Real-Time Coordinated Signal Control Using Agents with Online Reinforcement Learning

Min Chee Choy
Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3, E4 #05-48
Singapore 117576
Tel: +65-6874-5035
Email: engp1637@nus.edu.sg

Ruey Long Cheu*
Department of Civil Engineering
National University of Singapore
1 Engineering Drive 2, E1A #07-03
Singapore 117576
Tel: +65-6874-2153
Email: cheu@nus.edu.sg

Dipti Srinivasan
Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3, E4 #05-48
Singapore 117576
Tel: +65-6874-6544
Email: dipti@nus.edu.sg

Filippo Logi
Verkehrstechnik und Verkehrplanung
Technische Universitat of Munchen
Arcisstrasse 21, 80290, Germany
Tel: +49-089-289-22569
Email: logi@fgv.tum.de

Paper #03-2763
Reviewed by TRB Committee A5008 – Artificial Intelligence
For presentation at the 82nd Annual Meeting of the Transportation Research Board
and for consideration for publication in the Transportation Research Record

November 2002

* Corresponding author
Abstract: This paper introduces a multi-agent architecture for real-time coordinated signal control in an urban traffic network. The multi-agent architecture consists of three hierarchical layers of controller agents: intersection, zone and regional controllers. Each controller agent is implemented by applying artificial intelligence concepts namely fuzzy logic, neural network and evolutionary algorithm. Based on the fuzzy rule base, each individual controller agent recommends an appropriate signal policy at the end of each signal phase. These policies are later processed in a policy repository before being selected and implemented into the traffic network. In order to cope with the changing dynamics of the complex traffic processes within the network, an online reinforcement learning module is used to updating the knowledge base and inference rules of the agents. This multi-agent system with online reinforcement learning concept has been implemented in a network consisting of 25 signalized intersections, in a microscopic traffic simulator. Our initial test results have shown that the multi-agent system has improved the traffic condition in terms of average delay and total vehicle stoppage time, compared to that of a fixed-time traffic signal control.
INTRODUCTION

With the increase in urbanization and traffic congestion comes a greater demand to operate our road systems with greater efficiency. Optimization of traffic signals for the efficient movement of traffic on urban streets constitutes a challenging part of an Urban Traffic Control System (UTCS). Traffic responsive, closed-loop systems or adaptive traffic signal systems is becoming increasingly critical for transportation agencies to meet their day-to-day operations and management needs (1). Recent trend has shown the shift in the centralized traffic control to the distributed control schemes due to the later’s advantages in flexibility and scalability (2). For a large-scale UTCS, it may be difficult or impossible to tell whether the traffic network is flowing smoothly and assess its current state or the effects of modifying any of the traffic control parameters due to some of the nonlinear and stochastic traffic processes in a traffic network.

This paper presents an application of a distributed multi-agent architecture with online, unsupervised and reinforcement learning in traffic signal control. Embedded in the multi-agent system are various artificial intelligent and relatively new computational paradigms such as fuzzy logic, neural networks and evolutionary algorithm. This approach aims to leverage on the synergistic relationship between neural network and fuzzy logic systems (3-6) as well as to provide an innovative approach for online learning and self-organization via reinforcement learning (7-10).

Some of the past researches that involve the application of artificial intelligence techniques for signal control are can be found in (11-16). A few of these researches dealt with simplified traffic conditions (11, 13, 14) and hence the performance of such systems under a large complex traffic network has yet to be determined. In addition, the absence of an effective adaptation scheme in some systems, such as that in (11), limits the performance of the system in a dynamically changing traffic environment. Based on the researches mentioned, there remains scope for research into an online, adaptive and distributed multi-agent system for signal control in a real traffic network. As such, the main objective of this paper is to propose and demonstrate the effectiveness of a new distributed, cooperative problem solving approach using multiple interacting, autonomous agents in providing effective real-time signal optimization strategies.

PROPOSED HIERARCHICAL MULTI-AGENT ARCHITECTURE

For this paper, a traffic network simulating a real-world scenario in a central business district (CBD) consisting of 25 signalized intersections has been selected. This traffic network may be viewed as one region and is divided into several zones. Each zone consists of several intersections. The multi-agent architecture is constructed in the same hierarchical manner. As such, the architecture consists of three layers: the lowest layer consists of Intersection Controller Agents (ICA), the middle layer consists of Zone Controller Agents (ZCA) and the highest level consists of one Regional Controller Agent (RCA). The RCA controls all the ZCAs and each ZCA is in charge of several ICAs. The three-layered multi-agent architecture is shown in FIGURE 1.

The problem of real-time network-wide signal control is divided into several sub-problems, each with a different scale and magnitude. Individual agents from each layer of the architecture are tasked to manage the respective sub-problems according to their positions in the hierarchy. Each agent is a concurrent logical process capable of querying the environment (e.g., sensors and agents in the lower hierarchy within its control) and making decisions autonomously. The agents of each layer decide the policies (which comprises of signal timing plan adjustments and the direction of offset coordination) that they deem appropriate based on the conditions of the intersections under their jurisdictions. The higher-level agents who have a higher level of perception at zone or regional levels mediate these policies and if necessary, they can present a set of higher-level policies that overwrite the decisions made by lower level agents. Since each agent has its own knowledge base and inference engine, and is free to behave autonomously, individual agent can derive a unique and local solution of a larger zone or regional traffic control problem. For illustration purpose, a typical ZCA is shown in FIGURE 2.

The policy repository is a dynamic database for storing all the policies recommended by the agents at the end of each policy evaluation period. The end of an policy evaluation period is when all the intersections have finished their current signal phases. Therefore, the average policy evaluation period is approximately the average duration of a signal phase. After each period, the previously recommended policies are updated with a new set of policies. By making use of the policy repository, agents in the higher levels then performs arbitration and conflict resolution for the entire set of recommended polices before sending the chosen set of policies to the policy interpreter. The function of the policy interpreter is to translate the chosen set of policies into actions, which may result in adjustment of the various signal timing parameters such as phase length and direction of offset coordination, for the affected intersections.
FUZZY-NEURAL DECISION MAKING MODULE

Each agent has a Fuzzy-Neural Decision Making (FNDM) module that encompasses a knowledge base and an inference engine for the agent. In general, the FNDM modules of the RCA, ZCAs and ICAs have the same architecture, but with different input and output. For illustration purpose, the architecture of the FNDM module of a ZCA is shown in FIGURE 3. The FNDM module consists of two main functional blocks, namely, the Signal Policy Inference Engine for generating an appropriate signal policy and the Cooperative Factor Inference Engine for generating a suitable cooperative factor with the neighboring agents in the same hierarchy (in the case of the ZCA as shown in FIGURE 3, it is the inter-zonal cooperative factor). The architecture of the FNDM module follows the multi-layer feed-forward neural network that is implemented to mimic the fuzzy reasoning mechanism (17-19). As such, the fuzzy-neural architecture consists of five layers to represent, in between the layers, the fuzzification, implication, consequent and defuzzification processes. Using this approach, the FNDM architecture provides means to justify the choices of signal policy and level of cooperation.

At the lowest hierarchy, each ICA takes in the lane-specific occupancy, flow and rate of change of flow of the different intersection approaches as inputs. The occupancy and flow are two common input variables of a traffic signal control system, while the rate of change of flow is added to reflect change in traffic demand over time. These input parameters are measured by loop detectors during the green phase of an approach. In order to quantify the traffic conditions of the intersections in a zone, the FNDM module of the ZCA takes in each intersection’s representative occupancy, flow and rate of change of flow as its inputs. The fuzzy sets of Occupancy, Flow and Rate of Change of Traffic Volume are as shown in FIGURE 4 (the italic fonts are used to denote linguistic fuzzy sets). As can be seen from FIGURE 4, three linguistic labels, namely High, Medium and Low are used to describe the degree of membership for each of the fuzzy sets. The antecedents of the fuzzy rules are defined by properly linking the nodes in the second layer to those in the third layer. The third layer fires each rule based on the T-norm fuzzy operation, implemented using the MIN operator.

Nodes in the third layer define the degree of current traffic loading of the zone (i.e., High, Medium and Low Loads) and the level of cooperation needed for the intersections within the zone (i.e., High, Medium and Low Degrees of Cooperation). The nodes in the fourth layer represent the various consequents that correspond to the fuzzy rules in the FNDM module. For the Signal Policy Inference Engine, the consequents consist of the various signal timing adjustment policies (more details will be covered in a later section). For the Cooperative Factor Inference Engine, the consequents consist of the various possible levels of cooperation. Since some of the fuzzy rules share the same consequent, the S-norm fuzzy operation is used to integrate the rules. For this research, the S-norm fuzzy operation is implemented using the MAX operator. Finally, the fifth layer performs the defuzzification process in order to obtain crisp values correspond to the chosen signal policy and cooperative factor.

ONLINE REINFORCEMENT LEARNING MODULE

Reinforcement learning is a concept that has been successfully applied to enable multi-agent systems to learn and adjust their internal decision parameters without having to know the dynamics of the external environment (8, 9, 20). It is a guided trial and error process whereby an agent transforms the environment under its control from one state into another that contains or progresses towards its goal. Learning from the environment is robust because agents are directly affected by the dynamics of the environment. Another advantage of reinforcement learning is the ability to learn in an unsupervised manner due to the trial and error approach. FIGURE 5 shows the online reinforcement learning (ORL) module that is part of the multi-agent architecture. The role of the ORL module is to update the knowledge base and inference rules of the FNDM module within the agents, based on online data, simultaneously as the FNDM modules are being executed.

The ORL module is implemented using similar fuzzy-neural concepts that have been applied in the FNDM module. An example of the fuzzy rules in the ORL module is:

IF {(Overall Aggregate Occupancy is HIGH) AND (Overall Aggregate Flow is HIGH) AND (Overall Aggregate Rate of Change of Traffic Volume is HIGH)} THEN {(Traffic Loading is HIGH) AND (Congestion Expectation is HIGH)}

If this rule is satisfied, it is highly probable that a high traffic state value will be chosen during the defuzzification process, representing a congested traffic condition in the intersection, zone or region. The traffic state may be viewed as a qualitative description of average intersection delay in the intersection, zone or region. It is similar to the intersection level of service (LOS) concept, and is denoted by an integer value of between 1 and 8, with 1 being...
the “best state” and 8 being the “worst state”. As such, the ORL module is designed to generate reinforcement based on the comparison of the current estimated state value of \( (S_c) \) with the previous state value \( (S_p) \). A single reinforcement is generated and backpropagated down the hierarchy of controller agents (e.g., from the RCA, to ZCAs, and to ICAs) to facilitate online learning via weight adjustments of the neurons. The reinforcement signal \( R \) that is generated from the ORL module can be derived as follows:

\[
R = \lambda (S_c - S_p) - (S_c - S_b)
\]  

(1)

where, \( \lambda \) is the state change sensitivity constant, \( S_c \) is the current state value, \( S_b \) is the best state value and \( S_p \) is the previous state value. For there to be a positive reinforcement, it is necessary that \( S_c > S_p \) and \( \lambda (S_c - S_p) > (S_c - S_b) \).

Using the backpropagation technique, the change of weight \( \Delta W_{ij} \) from neuron \( i \) to the activated output neuron \( j \) is as follow:

\[
\Delta W_{ij} = -\eta \frac{\partial R}{\partial W_{ij}} = \eta S_j Y_i
\]  

(2)

where \( \eta \) is the learning rate, \( S_j \) is the gradient for the output neuron \( j \), and \( Y_i \) is the output of neuron \( i \). Note that,

\[
S_j = R_j \phi_j^l (V_j)
\]  

(3)

\[
V_j = \sum_{i=1}^{m} (W_{ij} Y_i)
\]  

(4)

in which \( m \) is the number of inputs for neuron \( j \), \( R_j \) is the backpropagated reinforcement value at output node \( j \) and \( \phi_j \) is the transfer function for neuron \( j \). The superscript \( l \) in \( \phi_j^l \) denotes the first order derivative of \( \phi_j \). For hidden layers of the neural network, the local gradient \( S_i \) for hidden neuron \( i \) with \( k \) neurons on its right is defined as follows:

\[
S_i = \phi_i^l (V_i) \sum_{j=1}^{k} (S_j W_{ij})
\]  

(5)

where

\[
V_i = \sum_{n=1}^{p} (W_{in} Y_n)
\]  

(6)

in which \( p \) is the number of inputs for neuron \( i \). Hence, if a fuzzy relation represented by a neuron is appropriate for a particular traffic condition, a positive reinforcement in the form of a positive \( R \) will be received and the weights of the neuron will increase positively.

The ORL module also serves as a critic for the performances of all the FNDM modules. However, it is necessary for the critic to adapt to the changing dynamics of the traffic control problem (10). This can be achieved through an additional traffic state estimator. The well-known Webster delay formula (21) is used to estimate the average intersection delay at a local intersection, in a zone, or in the entire region. At the end of each evaluation period, each ICA will calculate the average intersection delay by the Webster formula (in seconds per vehicle) using the approach volume measured by loop detectors. An intersection’s average delay is taken from the delay of its approaches, weighted by the approach volume. Similarly, using the volume as the weight, the average delay of the zone followed by that of the region are aggregated. Finally, the state value (from 1 to 8) that corresponds to the overall traffic condition in the entire region is assigned in increasing magnitude of the average delay. Based on this mapping, the error of state estimation, \( E \), made by the ORL module is computed as follows:

\[
E = \frac{1}{2} (\text{ORL traffic state value} - \text{traffic state value derived from Webster formula})^2
\]  

(7)
E is backpropagated as the error term replacing the reinforcement term, R in equation (2), for weight adjustment in the ORL module.

**SIGNAL CONTROL POLICIES**

For the proposed multi-agent architecture, agents control traffic flow within the traffic network through the selection and implementation of signal policies. Hence, each agent will have to recommend different signal policies derived from its perception on how the current traffic demand within its area of jurisdiction should be dealt with. There are three hierarchical sets of signal policies, namely intersection policies for ICA, zone policies for ZCA and regional policies for RCA.

The regional policy is the highest-level policy and it can affect all the intersections in the region. On the other hand, the intersection policy is the lowest-level policy and it only affects local intersection control by means of the ICA. Altogether, there are eight policies for each set, each addressing a different type of traffic condition. Each signal policy mainly concerns with adjusting the respective green time of the current phase as well as the offset with respect to a reference intersection. The policies are arbitrary derived and supplemented with observations on the magnitude of changes in phase time with traffic flow in the actual network. Intersection policies can make relatively larger magnitude of adjustments to the green time since the ICA is most sensitive to the local need of the intersection. On the other hand, the regional policies can only make moderate amount of adjustments to the green time since these high-level policies tend to affect many intersections and it is necessary that no abrupt changes be made to the overall signal plan. FIGURE 6 shows in detail the three sets of signal policies. Within each policy evaluation period, all the policies recommended by the controller agents are deposited in the policy repository. Not all the recommended policies are implemented immediately, but an arbitration process that gives priorities to higher-level policies is employed at the end of each policy evaluation period to select the most appropriate policies.

Once a policy has been selected, the agents perform coordinated offset adjustments dynamically if higher-level policies are recommended, e.g., policies 6, 7, 8 in the ZCA and RCA. This is done by coordinating the amount of offset between successive intersections with respect to a reference intersection, taking into consideration the link travel time. For this paper, offset adjustment can be done in two levels, namely regional level (whereby the number of intersections to be coordinated can be up to the entire region, theoretically) and zone level (whereby the number of intersections to be coordinated can be up to the entire zone). For each level, offset adjustment is calculated using an algorithm similar to the method for determination of offset in a grid as stated in (22).

**SIMULATION AND RESULTS**

The network used to evaluate the performance of the proposed multi-agent architecture is based on a section of the CBD of Singapore around the Raffles City and Suntec City area. The network’s traffic operations were simulated using Version 3.07 of PARAMICS microscopic traffic simulation tool (23). The necessary data for simulation was obtained from the Land Transport Authority (LTA) of Singapore.

The use of PARAMICS provides the advantages of microscopic simulation whereby details of the traffic operations such as driver behaviors can be modeled to reasonably close to the actual scenario. Vital information such as lane occupancy, flow and rate of change of flow was obtained via loop detectors in real-time. These loop detectors were coded in the simulated network at stop lines of the intersection approaches, same as the real-world installations. In addition, signal control directives (i.e., agents output, in the form of selected policy actions) can be implemented in the simulation via the PARAMICS Application Programming Interface (API) to effect the latest signal control actions while the simulation is running. Such API feature is not available in many other popular microscopic traffic simulation models, for examples, CORSIM (24). Using the API, information such as occupancy and flow during the green phase of each approach was extracted from the loop detectors, and rate of change of flow calculated. These data were fed into the respective ICAs at the end of each signal phase. Policies recommended from the various levels of controller agents were translated into signal control directives and changes were made to the signal plans in real-time, again by means of the PARAMICS API.

The actual intersections modeled in the simulated network and their respective logical representations are as shown in FIGURE 8. As can be seen, there are 25 ICAs (marked by the red circles) for the 25 signalized intersections, five ZCAs (marked by green circles) and one RCA (marked by the blue circle). Each zone has five ICAs and the jurisdiction of each zone is fixed throughout the simulation (although the zone boundary can be dynamically adjusted with the change in traffic conditions in future research). Java programming language and multi-threading technology is used to implement the logics in the FNDM and ORL modules in the multi-agent architecture.
Tests have been conducted to evaluate the performance of the traffic network with the existing signal control plans and with the proposed multi-agent architecture. The existing signal settings were based on the signal plans provided by LTA, between 0630 hrs to 1230 hrs on a typical weekday. Observation from the real signal plan data revealed that, the above period may be divided into five sub-periods: 0630-0730 hrs, 0730-0830 hrs, 0830-1030 hrs, 1030-1130 hrs and 1130-1230 hrs. Within each sub-period, the cycle time, phase plans and phase times of the intersections remained approximately constant. Therefore, for benchmarking purpose, the existing signal timing plans were implemented as fixed-time coordinated control in the five sub-periods in the simulation. Offsets were fixed along the major corridors, based on LTA data and adjusted to maximize the platoon progression in PARAMICS graphical user interface.

Even with the implementation of the multi-agent architecture, the signal plans were still subjected to the following restrictions:

- The signal phasing sequences provided by LTA were used, and the sequences were fixed.
- The cycle time was restricted to between 60 to 120 seconds.
- Each phase ended with an amber time of 3 seconds, followed by an all-red time of 1 second.
- Each phase has a minimum green of 10 seconds and maximum green of 40 seconds.

The total simulation period was six hours excluding the warm-up time of one hour. Hourly origin-destination matrices were derived from the intersection turning volume provided by LTA. Three repeated simulation runs, each with a different random number seed, were conducted. The average delay per vehicle and total stoppage time computed from all the vehicles are used as the performance criterion. From FIGURE 8 which plots the average results obtained from the three simulation runs, it can be seen that the average vehicle delay and total stoppage time in the network are lower when the multi-agent architecture was implemented. Through using the agents, the average delay per vehicle has been reduced by approximately 40% and the total stoppage time for vehicles has been reduced by approximately 50% at the end of the simulation run. The traffic network without the agents was beginning to degenerate into a pathological state after 1130 hrs as suggested by the positive gradients of the curves for the average delay and total stoppage time. FIGURE 9 compares the fluctuation of cycle time of at an intersection (ICA8) throughout the simulations. With the multi-agent control, the cycle time was adjusted continuously in response to the immediate traffic demand. FIGURE 10 plots the offsets selected by the policy arbitrator and implemented at the same intersection by the multi-agent system, in one of the simulation runs. In this figure, a 0 in the vertical axis is used to indicate no offset. A value of 1 indicates that this intersection was identified as the critical intersection of the zone. Directions 2 to 5 are assigned to incoming traffic, while 6 to 9 are for outgoing traffic. With the existing timing plan, the offset was fixed in direction 4 to 6, the direction with the highest traffic volume, throughout the simulation. However, the multi-agent system either assigned most of the offsets in the opposite direction (2 to 8), or chose to have no offset at all. As an example to show the improvement of the multi-agent control on queue length, screenshot of the network’s hotspots as shown in PARAMICS graphical user interface (shown in red circles, with diameter proportion to queue length of 13 or more vehicles) at 0930 hrs has been taken and presented in FIGURE 11.

**SUMMARY**

In this paper, a coordinated traffic responsive strategy is implemented using the distributed multi-agent architecture with unsupervised and online reinforcement learning. The constant updating of fuzzy rules in the FNDM module by the ORL module ensures the up-to-date representation of the traffic dynamics and responses to control actions. It has been shown that, through online reinforcement learning, the agents are able to hone their perceptions and represent the dynamics of the system. In doing so, each agent is capable of coordinating its local goal with the zone and regional objectives autonomously. The performance of the agent architecture is evaluated using a simulated model of an actual traffic network. Comparison is made between the performance of the agents’ dynamic signal policies and existing signal timing plans. Tests have shown that the multi-agent architecture can lead to improvement in terms of lower average delay per vehicle and total stoppages in the network. With this promising result, the authors will be conducting a comprehensive evaluation of the multi-agent traffic control scheme compared to some traffic responsive UTCS such as the SCATS, in a simulated environment. In addition, alternative methods for online adaptation will also be explored to enhance the online learning process by the multi-agent system. The signal control policies may also be fine-tuned to improve the system performance.
ACKNOWLEDGEMENT

The authors would like to thank the Land Transportation Authority of Singapore (LTA) for providing data necessary for the simulation modeling.

REFERENCES


LIST OF FIGURES

FIGURE 1  Schematic Diagram of the Multi-agents Architecture.
FIGURE 2  Schematic Diagram of a Zone Controller Agent.
FIGURE 3  The Fuzzy-Neural Decision Making Module of a Zone Controller Agent.
FIGURE 4  Fuzzy Sets of Occupancy, Flow and Rate of Change of Traffic Volume.
FIGURE 5  The Online Reinforcement Learning Module.
FIGURE 6  Different Levels of Signal Policies.
FIGURE 7  The Hierarchical Agent Architecture.
FIGURE 8  Results of Simulation Runs With and Without Agents.
FIGURE 9  Cycle Length at ICA8 With and Without Agents.
FIGURE 10  Direction of Offset Co-Ordination for ICA8 With and Without Agents.
FIGURE 11  Hotspots at 0930 hrs.
FIGURE 1 Schematic Diagram of the Multi-Agent Architecture.
FIGURE 2 Schematic Diagram of a Zone Controller Agent.
FIGURE 3 The Fuzzy-Neural Decision Making Module of a Zone Controller Agent.
FIGURE 4 Fuzzy sets for Occupancy, Flow and Rate of Change of Traffic Volume.
FIGURE 5 The Online Reinforcement Learning Module.
**Intersection Policies**

Intersection Policy 1 ⇒ Decrease the current green time of this intersection by 12%.
Intersection Policy 2 ⇒ Decrease the current green time of this intersection by 8%.
Intersection Policy 3 ⇒ Decrease the current green time of this intersection by 4%.
Intersection Policy 4 ⇒ Decrease the current green time of this intersection by 2%.
Intersection Policy 5 ⇒ Increase the current green time of this intersection by 2%.
Intersection Policy 6 ⇒ Increase the current green time of this intersection by 4%.
Intersection Policy 7 ⇒ Increase the current green time of this intersection by 8%.
Intersection Policy 8 ⇒ Increase the current green time of this intersection by 12%.

**Zone Policies**

Zone Policy 1 ⇒ Choose 2 intersections with the lowest rate of change of traffic volume. Decrease their current green time by 10%.
Zone Policy 2 ⇒ Choose the intersection with the lowest rate of change of traffic volume. Decrease its current green time by 10%.
Zone Policy 3 ⇒ Choose 3 intersections with the lowest occupancy. Decrease their current green time by 5%.
Zone Policy 4 ⇒ No zone level policy.
Zone Policy 5 ⇒ No zone level policy.
Zone Policy 6 ⇒ Choose 3 intersections with the highest occupancy. Increase their current green time by 5%. Offset adjustment.
Zone Policy 7 ⇒ Choose the intersection with the highest rate of change of traffic volume. Increase its current green time by 10%. Offset adjustment.
Zone Policy 8 ⇒ Choose 2 intersections with the highest rate of change of traffic volume. Increase their current green time by 10%. Offset adjustment.

**Regional Policies**

Regional Policy 1 ⇒ Choose 2 zones with the lowest rate of change of traffic volume. Decrease the current green time of all their intersections by 4%.
Regional Policy 2 ⇒ Choose the zone with the lowest rate of change of traffic volume. Decrease the current green time of all its intersections by 4%.
Regional Policy 3 ⇒ Choose 3 zones with the lowest occupancy. Decrease the current green time of all their intersections by 4%.
Regional Policy 4 ⇒ No regional level policy.
Regional Policy 5 ⇒ No regional level policy.
Regional Policy 6 ⇒ Choose 3 zones with the highest occupancy. Increase the current green time of all their intersections by 4%. Offset adjustment.
Regional Policy 7 ⇒ Choose the zone with the highest rate of change of traffic volume. Increase the current green time of all its intersections by 4%. Offset adjustment.
Regional Policy 8 ⇒ Choose 2 zones with the highest rate of change of traffic volume. Increase the current green time of all their intersections by 4%. Offset adjustment.

FIGURE 6 Different Levels of Signal Policies.
FIGURE 7 The Hierarchical Agent Architecture.
FIGURE 8  Results of Simulation Runs With and Without Agents.

(a) Average delay per vehicle

(b) Total stoppage time of all vehicles
FIGURE 9  Cycle Length at ICA8 With and Without Agents.
FIGURE 10  Direction of Offset Co-Ordination for ICA8 With and Without Agents.
Note: In PARAMICS, a red circle denotes a queue length of 13 or more vehicles, and the diameter of the circle is proportional to the queue size

**FIGURE 11** Hotspots at 0930 hrs.