optimization in road networks," *Transp. Res., B*, vol. 30, no. 2, pp. 99–114, 1996.
[11] N. V. Findler and J. Stapp, "Distributed approach to optimized control of street traffic signals," *J. Transp. Eng.*, vol. 118, no. 1, pp. 99–110, 1992.
[12] N. V. Findler, S. Surender, Z. Ma, and S. Catrava, "Distributed intelligent control of street and highway ramp traffic signals," *Eng. Appl. Artif. Intell.*, vol. 19, no. 3, pp. 281–292, 1997.
[13] C. P. Pappis and E. H. Mamdani, "A fuzzy logic controller for a traffic junction," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, no. 10, pp. 705–717, Oct. 1977.
[14] S. Chiu and S. Chand, "Adaptive traffic signal control using fuzzy logic," in *IEEE Proc. Intelligent Vehicles Symp.*, Detroit, MI, 1992, pp. 101–106.
[15] T. K. Ho, "Fuzzy logic traffic control at a road junction with time-varying flow rates," *Electron. Lett.*, vol. 32, no. 17, pp. 1625–1626, 1996.
[16] J. C. Spall and D. C. Chin, "A model-free approach to optimal signal light timing for system-wide traffic control," in *Proc. 33rd Conf. Decision and Control*, Orlando, FL, 1994, pp. 1868–1875.
[17] F. S. Ho and P. Ioannou, "Traffic flow modeling and control using artificial neural networks," *IEEE Control Syst.*, vol. 16, no. 5, pp. 16–26, 1996.
[18] T. H. Heung and T. K. Ho, "Hierarchical fuzzy logic traffic control at a road junction using genetic algorithms," in *IEEE Int. Conf. Fuzzy Systems Proc. (FUZZ-IEEE)*, Anchorage, AK, 1998, pp. 1170–1175.
[19] W. H. Lin and C. H. Wang, "An enhanced 0-1 mixed-integer LP formulation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 238–245, Dec. 2004.
[20] I. Pavlidis, V. Morellas, and N. Papanikolopoulos, "A vehicle occupant counting system based on near-infrared phenomenology and fuzzy neural classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 2, pp. 72–85, Jun. 2000.
[21] L. Zhao and C. E. Thrope, "Stereo- and neural network-based pedestrian detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 3, pp. 148–154, Sep. 2000.
[22] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.
[23] C. R. Reeves, *Genetic Algorithms: Principle and Perspective: A Guide to GA Theory*. Boston, MA: Kluwer, 2003.
[24] J. Yu, Z. Ye, and C. Guo, "Using genetic algorithms in fine-tuning of fuzzy logic controller," in *IEEE Int. Conf. Intelligent Processing Systems*, Beijing, China, 1997, pp. 597–601.
[25] I. de la Escalera, J. M. Armingol, J. M. Pastor, and F. J. Rodriguez, "Visual sign information extraction and identification by deformable models for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 57–68, Jun. 2004.
[26] M. Sniedovich, *Dynamic Programming*. New York: Marcel Dekker, 2003.
[27] T. K. Ho, J. P. Norton, and C. J. Goodman, "Optimal traffic control at railway junctions," *IEE Proc., Electr. Power Appl.*, vol. 144, no. 2, pp. 140–148, 1997.
[28] H. T. Yang and K. Y. Huang, "Direct load control using fuzzy dynamic programming," *IEE Proc., Gener. Transm. Distrib.*, vol. 146, no. 3, pp. 294–300, 1999.
[29] J. Sadr and R. P. Malhame, "Decomposition/aggregation-based dynamic programming optimization of partially homogeneous unreliable transfer line," *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 68–81, Jan. 2004.
[30] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middleham, "Traffic flow modeling of large-scale motorway networks using macroscopic modeling tool METANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 4, pp. 282–292, Dec. 2002.
[31] Di Febbraro, D. Giglio, and N. Sacco, "Urban traffic control structure based on hybrid Petri nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 224–237, Dec. 2004.

Tsin Hing Heung received the B.Eng. degree in computer engineering from the Chinese University of Hong Kong, Hong Kong, in 1995, and the M.Phil. degree in electrical engineering from the Hong Kong Polytechnic University, Hong Kong, in 2001.

He is now working for the Hong Kong Special Administrative Region Government in defining interoperability framework and data standards for information and communication-technology systems.

Tin Kin Ho (S'91–M'92) received the B.Eng. and Ph.D. degrees in electronic and electrical engineering, in 1988 and 1994, respectively, from the University of Birmingham, U.K.

He joined the Department of Electrical Engineering of the Hong Kong Polytechnic University, Hong Kong, in 1992 and is currently an Associate Professor. His research interests include railway signaling and operation simulation, railway maintenance and condition monitoring, traffic control in networks, and applications of artificial intelligence techniques. He has been an Honorary Professor of Beijing Jiaotong University, Beijing, China.

Yu Fai Fung (S'89–M'90) received the B.Sc.(Eng) degree in electronic engineering and the Ph.D. degree in image processing from the University College of London, U.K., in 1986 and 1991, respectively.

He has been with the Department of Electrical Engineering, Hong Kong Polytechnic University, Hong Kong, since 1993, and is now an Assistant Professor. His research interests include parallel computer architectures, image processing, computer communication, and Internet applications.